

Universidad de las Ciencias Informáticas

Facultad 2



**Título: Automatización del subsistema DASNA.
Análisis y Diseño del módulo de control de la Formación
Emergentes de Auditores.**

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

Autor(es): Daiamna Pi Acuña

Julio Cesar Sanz Medrano

Tutor(es): Lic. Ivannis Suárez Jérez

Cuidad de la Habana, Julio 2008

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Daiamna Pi Acuña

Nombre completo del primer autor

Julio Cesar Sanz Medrano

Nombre completo del segundo autor

Lic. Ivannis Suárez Jérez

Nombre completo del Tutor

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

Albert Einstein

Agradecimientos

A mi papá Pedro Pablo por su constancia y su intransigencia. Por saber impulsarme durante el desarrollo de mi carrera. Por ser amigo y un magnífico padre a la misma vez.

A mi mamá Margarita por ser mi guía, mi ejemplo a seguir, por sus consejos, por darme fuerzas para seguir adelante, pero sobre todo por haber tenido fe en mí en todo momento.

A mi hermana Diana Noelia, por demostrarme la alegría siempre, por su ternura y sobre todo por considerarme su ejemplo.

A mi Tío Miguel, Tía Blanca Nieves y mi primo Leosbel por sus constantes preguntas, sus continuas preocupaciones durante estos 5 años de mi carrera.

A mi toda mi familia que aunque algunos se encuentren lejos supieron darme las fuerzas necesarias para llegar hasta el final.

A mis grandes amigas Yeilín, Ivette y Rita Elena por acompañarme en estos cinco años llenos de amor, alegría, tristezas pero sobre todo por demostrarme el verdadero significado de la amistad y hacer que nuestros sueños se hicieran realidad.

A mi gran amigo Arian Zulueta por darme un ejemplo a seguir, por sus constantes regaños, su sabiduría y por su enorme capacidad de saber entenderme en el momento preciso. Por sus revisiones y sugerencias a lo largo de estos cinco años.

A Luis Angel Santos por su cariño, por estar junto a mí en la recta final de mi carrera y demostrarme que puedo seguir adelante.

A mi tutor Ivannis por ayudarme siempre brindándome sus conocimientos.

A todas mis amistades, en especial a Dietmar Pupo que aún en la distancia mostraron interés en mi trayectoria por la universidad.

A Fidel y a la Universidad de las Ciencias Informáticas por darme la oportunidad de realizar mis sueños convirtiéndome en Ingeniera y así poder enorgullecer a muchas personas.

Daiana

Quiero agradecerles a todas esas personas que a lo largo de mi vida han contribuido a ser lo que soy porque sin ellos hoy no estuviera aquí.

A todos los colombianos, en especial a Eduardo, Michel, Luisí, Yisnier... Además, agradecerles a todos los amigos que he conocido estos 5 años, a mis compañeros de aula que me han apoyado y me han ayudado a la realización de este trabajo de diploma, a mi tutor Ivannis y a la Revolución Cubana por darme la oportunidad de estudiar.

Agradecerles a mi pareja de tesis y a todos los estudiantes del proyecto que nos brindaron su ayuda en los momentos en que la necesitamos.

Julio Cesar

A mis padres, mi hermana, a mi familia en general, a Arian Zulueta y todas aquellas personas que ocupan un lugar especial en mi vida.

Daiamna

A todos mis amigos y a todas las personas que son especiales en mi vida.

A mi familia, en especial a mi mamá y a mi hermano. Además a mi primo, a mi tía, a mis abuelos aunque uno ya no esté y a mi gran amiga y mi gran amor.

Julio Cesar

RESUMEN

La investigación está basada en el análisis y diseño de una aplicación Web para la gestión de matrículas del programa Formación Emergente de Auditores (FEA). Este programa es el encargado de controlar dichas inscripciones tanto en FEA como en las Sedes Universitarias Municipales (SUM) y se encuentra dentro de la Dirección de Atención al Sistema Nacional de Auditoría (DASNA) que se encarga de supervisar las demás direcciones y delegaciones del Ministerio.

En el programa FEA, todos los procesos son realizados de forma manual, las matrículas de los estudiantes, la baja de los mismos y la confección de una serie de documentos requeridos por el ministerio que son almacenados en archivos físicos.

Mediante el siguiente trabajo de diploma se propone una solución a estos problemas por medio del diseño de una aplicación Web que facilite el almacenamiento de la información de las matrículas de los estudiantes tanto en FEA como en SUM. De esta forma se evitará la pérdida de información de las matrículas de los estudiantes y facilitará las búsquedas de las mismas y demás informes sean más rápidas.

ÍNDICE

RESUMEN	VII
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Formación de auditores en el mundo.....	5
1.2 Formación de auditores en Cuba.....	7
1.3 Metodologías, técnicas y herramientas utilizadas para el desarrollo.....	9
1.3.1. Metodología de desarrollo.....	9
1.3.2. Herramientas de desarrollo	13
1.3.2.1. Herramienta de gestión de proyectos.....	13
1.3.2.2. Herramienta de gestión de configuración.....	13
1.3.2.3. Herramienta CASE.....	14
1.3.2.4. Herramienta de control, seguimiento y gestión de errores.....	15
1.3.3. Sistema Operativo.....	16
1.3.4. Plataforma de desarrollo	16
1.3.5. Lenguaje(s) de programación.....	17
1.3.6. Entorno(s) de Desarrollo Integrado.....	18
1.3.7. Framework(s), componente(s) y/o librería(s).....	18
1.3.8. Gestor de base de datos	19
1.3.9. Servidor de aplicaciones y/o web.....	19
1.4. Conclusiones.....	20
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	22
2.1. Problema y situación problemática.....	22
2.1.1. Objetivos estratégicos de la organización.....	22
2.1.2. Flujo actual de los procesos involucrados en el campo de acción.....	23
2.1.3. Análisis crítico de cómo se ejecutan actualmente esos procesos	24
2.2. Procesos que serán objeto de automatización.....	25
2.3. Propuesta de sistema.....	25
2.4. Modelo del negocio	25
2.4.1. Especificación de los requisitos de software.....	30
2.4.1.1. Requisitos Funcionales	30
2.4.1.2. Requisitos no Funcionales	31
2.4.2. Actores del sistema.....	33
2.4.3. Definición de los casos de uso.....	33

2.5. Conclusiones.....	55
CAPÍTULO 3: DISEÑO DEL SISTEMA	56
3.1. Descripción de la Arquitectura de Software Propuesta.....	56
3.2. Descripción del Framework.....	57
3.3. Modelo de diseño	65
3.3.1. Patrones de diseño.	66
3.3.1.1. Patrones GRAPS.	66
3.3.1.2. Diagramas de clases del diseño.	68
3.3.1.3. Diagramas de secuencia.	73
3.3.2. Diseño de la base de datos.....	76
3.3.2.1. Modelo Entidad Relación	77
3.4. Conclusiones.....	78
CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD	79
4.1. Estimación: Cocomo II con salida de puntos de función.	79
4.2. Conclusiones.	83
CONCLUSIONES GENERALES	84
RECOMENDACIONES	85
BIBLIOGRAFÍA CONSULTADA.....	86
BIBLIOGRAFÍA.....	87
ANEXOS	89
Anexo #1 Descripción de Casos de Uso del Sistema	89
CU. Gestionar Confirmación de Matrícula.....	89
Anexo #2. Prototipos de Interfaz de Usuario	95
Fig. A-1. Gestionar Matrícula en FEA	96
Fig. A-2. Matricular estudiante en FEA.....	97
Fig. A-3. Pasar estudiante de año en SUM	98
Fig. A-4. Informe General de matrículas en FEA.	99
Fig. A-5. Insertar pre ubicación laboral	100
Fig. A-7. Mostrar estudiantes por año de estudio en SUM	102
GLOSARIO DE TÉRMINOS.	103

INTRODUCCIÓN

Las tecnologías de la información, actualmente son elementos fundamentales para la superación y desarrollo de un país. Por eso, los países desarrollados basan su crecimiento en la aplicación y la programación estratégica de las herramientas computacionales y han definido políticas que los inducirán a su permanencia en el dinamismo mundial de los próximos años.

Cuba, a pesar de ser un país subdesarrollado, ha implementado técnicas que han beneficiado al desarrollo tecnológico de la sociedad, identificando desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones. En tal sentido, la informática en Cuba se desarrolla de manera acelerada y ascendente, lo que ha coadyuvado incuestionablemente a humanizar el trabajo, centralizar la información y que esta fluya eficaz y rápidamente, significando inequívocamente en un aumento considerable de nuestro nivel de vida.

La creciente necesidad de productos informáticos del mercado, apoyado por el éxito de otras producciones intelectuales realizadas por el país, llevan a la dirección de la Revolución Cubana a pensar en la Informatización de la Sociedad. Son grandes los esfuerzos que ha desarrollado el estado cubano en aras de diversificar esta importantísima tarea, entre las que se destacan la inauguración en el año 2002 de una Universidad específica para las Ciencias Informáticas, y en la década de los 80 la creación de los Joven Club de Computación, donde se han preparado miles de jóvenes para poder enfrentar esta vital tarea, lo que ha significado un aumento del nivel técnico profesional y por ende de nuestra cultura general integral, poniéndonos en un lugar cimero a escala internacional.

Actualmente, las técnicas de computación en el país abarcan la introducción masiva de equipamiento informático, llevando a cabo la informatización en los diversos sectores de la salud, la educación y la economía. Ante este nuevo entorno informático, se están llevando a cabo las capacitaciones a los profesionales en áreas distintas a la informática y las telecomunicaciones, capaces de enfrentar los retos que se tienen hoy en día. De esta forma, la presencia de la computación en los sectores productivos es un factor determinante para su funcionamiento.

Una prueba de esto es el desarrollo informático de la aplicación para el Ministerio de Auditoría y Control, obteniendo con esto una informatización que brinde beneficios para el almacenamiento de grandes volúmenes de información y que faciliten una búsqueda adecuada de la información.

Este ministerio se creó como un organismo de la Administración Central del Estado, encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en materia de Auditoría Gubernamental, Fiscalización y Control Gubernamental, así como para regular, organizar, dirigir y controlar metodológicamente el Sistema Nacional de Auditoría. Cuenta con diferentes direcciones entre las que se encuentran la Dirección de Auditoría Gubernamental, Dirección Jurídica, Dirección de Recursos Humanos, Dirección de Desarrollo, Dirección de Atención al Sistema Nacional de Auditoría, etc., las cuales tienen, en su esfera de competencia, funciones y atribuciones.

La Dirección de Atención al Sistema Nacional de Auditoría es la encargada de supervisar a las Direcciones y Delegaciones del Ministerio, así como al resto de las unidades del Sistema Nacional de Auditoría, a los efectos de garantizar la realización del encargo estatal encomendado al Ministerio. A esta dirección está vinculada la Formación Emergente de Auditores (FEA), programa implantado para suplir la carencia de auditores que hay en todo el país. Este programa brinda la posibilidad a los estudiantes que estén en 4to año del tecnológico y los graduados de técnico medio en contabilidad de recibir un curso y convertirse en auditores, inscribiéndose más tarde, para completar dicho objetivo, en las Sede Universitarias Municipal (SUM).

Posterior a la matrícula de los estudiantes a este programa, cada provincia envía documentos que guardan la información de dichas matrículas, confeccionando más tarde un documento general. Estos documentos son enviados a las personas encargadas de archivar la información por lo que en ocasiones se pierde el control de los estudiantes que están matriculados en la FEA. De igual manera, debido a que los listados se actualizan una vez al año, no tienen la posibilidad de tener un registro de todos los estudiantes que causan baja a mediado de curso, pues estos documentos, una vez enviados se archivan, lo que imposibilita una búsqueda rápida de la información.

Estos procesos realizados en el programa de Formación Emergente de Auditores, se realizan manualmente, lo que atrasa la búsqueda de información y almacenamiento de la misma que son de gran importancia para el trabajo de la dirección.

Este trabajo surge como la necesidad de dar una solución a todos los problemas antes dicho, de ahí que se planteó como **problema científico** la siguiente interrogante : ¿Cómo facilitar la gestión del control de estudiantes vinculados a la Formación Emergente de Auditores(FEA) de la Dirección de Atención al Sistema Nacional de Auditoría en el Ministerio de Auditoría y Control?

Para dar solución al problema planteado anteriormente se definió como **Objeto de estudio**: Los procesos que se llevan a cabo en la Dirección de Atención al Sistema Nacional de Auditoría, del Ministerio de Auditoría y Control.

Y como **campo de acción**: Los procesos de matrícula del Módulo de control de la Formación Emergentes de Auditores, de la Dirección de Atención al Sistema Nacional de Auditoría, del Ministerio de Auditoría y Control.

Se trazó como **objetivo general**: Modelar un sistema informático para la automatización de los procesos de control de la Formación Emergente de Auditores en la Dirección de Atención al Sistema Nacional de Auditoría del Ministerio de Auditoría y Control en el país.

Para cumplimentar el objetivo general se diseñaron los siguientes **objetivos específicos**:

- a) Estudiar los procesos que se llevan a cabo en el módulo de control de la FEA.
- b) Proponer a través del diseño un Sistema que facilite los procesos de la FEA.
- c) Realizar una propuesta de prototipo de interfaz de usuario no funcional.

Para cumplir con los objetivos anteriormente expuestos se trazan las tareas de investigación siguientes:

- Consultar la bibliografía existente para la fundamentación de conceptos.
- Realizar entrevistas y encuestas a personas/actores implicadas en los procesos para poder entender y modelar los procesos de la FEA.
- Realizar el análisis y diseño del sistema.
- Justificar las herramientas que se utilizarán para la realización del trabajo.
- Diseñar una propuesta de interfaz de usuario no funcional.

EL contenido de este documento está estructurado en cuatro capítulos distribuidos como sigue:

En el primer capítulo **Fundamentación Teórica** se tratan aquellos temas que constituyen la fundamentación teórica de la investigación a realizar. Incluye un estado del arte del tema tratado en el

ámbito tanto nacional como internacional, de las tendencias, técnicas, tecnologías y metodologías, tratando los procesos de matrículas de estudiantes mediante procesos computarizados utilizados mundialmente.

En el segundo capítulo **Características del sistema** se describe a profundidad el objeto de estudio, así como la modelación de los procesos de negocio. Se proporciona una descripción de la solución propuesta, definiéndose los requisitos que debe cumplir la misma.

En el tercer capítulo **Diseño del sistema** se describe a profundidad la construcción de la propuesta de solución mediante los diversos artefactos que especifica el proceso de software utilizado. Se describe el modelo de negocio, los requerimientos y los casos de uso del sistema.

En el cuarto capítulo **Estudio de la factibilidad** se realiza un breve análisis de la rentabilidad del proyecto. A través de este estudio se podrá obtener el tiempo de desarrollo del proyecto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se pretende abordar el tema referente a la formación de auditores tanto en el mundo como en Cuba. Se tratan las características de algunos de los softwares existentes para hacer un posterior análisis con la perspectiva de una posible conciliación. Se referirá a los desarrollos actuales de los procesos informatizados mediante las técnicas, tecnologías y metodologías.

1.1 Formación de auditores en el mundo.

Durante muchas décadas, la auditoría permaneció unida a la detección y prevención del fraude y las irregularidades, la evolución económica y social de las últimas décadas ha traído consigo cambios sustanciales en el campo de la auditoría. Uno de los cambios fue la adecuación de los códigos de ética profesional como respuesta al consumismo o al acelerado desarrollo de las computadoras que han permitido almacenar y registrar la información de una forma más operativa.

Dada la situación actual del mercado laboral, muchos de los nuevos Licenciados en Administración y Dirección de Empresas se inician profesionalmente trabajando en una Firma de Auditoría. Es por eso que se inició una nueva era para formar auditores.

El Registro Internacional de Auditores Certificados (The International Register of Certificated Auditors, IRCA) es el primer organismo internacional de certificación de auditores de sistemas de gestión. Más de 50,000 estudiantes eligen cada año completar cursos de formación certificados por el IRCA en más de 100 países. Actualmente, hay más de 90 organizaciones de formación certificadas por el IRCA, las que en su conjunto representan más de 5.000 cursos certificados por año.

El IRCA ofrece programas de certificación que reconocen la competencia de auditores que auditan sistemas de gestión de la calidad, ambiental, seguridad y salud ocupacional, seguridad en la información, seguridad en alimentos, seguridad marítima, desarrollo de software.

Muchos auditores completan los cursos certificados para cumplir con uno de los requisitos para la certificación IRCA como auditores. Otros eligen formación certificada por el IRCA porque ésta

representa una calificación de referencia aceptada por la industria para profesionales involucrados en el desarrollo, implementación y mantenimiento de sistemas de gestión.

La Universidad EAFIT es otro de los centros que brinda cursos de capacitación para la formación de Auditores Internos de Calidad. Tiene el objetivo de crear auditores idóneos para realizar un adecuado proceso de auditorías internas en el marco de los lineamientos de la norma internacional ISO 19011.

Esta Norma Internacional proporciona orientación sobre los principios de auditoría, la gestión de programas de auditoría, la realización de auditorías de sistemas de gestión de la calidad y auditorías de sistemas de gestión ambiental. La aplicación de esta Norma Internacional a otros tipos de auditorías es posible en principio, siempre que se preste especial atención a la identificación de la competencia necesaria de los miembros del equipo auditor.

Otros organismos como el Organismo Nacional de Normalización y Certificación (NORMEX), brinda la capacitación de un curso teórico práctico que comprende los principales aspectos relacionados con la planificación, ejecución y seguimiento de auditorías, así como los criterios para la calificación de auditores, para auditar sistemas de gestión de la calidad (SGC) y ambiental. Está dirigido al personal responsable del área de gestión de la calidad de las empresas que realicen actividades de auditor.

En cada capacitación de estos cursos se debe de tener un control estricto de quienes son los estudiantes ingresados, para esto hacen faltan software que te permitan llevar un seguimiento de los mismos.

Software para gestionar estudiantes en el mundo.

- ✓ **Centre:** Software de gestión de estudiantes diseñado para dirigir las necesidades más importantes de administradores, profesores, el personal de apoyo, padres, estudiantes, y el personal administrativo. Mediante este software los estudiantes pueden ser buscados por no sólo el género, la identidad étnica, el nivel del grado, etc., sino también según sus actividades escolares, ausencias, grados y la fila de clase. Permite la impresión de documentos con los datos de los resultados de búsquedas requeridos.
- ✓ **DocCF:** Software de gestión escolar de alumnos que permite hacer registro de matrículas, gestión de docentes, evaluación de docencia, asignación de horarios, administración de biblioteca, registro de calificaciones y generación de boletines escolares, envío de emails,

mensajes de texto, control de pagos y pensiones, inventario escolar, control de ausentismo, generador de consultas, informes personalizados y estadísticas.

Estos software no satisfacen las necesidades de gestión de matrículas en el programa FEA existentes en el ministerio. El DocCF esta dividido por módulos que facilita el trabajo con el mismo, algunos de los cuales son necesarios para la aplicación a realizar como matricular a un estudiante, pero presenta la dificultad de que además de ser un software propietario, opera en sistemas de Red Windows NT o Windows 2000/2003 Server en puestos de trabajo Windows 98/2000/XP/2003/Vista, usando base de Datos Microsoft Access, MS SQL Server o MySQL. El Centre es un software desarrollado en ambiente web, open source, pero para poder utilizarlo es necesario pagar una licencia.

Debido a estos inconvenientes no se pueden adaptar dichos software para la aplicación a realizar.

1.2 Formación de auditores en Cuba.

El desarrollo alcanzado en el control económico y administrativo, en la fiscalización superior desde la creación de la Oficina Nacional de Auditoría, como un órgano adscrito al Ministerio de Finanzas y Precios, determinó la creación del Ministerio de Auditoría y Control, como uno de los mayores centros creados para combatir las ilegalidades y la corrupción.

El país contó hasta el año 2007 con 3 564 auditores, todos graduados como resultado de la implementación del programa FEA surgido bajo el calor de la batalla de ideas. Este programa se puso en marcha en todas las provincias del país para formar la fuerza técnica y profesional que necesita emplear en la tarea de velar día y noche por el orden y el control de los recursos y bienes del Estado y la Revolución. Es un programa de captación a partir de técnicos medios en contabilidad y adiestrados universitarios.

Surgió con la idea de cubrir las necesidades de auditores que existen en el Sistema Nacional de Auditoría. Este sistema cuenta con unidades centrales de Auditorías Internas de los Organismos de la Administración Central del Estado (OAEC) y de los Consejos de Administración, unidades de auditorías de los niveles intermedios y auditores internos de las empresas del país. Debido a que la formación de auditores por vías convencionales abarca un período extenso, este programa alternativo

brinda la posibilidad de preparar a los especialistas en la referida disciplina en un menor plazo de tiempo.

Unido a la capacitación técnica y profesional, estos cursos inculcan en los estudiantes valores tales como la fidelidad a la Revolución, la honestidad y la modestia en su desempeño.

El objetivo fundamental de la FEA es llegar a tener en cada empresa y unidad presupuestada, un auditor interno. El reto que tienen estos jóvenes por delante es muy grande y según las palabras de nuestro Comandante en Jefe Fidel Castro:

«Es necesario mantener una conducta ética, ser honestos, sinceros, capaces de defender la verdad a cualquier precio y en cualquier circunstancia; ejercer la crítica y la autocrítica constructiva y oportuna; y, asumir en nuestros actos, el concepto de Revolución ».

Con el crecimiento de esta fuerza laboral se logrará un mayor impacto en las tareas de ordenamiento y control de la calidad del sistema de auditoría en Cuba.

Todo programa que relacione estudiantes, tiene vinculado dentro un sistema que permita llevar un control de los mismos. Actualmente se han desarrollado sistemas para la gestión universitaria en nuestro país, entre ellos se encuentran:

- **SAIDO:** Sistema Automatizado de Información Docente, uno de los primeros Sistemas diseñados e implantados por el Centro de Cibernética Aplicado a la Medicina (CECAM). A partir de su explotación se han tenido que incorporar nuevos módulos y subsistemas y elaborar otros sistemas relacionados con él, de acuerdo a las necesidades de los usuarios y al avance de las tecnologías computacionales como son :
 - **SAIDO-F:** Sistema automatizado de información docente. Se encuentra en explotación en todas las facultades del país. Es un sistema de base de datos que contiene y procesa información sobre los estudiantes de ciencias médicas. Incluye módulos de matrícula, actualización y notas. Salidas para información al Instituto y el Ministerio de Salud Pública (MINSAP) y permite obtener por selección listados, tablas y totales sobre los datos almacenados.
 - **SAIDO_FV:** Sistema de Información para Graduados: Está implantado en todas las

facultades del país. Surge debido a la necesidad de obtener la certificación del graduado con todas sus notas.

- **CE:** El Sistema de Curso Electivo. Surge debido a la necesidad de las facultades de contar con un sistema que de forma rápida y fácil les permitiera asignarles a los estudiantes de las distintas especialidades el curso que iban a recibir en el tiempo que tienen previsto para su asignatura Tiempo Electivo. El sistema recoge del SAIDO_F los datos generales de los alumnos agregándose a este los cursos que piden los estudiantes en sus diferentes opciones. Permite distribuir los cursos de acuerdo al promedio de cada estudiante.
- **SAIDO-I.:** Sistema Automatizado de Gestión docente para Institutos El sistema es actualizado con los datos de las facultades, las cuales mensualmente envían su información al Instituto, allí es consolidada y se envían al ministerio los modelos requeridos así como se brinda informaciones a los distintos niveles que la solicitan.
- **PREMED:** Sistema Automatizado de Información Docente para Estudiantes Extranjeros en el Sistema de Premédica. Surge con el objetivo de brindar información estadística sobre los estudiantes extranjeros que pasan su periodo de premédica en la facultad “Victoria de Girón” para después incorporarse a la escuela latinoamericana de Medicina.

Estos softwares que ya están implementados, no cubren las necesidades del sistema que se llevará a cabo en el ministerio para formar auditores emergentes, debido a esto es preciso diseñar un programa que supla las carencias de un software para la gestión de la Formación Emergente de Auditores.

1.3 Metodologías, técnicas y herramientas utilizadas para el desarrollo.

1.3.1. Metodología de desarrollo.

Todo desarrollo de software es riesgoso y difícil de controlar, para eso es necesario llevar a cabo una metodología que guíe con un conjunto de pasos y procedimientos el desarrollo el software.

Una metodología es la forma de dividir un proyecto en etapas, definiendo las tareas se llevan a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que se emplean especificando como se controla y gestiona un proyecto.

El sistema informático que se ha de realizar será un ambiente Web para contribuir a la gestión de los procesos de La Dirección de Atención al Sistema Nacional de Auditoría del Ministerio de Auditoría y Control en Cuba, y dentro de esta la Formación Emergente de Auditores. Este sistema informático tendrá como objetivo lograr un buen desenvolvimiento en el trabajo a realizar, alcanzar un buen desempeño en el trabajo y que satisfaga las necesidades del ministerio en cuanto a la realización de las matrículas de estudiantes al programa FEA.

Actualmente existen dos grupos en los cuales se dividen las metodologías de modelado, las metodologías tradicionales y los procesos ágiles. Dos ejemplos de estos grupos son las metodologías:

- Rational Unified Process (RUP).
- Extreme Programing (XP).

Extreme Programing (XP)

Extreme Programing (XP) es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo y pequeños equipos .La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

XP se basa en pruebas unitarias realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir, o sea, es obtener los posibles errores. Con la refabricación se hace una reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio. Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Por ser una metodología utilizada para proyectos de corta duración no se puede utilizar en la realización del trabajo por la gran envergadura que tiene el mismo. El equipo de trabajo del proyecto es variable y se necesita que los documentos generados sean específicos para la continuidad de la realización del mismo y XP es una metodología ágil que genera menos documentación y el usuario es parte del equipo de desarrollo por lo que no se hacen toma de requisitos.

Rational Unified Process (RUP)

Para el modelado del sistema se utilizará la metodología RUP debido a que este es un proceso que define claramente quien, cómo, cuándo y qué debe hacerse; y, como su enfoque esta basado en modelos utiliza un lenguaje bien definido para tal fin, el UML.

Aporta artefactos como los casos de uso, que se definen a partir de los requerimientos. Permite la ejecución iterativa del proyecto y del control de riesgos.

A través de un proyecto guiado por RUP, los requerimientos funcionales son expresados en la forma de casos de uso, que guían la realización de una arquitectura ejecutable de la aplicación. Además el proceso focaliza el esfuerzo del equipo en construir los elementos críticos estructuralmente y del comportamiento (llamados Elementos Arquitecturales) antes de construir elementos menos importantes. La mitigación de los riesgos más importantes guía la definición / confirmación del alcance en las primeras etapas del ciclo de vida. Finalmente RUP particiona el ciclo de vida en iteraciones que producen versiones incrementales de los ejecutables de la aplicación.

Principales características de RUP.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

- **Iterativo e Incremental:** Para facilitar la gestión un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias

iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

- **Guiado/Manejado por casos de uso:** La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios.
- **Centrado en arquitectura:** La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, gestores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales.

RUP define como lenguaje unificado de modelado UML (Unified Modeling Language) el cual está diseñado para visualizar, especificar, construir y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema. Los diagramas fundamentales que emplea UML son los siguientes:

- **Diagrama de casos de uso:** describe las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.
- **Diagrama de clases:** muestra las clases y las relaciones entre ellas.
- **Diagrama de secuencia:** muestra los objetos y sus múltiples relaciones entre ellos, además del intercambio de mensajes en un momento dado.
- **Diagrama de colaboración:** muestra objetos y sus relaciones, destacando los objetos que participan en el intercambio de mensajes.
- **Diagrama de implementación** muestra las instancias de los componentes y sus relaciones.

Para poder representar y describir los procesos del negocio se utilizó IDEF0, debido a que proporciona un marco de trabajo para poder representar y entender los procesos del negocio, determinando el

impacto de los diferentes sucesos y definiendo cómo los procesos interactúan unos con otros mediante flujos de información permitiéndonos identificar actividades poco eficientes o redundantes.

Un modelo IDEF0 se compone de una jerarquía de diagramas que permiten describir un sistema como una jerarquía de actividades o funciones y las relaciones existentes entre ellas.

Una actividad se compone mediante un rectángulo y una flecha saliente y una entrante que representan información, personas, lugares, cosas, conceptos, eventos. Se pueden descomponer a su vez en subactividades de mayor detalle, dando lugar a la típica estructura jerárquica de los diagramas IDEF0.

El IDEF0 utiliza unos grafos de visualización de sus elementos, no sólo para facilitar la aplicación del método, sino para diferenciar claramente las magnitudes a tratar en aplicaciones de software.

1.3.2. Herramientas de desarrollo

Para el desarrollo de esta aplicación web, se utilizará las herramientas relacionadas con el sistema operativo a utilizar, tanto para el desarrollo del modelado del sistema, como la aplicación web.

1.3.2.1. Herramienta de gestión de proyectos

Las herramientas a valorar son Dotproject y Trac. Ambas herramientas son open source, basadas en la web, multiusuario, multiplataforma, permiten la gestión de proyectos, tareas y usuarios, integración con LDAP (autenticación por usuarios del dominio), permiten la creación de diagramas de Gantt (el trac mediante un plugin, Dotproject lo tiene por defecto) y la gestión de documentos.

Se utilizará el Trac ya que proporciona integración con el servidor de código subversión, además de una wiki integrada y varias facilidades de reportes (lo cual no posee el Dotproject), así como un calendario (timeline) que muestra todos los eventos del proyecto, lo que permite ver el progreso del mismo de forma muy fácil.

1.3.2.2. Herramienta de gestión de configuración.

La herramienta propuesta es Subversion que es un controlador de versiones empleado en la administración de archivos utilizados en el desarrollo de software o contenido. Esta herramienta está diseñada para reemplazar CVS (Concurrent Version System), el cual posee varias deficiencias entre las cuales se pueden mencionar que es lineal $O(n)$ y trata los archivos binarios internamente como si fueran de texto. En CVS cada archivo versionado tiene un número de versión independiente.

Subversion tiene una fuerte integración con Apache lo que permite definir controles de acceso avanzados y navegación vía web para consultar el depósito de archivos. El repositorio presenta un único número de versión, lo que identifica todos los archivos del repositorio en cierto punto de tiempo. Subversion independientemente del número de ramificaciones creadas mantiene un árbol diferencial de cambios, minimizando así el espacio consumido en el depósito, creando copias diferenciales de archivos binarios.

Presenta una serie de herramientas gráficas (interfaces o clientes) que facilitan el trabajo con él entre las que se pueden mencionar:

- TortoiseSVN: la interfaz más popular diseñada para Windows.
- Subclipse: plugin que integra subversion al IDE eclipse.
- RapidSVN: interfaz diseñada para Linux
- KDESvn: muy parecido en apariencia, funcionamiento y características a TortoiseSVN (integración con entorno de escritorio KDE)
- SvnWorkbench: interfaz para Linux.

1.3.2.3. Herramienta CASE.

Las herramientas CASE (Computer Aided Software Engineering) o de modelado son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo del sistema de información que permiten el incremento en la velocidad de desarrollo de los sistemas. A los analistas, le permite, tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar, aumentando de esta forma la productividad, a través de la automatización de determinadas tareas, como la generación de código y la reutilización de objetos o módulos.

Las etapas del proceso de desarrollo de software permiten automatizar el dibujo de diagramas. Ayudar en la documentación del sistema y en la creación de relaciones en la base de datos y generar estructuras de código.

La herramienta propuesta para el modelado es Visual Paradigm ya que es una de las herramientas UML CASE muy completa y fácil de usar, presenta soporte multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación.

Visual Paradigm-UML proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

Presenta 2 tipos de diagramas de modelamiento de bases de datos: entidad-relación (ERD) y mapeo objeto relacional (ORM). Los diagramas ERD modelan la base de datos a nivel físico y los ORM muestran la relación entre las clases (orientado a objeto) y la entidad (de la base de datos). Esta característica permite generar además del script de la base de datos, el código de las clases persistente (o clases de entidad) en el lenguaje escogido.

Rational Rose es una herramienta CASE que presenta características que impiden su utilización entre las que se encuentra que es una herramienta la cual su sistema operativo apropiado es Windows 2000, Windows NT y Windows XP , o sea que no es multiplataforma, además de que es propietaria.

1.3.2.4. Herramienta de control, seguimiento y gestión de errores.

Las herramientas a considerar son bugzilla y Trac. Se propone utilizar Trac ya que está optimizado para subversion, siendo la interfaz web del subversion. Este provee una plataforma completa para colaboración entre usuarios y desarrolladores, con una interfaz de usuario más amigable e intuitiva, donde los errores son registrados como tareas. En el caso de bugzilla, los errores, los cambios, las peticiones de nuevas funcionalidades son registradas como defectos.

Trac crea una entrada en la wiki a cada cambio en el repositorio, con un seguimiento y documentación de errores, tareas y cambios. Tiene una integración con eclipse mediante subclipse y Mylyn, lo que permite asignar errores (archivos) a la tarea correspondiente, además de notificar al usuario de dichas tareas.

1.3.3. Sistema Operativo.

Para el desarrollo se decidió trabajar sobre Linux, ya que sus principales competidores como Mac OS requiere de una arquitectura de hardware específica (Apple), la cual es de alto presupuesto y Windows es un software propietario, no gratuito, el cual no se puede utilizar para el desarrollo de software al no poder adquirir la licencia requerida por estar sujetos a las leyes del embargo económico impuesto a nuestro país por los EE.UU.

De las distribuciones de Linux existentes se decidió optar por Ubuntu Gutsy Gibbon (7.10) para el cliente ya que presenta un entorno fácil de trabajar, con una amplia gama de software equivalente a los utilizados en Windows, documentación disponible en varios idiomas (entre ellas español), gran comunidad de usuarios, libre, gratis.

1.3.4. Plataforma de desarrollo

Una plataforma de desarrollo brinda soluciones necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Ofrecen un entorno y unas herramientas de desarrollo. En consecuencia, su interés radica en la posibilidad de construir soluciones adaptadas a cada caso.

Para el desarrollo de la aplicación se trabajará con la plataforma LAPP (Linux – Apache – PostgreSQL – PHP), debido a todas las posibilidades que brinda esta plataforma. Esta ensamblada en software libre y tiene una comunidad de usuarios grandes porque brinda soporte y ayuda.

1.3.5. Lenguaje(s) de programación.

Se sugiere utilizar PHP para la programación de las páginas servidoras, XML para la transferencia de datos por los servicios Web y JavaScript para la implementación de las funcionalidades en el cliente (AJAX).

PHP (Personal Home Page) es un lenguaje de programación que sirve principalmente para proporcionar características dinámicas a una página web. Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas donde una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre las bases de datos que soporta PHP pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL. Ofrece también la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. Junto con Apache, MySQL y Linux forma la famosa LAMP, uno de los principales responsables del desarrollo de la web 2.0.

PHP es multiplataforma, open source, delega todo lo ajeno al procesamiento de código al servidor web, presenta una Infinidad de librerías para el trabajo con imágenes, correo, redes, xml, servicios web, matemática, etc., así como también una gran cantidad de framework que facilitan el trabajo del desarrollador web.

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. Es un formato que permite la lectura de datos a través de diferentes aplicaciones que sirve para estructurar, almacenar e intercambiar información.

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Es un lenguaje interpretado que no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente. Mediante este se pueden desarrollar scripts que ejecuten acciones en respuesta a estos eventos. El modelo de objetos está reducido y simplificado, pero incluye los elementos necesarios para que los scripts puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

1.3.6. Entorno(s) de Desarrollo Integrado.

Integrated Development Environment (entorno de desarrollo integrado IDE), es un editor de código que se utiliza para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación. Un mismo IDE puede funcionar con diferentes lenguajes de programación, un buen ejemplo de ello es Eclipse.

Se utilizará el eclipse con el plug-in PDT (plug-in para programar en PHP con el IDE Eclipse). Este presenta un entorno amigable, resaltado de sintaxis, completamiento de código, alta integración con el framework Symfony, gran cantidad de plugin que le extienden la funcionalidades tales como Aptana (para el desarrollo con AJAX), subclipse para integración con subversion, Mylyn para integración con trac, que hace al eclipse una herramienta de desarrollo potente.

1.3.7. Framework(s), componente(s) y/o librería(s).

Se usará el framework Symfony como librería base para la programación del proyecto. Es un framework programado en PHP5, en el que su desarrollo se basa en el patrón MVC, lo que crea una independencia entre las capas del desarrollo.

Utiliza Propel para el acceso a bases de datos, dando de esta forma transparencia con respecto al gestor de bases de datos utilizado, lo que permite general el esquema de clases persistentes a partir de la base de datos o realizar la operación inversa. Utiliza Lime para la realización de prueba unitarias y funcionales al código, permitiendo tener una menor cantidad de errores y realizar un desarrollo basado en pruebas. Puede utilizar componentes de otros framework de PHP y exhibe una documentación abundante en varios idiomas (entre ellos español) con una comunidad de usuarios activa y grande y una disponibilidad de un gran número de plugins, dando también la posibilidad de crear el nuestro.

Se propone además utilizar el framework **Extjs** para el trabajo con AJAX, aparte de que el symfony trae las librerías prototype y scriptaculous.

Entre los componentes que esta librería ofrece encontramos cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos desarrollos al estilo de Web 2.0.

1.3.8. Gestor de base de datos

PostgreSQL es el gestor propuesto para el desarrollo del proyecto. Está considerado como la base de datos de código abierto más avanzada del mundo que proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays. Presenta una alta concurrencia mediante acceso concurrente multiversión (MVCC), lo que permite a un proceso escribir y a otros acceder a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al bloqueo por tablas o por filas común en otros gestores (MySQL por ejemplo), eliminando la necesidad del uso de bloqueos explícitos. Permite la herencia entre tablas, lo que facilita el desarrollo orientado a objetos. Es rápido y seguro, soporta vistas, triggers, cursores, claves ajenas, consultas anidadas o subselect y es multiplataforma.

PostgreSQL soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario. Tolera integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

1.3.9. Servidor de aplicaciones y/o web

Apache es el programa servidor HTTP. Gracias a él podemos practicar la creación y publicación de documentos php con una estabilidad y eficacia ampliamente comprobada en la gran cantidad de servidores apache actualmente en uso.

Es un servidor de web multiplataforma, flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1) que implementa los últimos protocolos, aunque se base en el HTTP / 1.1. Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos. Se desarrolla de forma abierta y por ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Como se puede apreciar en el gráfico, el 50% de los sitios de internet lo usan como servidor.

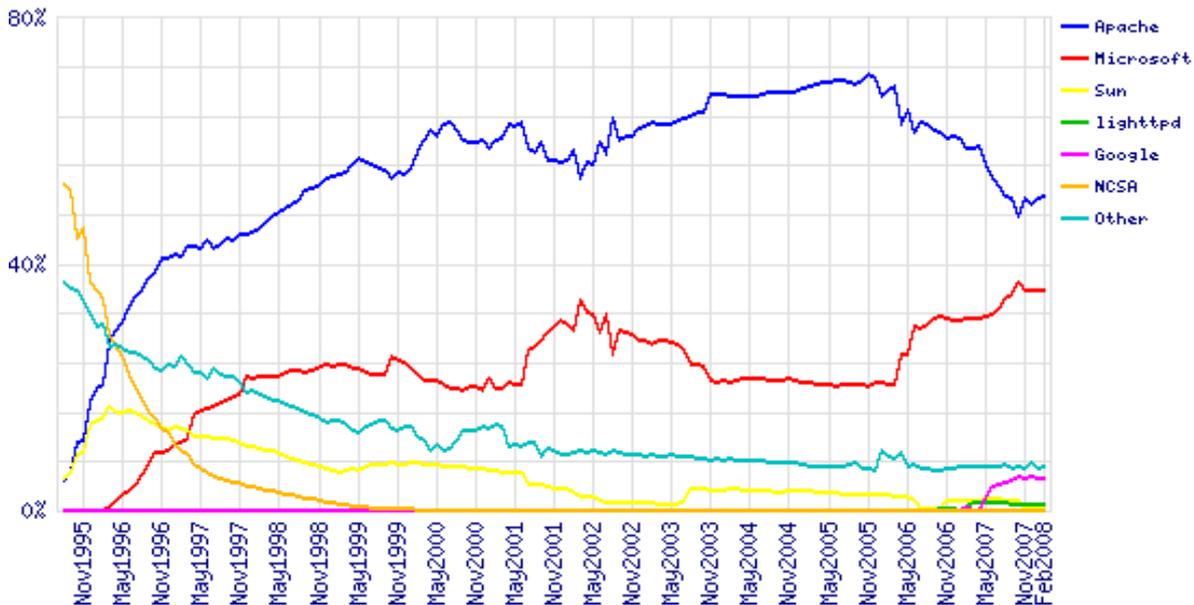


Fig. 1.2. Usos del Servidor apache

Este servidor presenta un gran número de módulos tales como

- **mod_ssl** – Comunicaciones Seguras vía TLS.
- **mod_rewrite** – reescritura de direcciones.
- **mod_dav** – Soporte del protocolo WebDAV.
- **mod_auth_ldap** – Permite autenticar usuarios contra un servidor LDAP.
- **mod_php** – Páginas dinámicas en PHP.
- **mod_security** – Filtrado a nivel de aplicación, para seguridad.

Su flexible sistema modular, permite cargar y descargar módulos sin necesidad de tocar el kernel.

1.4. Conclusiones

En este capítulo se abordaron los conceptos y aspectos generales para un mejor entendimiento del problema, además de las herramientas, lenguajes y servidor web a utilizar para el desarrollo de la aplicación, donde cada uno tiene un explicación argumentada del porqué de su elección.

La aplicación Web que se proveerá es la solución propuesta por los autores para resolver los problemas existentes en la Formación Emergente de Auditores de la Dirección de Atención al Sistema

Nacional de Auditoría del Ministerio de Auditoría y Control. Haciendo referencia demás en software implementados, se llegó a la conclusión de que ninguno de ellos abastecen las necesidades existentes en el Ministerio actualmente.

CAPÍTULO 2: CARATERÍSTICAS DEL SISTEMA

Los procesos de gestión de la Formación Emergente de Auditores (FEA) en la Dirección de Atención al Sistema Nacional de Auditoría del Ministerio de Auditoría y Control es el objeto de estudio de este Trabajo de Diploma. Este tiene la finalidad de encontrar una solución para ayudar a la FEA a realizar un mejor trabajo y así satisfacer las necesidades de los trabajadores del Ministerio.

En este capítulo se hace una breve valoración de los objetivos estratégicos de la organización y procesos de negocios que los soportan; se describen los sistemas automatizados que existen en la empresa y que estén vinculados al flujo actual de los procesos involucrados en el campo de acción, además de la crítica a los mismos y la descripción de cuales de ellos serán objeto de automatización. El sistema propuesto es descrito de forma general y se especifican los requisitos que debe cumplir el software así como debe de funcionar.

2.1. Problema y situación problemática.

2.1.1. Objetivos estratégicos de la organización

El Ministerio de Auditoría y Control es uno de los organismos más importantes para el Estado ya que juega un papel fundamental en el control de las disciplinas y las irregularidades de las ilegalidades. Este ministerio cuenta con direcciones para el buen funcionamiento del mismo, como son la Dirección de Atención al Sistema Nacional de Auditoría, Dirección de Auditorías y Controles Especiales, Dirección de Planificación, Análisis y Control y Dirección de Atención a la Ciudadanía.

Una de las tareas de la Dirección de Atención al Sistema Nacional de Auditoría (DASNA) es la organización, control y seguimiento de los estudiantes del Programa de Formación Emergente de Auditores (FEA). Este programa tiene como objetivo contar con un auditor interno en cada empresa o unidad presupuestada del país. El programa FEA surge por la necesidad que presenta actualmente el país de nuevos auditores pues se tiene un déficit creciente de los mismos. Esta estrategia además contribuye a que el país gane fuerzas en el campo de la auditoría.

La FEA tiene procesos fundamentales como son:

- Matricular estudiantes en FEA.
- Elaborar informe general de estudiantes matriculados en FEA.

- Matricular estudiantes en las Sedes Universitarias Municipales (SUM), una vez terminado un curso, pasar los estudiantes de año.
- Dar de baja a un estudiante.
- Elaborar informe de estudiantes que son baja.
- Mostrar estudiantes por año de estudio en SUM.

Todos estos procesos se llevan a cabo manualmente lo que dificulta la agilidad de los mismos y el tiempo de respuesta, por lo que se puede dar el caso de la pérdida de información en algún momento. La Dirección de Atención al Sistema Nacional de Auditoría (DASNA) tomó como estrategia para resolver dichos problemas automatizar los principales procesos que se llevan a cabo en la FEA para que de esta forma se ganara en tiempo y eficiencia a la hora de buscar información de estudiantes, elaborar informes y con ello se reduzca la pérdida de información.

2.1.2. Flujo actual de los procesos involucrados en el campo de acción.

Para matricular a los estudiantes en FEA es necesario que los estudiantes sean graduados universitarios en contabilidad o que estén en 4to año del técnico medio de esta especialidad. Una vez matriculados los estudiantes en la FEA, los funcionarios de la delegación provincial proceden a realizar un informe por provincia de los estudiantes que son matriculados, enviándolos más tarde a los técnicos del DASNA para la confección del informe general de los estudiantes matriculados en FEA.

Cuando el estudiante sale graduado del curso de capacitación de FEA, el funcionario de la delegación provincial le asigna una pre-ubicación laboral ocupando una plaza como auditor. Este programa brinda la posibilidad al estudiante que no posea nivel superior, de continuar superándose en las sedes universitarias de su municipio y una vez que culmine sus estudios se incorpora a la pre-ubicación laboral que se le fue asignada. En este caso hay que tener en consideración la disposición del estudiante a continuar sus estudios en las SUM.

Si el estudiante decide ser baja de la Formación Emergente de Auditores, pierde la matrícula en FEA y en SUM en caso de estar matriculado en la misma. Cuando se termina un curso, se procede a pasar al estudiante de año.

Una vez que el estudiante culmine los estudios en el programa FEA, sus datos son almacenados en un informe histórico de graduados. Cuando el estudiante se gradúa y comienza a ejercer como auditor, pasa a ser un habilitado en la profesión y sus datos son guardados en un informe de habilitados.

Cada vez que se realizan las matrículas los técnicos del DASNA confeccionan un Modelo CH5 donde se guardan los datos de los estudiantes y de control general del Programa FEA. Lo mismo ocurre cuando se le da de baja a un estudiante, del informe de estudiantes que son baja de la FEA enviado por la Delegación Provincial y el modelo CH5 que guarda los estudiantes matriculados en la SUM, el Técnico de DASNA procede a dar de baja a un estudiante de la SUM y genera un modelo CH5 actualizado (con la palabra baja) y un modelo CH5A que contiene los estudiantes que no ratifican matrícula en SUM.

Cada vez que culmina un curso, se procede a la confirmación de matrícula y a la re-matrícula. La confirmación de matrícula es para aquellos estudiantes que culminan un año del curso FEA y proceden a matricularse en SUM, y la re-matrícula es realizada a aquellos estudiantes que se encuentran en SUM y van a pasar de año. Después de que se realiza cada acción es confeccionado un informe con los datos de los estudiantes matriculados que le haya sido confirmada la matrícula y la re-matrícula.

2.1.3. Análisis crítico de cómo se ejecutan actualmente esos procesos

En la actualidad, se llevan a cabo procesos para la realización de cada matrícula de estudiantes o de la baja de los mismos, donde después de cada acción se realiza la confección de informes los cuales son posteriormente archivados. Entre los informes tenemos el informe de matrícula de estudiantes en FEA por provincia (Modelo CH1), donde se guardan en una plantilla, todos los datos de los estudiantes matriculados en la FEA en cada provincia del país. Posteriormente se confecciona el Informe general de matrículas en FEA en el que se agregan todos los datos de los estudiantes matriculados en el país mediante los demás documentos ya enviados por provincia.

En el informe de baja de estudiantes en FEA, así como también en el informe de baja de estudiantes en SUM, se guardan todos los datos de los estudiantes que hayan sido bajas del curso emergente respectivamente. Cuando el estudiante culmina sus estudios en FEA posteriormente se efectúa la graduación del mismo y se realiza el informe histórico de graduados en FEA, además de un informe de los estudiantes habilitados después que culminan sus estudios. Estos estudiantes habilitados son los que se gradúan en el curso FEA y se ubican en su respectiva pre-ubicación laboral.

El informe de confirmación de matrículas y re-matrículas es confeccionado cuando se realizan dichas acciones para llevar un control de los estudiantes que sigan en el curso FEA.

Estos procesos se hacen manualmente por lo que se hace casi imposible tener un control de todos los datos de los estudiantes guardados en estos documentos lo que trae como consecuencia que pueda existir alguna pérdida de los mismos.

2.2. Procesos que serán objeto de automatización.

Los procesos que serán objeto de automatización son aquellos que guardarán algún tipo de información referente a las acciones que se realizan a la hora de ejecutar cualquier actividad en la FEA, para ellos tenemos la realización de los informes de matrículas por provincia, los Modelos CH5, el informe de baja de estudiantes en FEA, el informe histórico de graduados y el informe de habilitados, así como también los informes de confirmación de matrículas y re-matrículas de los estudiantes.

La información de todos estos procesos se almacenará en una base de datos, de tal forma que los técnicos del DASNA tengan posibilidad de buscar información acerca de cada uno de los estudiantes en caso necesario, lo que reduce el grado de error que se puede presentar en los datos del estudiante.

2.3. Propuesta de sistema.

El Ministerio de Auditoría y Control no cuenta en la actualidad con un sistema que le permita la automatización de los procesos de búsqueda y almacenamiento de la información. Actualmente los técnicos del DASNA son los encargados de realizar estos procesos de matrículas manualmente, presentando inconformidades a la hora de realizar su trabajo. Todo esto condujo a la realización de un sistema Web automatizado, que lleve a cabo el perfeccionamiento de un sistema que brinde una solución real a todos los problemas existentes. Este ofrecerá opciones que posibiliten poder matricular a los estudiantes, modificar y eliminar sus datos, además de guardar los datos en bases de datos para después crear los informes solicitados con la opción de imprimirlos.

Será una interfaz sencilla y de fácil manipulación, pero a la vez segura y confiable.

2.4. Modelo del negocio

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos.

De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

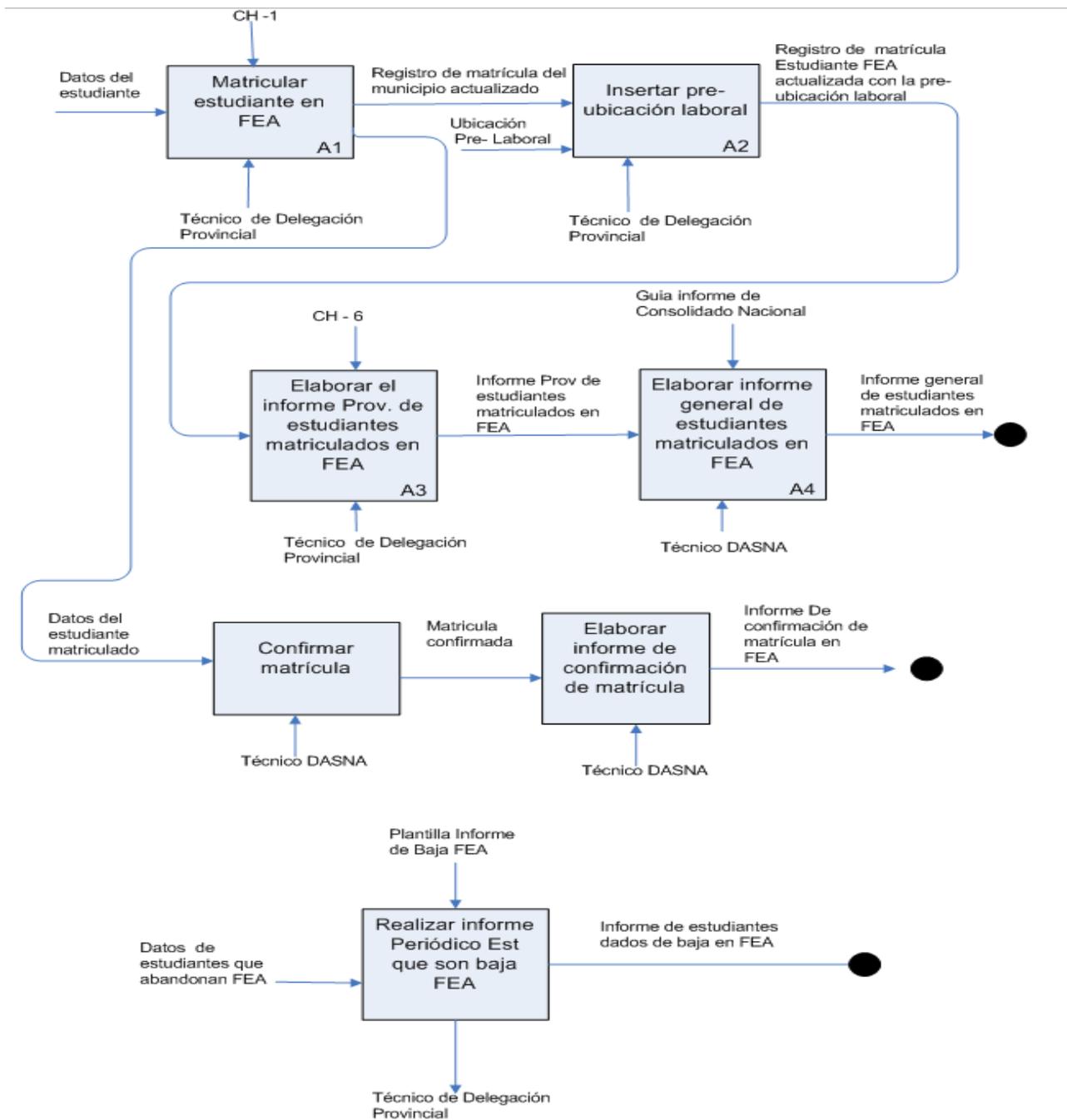


Fig. 2.1. Diagrama de descripción de los procesos de FEA

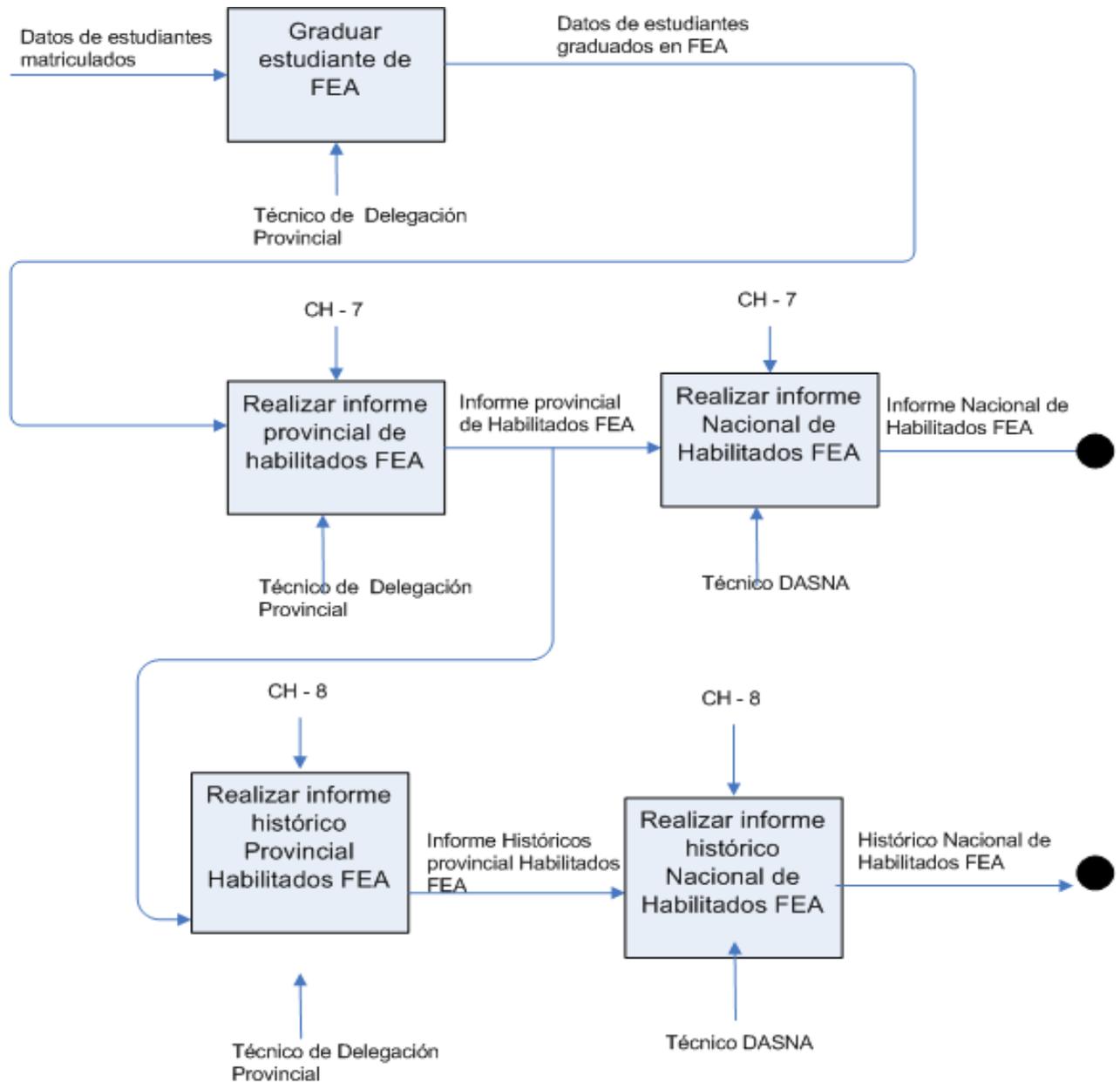


Fig. 2.2. Diagrama de descripción de los procesos de FEA (Cont.)

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

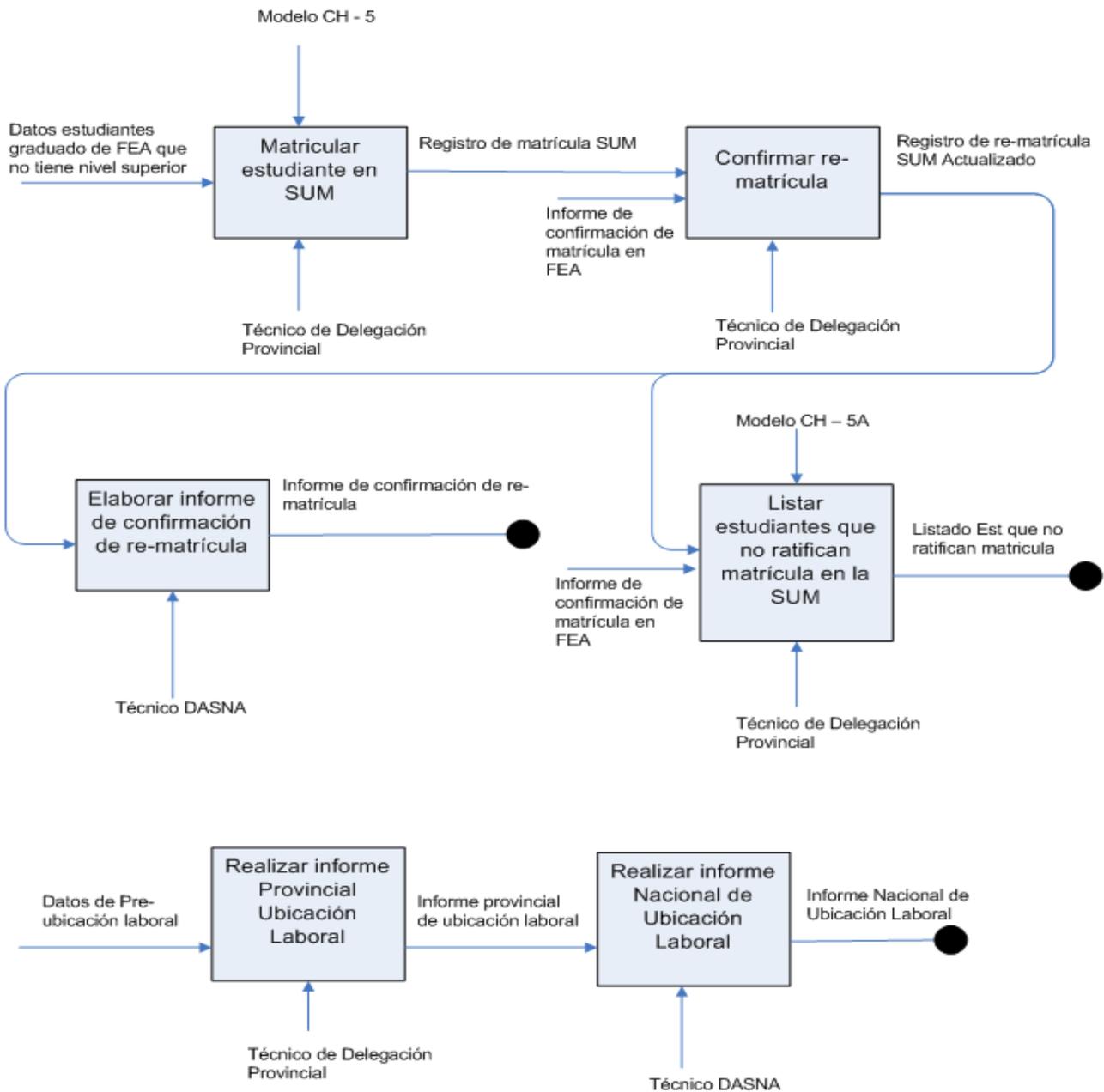


Fig. 2.3. Diagrama de descripción de los procesos de FEA (Cont.)

2.4.1. Especificación de los requisitos de software.

2.4.1.1. Requisitos Funcionales

Como los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, se escogió una serie de requisitos que nos permitirán entender más aún el sistema a automatizar.

RF1. Matricular estudiante en FEA.

RF2. Dar de baja a un estudiante de curso FEA.

RF3. Ver detalles de los estudiantes.

RF4. Elaborar informe general de estudiantes matriculados en FEA.

RF5. Elaborar informe de habilitados en FEA por provincia.

RF6. Elaborar informe nacional de habilitados.

RF7. Elaborar informe general de totales de habilitados por provincia.

RF8 .Elaborar informe general de totales de matrícula por provincia.

RF9 .Modificar datos de estudiantes matriculados en FEA.

RF10. Elaborar informe de estudiantes que son baja de FEA.

RF11. Graduar estudiante de FEA

RF12. Informe históricos de graduados por curso FEA (CH-8).

RF13. Matricular estudiante en SUM

RF14. Pasar los estudiantes de año en SUM.

RF15. Mostrar reporte de estudiantes en cada año de estudio.

RF16. Dar de baja a un estudiante de SUM.

RF17. Informe de estudiantes que son baja por año en SUM.

RF18. Informe históricos de graduados por organismo (CH-9).

RF19. Elaborar informe provincial de ubicación laboral.

RF20. Elaborar informe nacional de ubicación laboral.

RF21. Confirmar matrícula de estudiante en FEA

RF22. Confirmar re-matrícula de estudiantes en SUM

RF23. Elaborar informe de confirmación de re matrícula

RF24. Elaborar informe de confirmación de matrícula

2.4.1.2. Requisitos no Funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

- **Apariencia o Interfaz Externa:**

- Diseño sencillo, permitiendo la utilización del sistema sin mucha experiencia.

- No debe utilizarse tecnología de frames.

- Cada página no debe exceder los 500 KB en imágenes.

- **Usabilidad:**

- Garantizar un acceso fácil y rápido a los usuarios. El subsistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web en sentido general.

- **Rendimiento:**

- Garantizar tiempos de respuestas generalmente rápidos al igual que la velocidad de procesamiento de la información.

Las páginas Web no deben tardar más de 15 segundos en cargarse.

El sistema requerirá no más de 3 segundos para mostrar una página estática al cliente.

- **Soporte:**

Se requiere de la instalación de un servidor Web con funcionalidades relacionadas con el manejo de la Base de Datos con un gestor PostgreSQL y el servicio de interpretación de códigos con PHP 5. Por su parte los clientes requerirán de un navegador capaz de interpretar secuencias de códigos IE6 o Firefox.

- **Seguridad:**

Garantizar que la información sensible solo pueda ser vista por los usuarios con el nivel de acceso autorizado para ello. Garantizar que las funcionalidades del subsistema se muestren de acuerdo al nivel de usuario que esté activo. El sistema debe contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Cumplir con la confidencialidad de los datos, permitiendo que cada auditor solo pueda ver sus datos.

- **Políticos-culturales:**

Emplear en la aplicación idioma español. Contar con logotipos e imágenes que se encuentren en correspondencia con el carácter científico y profesional del tema. Algún cambio que se quiera realizar debe ser previamente consultado con la Dirección Auditorías y Comprobaciones Especiales del Ministerio de Auditoría y Control y canalizado por la UCI con la dirección de Producción central y de la Facultad a la que pertenece el proyecto.

- **Legales:**

Publicar en el Ministerio de Auditoría y Control como una herramienta para la ayuda en el trabajo del Ministerio. Se estará usando una herramienta de software libre, bajo la licencia GNU/GPL.

- **Confiabilidad:**

Validar la captación de datos en el caso de los auditores al inscribirse para evitar entradas inadecuadas.

- **Hardware:**

PC servidor: 160 GB disco duro, microprocesador de 3.00 GHz, 512 MB mínimo de RAM, 1GB máximo de RAM.

2.4.2. Actores del sistema

Los actores no son ninguna parte del sistema, ellos representan a cualquiera o algo que debe interactuar con el sistema. Un actor puede que sólo brinde información de entrada al sistema y reciba la respuesta del mismo.

Actores del Sistema	Justificación
Técnico de la delegación provincial	Es el encargado de realizar todas las acciones referentes a la matrícula de los estudiantes en sus respectivos municipios, confeccionando los documentos de matrícula y baja.
Técnico del DASNA	Es el encargado de recepcionar los datos enviados por el técnico de la delegación provincial y confeccionar los documentos generales de cada matrícula de estudiantes.

2.4.3. Definición de los casos de uso

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

Una relación es una conexión entre los elementos del modelo, por ejemplo la relación y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar como reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagrama intervienen algunos conceptos nuevos: un actor es una entidad

externa al sistema que se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema. Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un caso de uso extiende otro caso.
- Un actor se comunica con un caso de uso.
- Un caso de uso usa otro caso de uso.

2.4.3.1. Descripción de los casos de uso

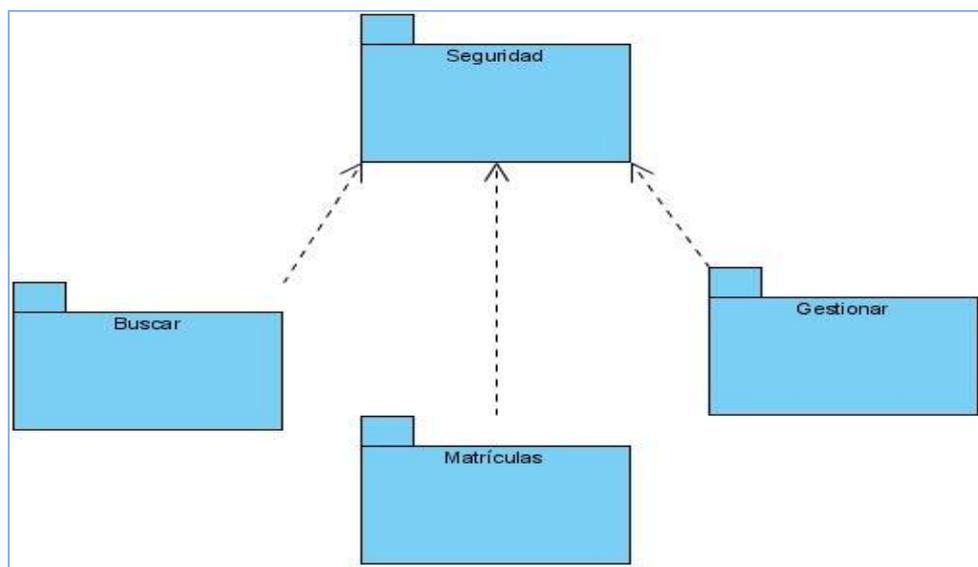


Fig. 2.4. Diagrama de paquetes

Diagrama de casos de uso del sistema

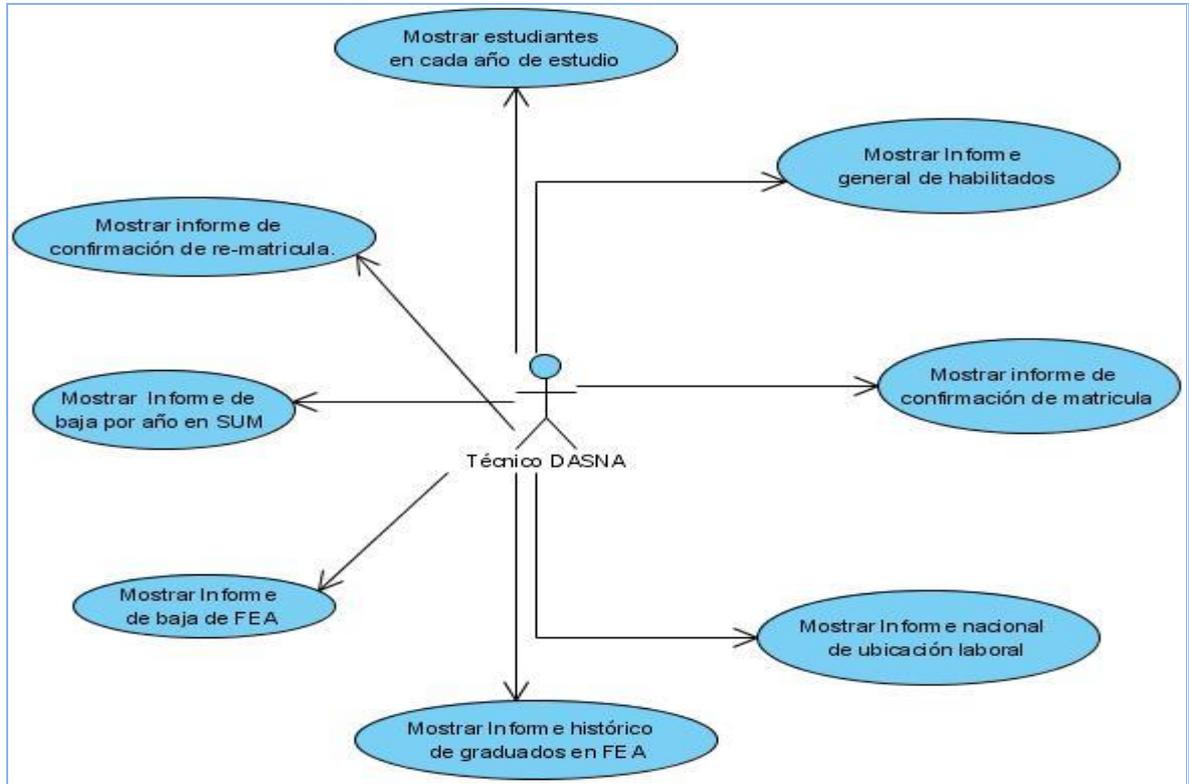


Fig. 2.5. Paquete Buscar

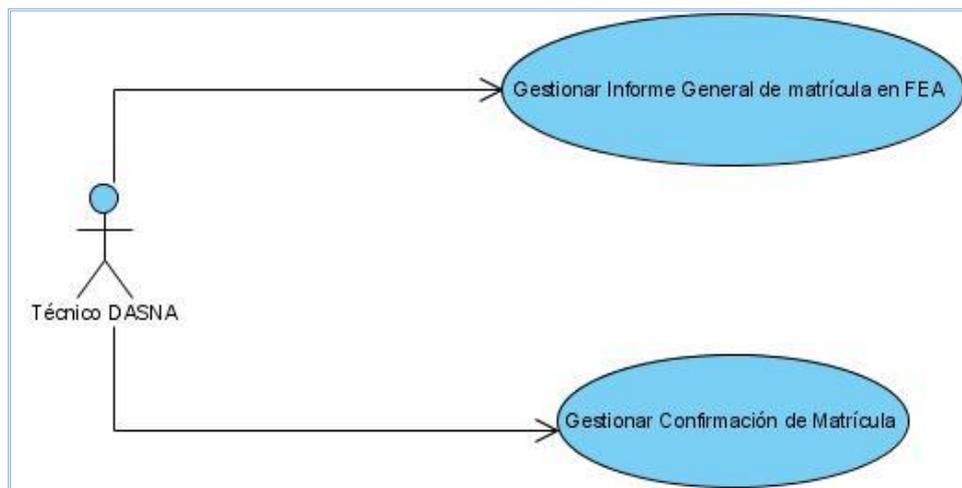


Fig. 2.6. Paquete Gestionar

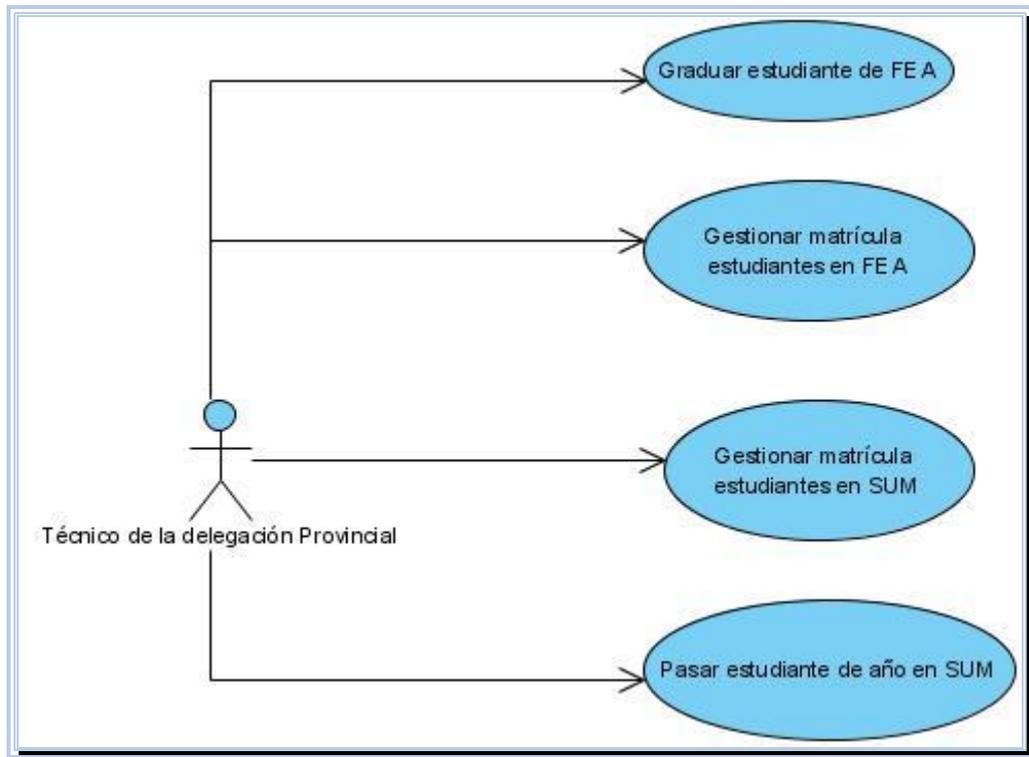


Fig. 2.7. Paquete Matrículas

Especificación de los Casos de Uso

A continuación se mostrará la descripción de algunos de los casos de uso, de los cuales son los clasificados como críticos. Para ver los demás ver Anexo #1

CU. Gestionar matrícula estudiante en FEA

Caso de Uso:	Gestionar matrícula estudiante en FEA
Propósito:	Brindar al técnico provincial la posibilidad de adicionar, eliminar o modificar datos de los estudiantes matriculados en FEA.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Actores:	Técnico de la delegación provincial	
Resumen:	Este caso de uso se inicia cuando el técnico provincial solicita adicionar, eliminar o modificar la matrícula de un estudiante. Si selecciona adicionar estudiante, el sistema solicita todos los datos, los almacena y actualiza los datos en la tabla estudiantes en el servidor de bases de datos. Si selecciona eliminar, el sistema muestra un listado de los estudiantes existentes mediante un criterio de búsqueda especificado. Selecciona al estudiante que desea eliminar, se confirman sus datos y se elimina de la base de datos. Si selecciona modificar un estudiante, el sistema muestra un listado de los estudiantes existentes mediante un criterio de búsqueda especificado. Selecciona al estudiante que desea modificar los datos, realiza la acción correspondiente, verifica los datos y los almacena en el servidor de bases de datos.	
Precondiciones:	El sistema debe mostrar los estudiantes matriculados en FEA para poder seleccionarlos a la hora de eliminarlos o modificar sus datos.	
Referencias	RF1,RF2, RF3, RF9	
Prioridad	Crítico	
Flujo Básico		
Acción del Actor	Respuesta del Sistema	
10.El técnico provincial escoge la opción “Matrícula de estudiante en FEA”.	<p>10.El sistema muestra un listado con todos los estudiantes matriculados en FEA con las opciones de :</p> <p>a) Adicionar se va a la sección “Adicionar Estudiante”.</p> <p>b) Eliminar se va a la sección “Eliminar Estudiante”.</p> <p>c) Modificar se va a la sección “Modificar</p>	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

datos del Estudiante”.

d) Detalles se va a la sección “Detalles del estudiante”.

Muestra un formulario con criterios de búsquedas y la opción

- Buscar

Prototipo de Interfaz de Usuario

Gestionar Matricula en FEA

Nombre: Provincia:

Apellidos: Curso FEA:



Nombre y Apellidos	Curso Fea	Provincia
Julio Cesar Sanz Medrano	2007-2008	Las Tunas
Pedro Perez Gil	2007-2008	Guantánamo
Juan Marichal Pupo	2006-2007	S: Spiritus
Margarita Garcia Gonzalez	2006-2007	Granma
Martolome Ibarra Castillo	2006-2007	Holguín
Miguel Barrientos González	2005-2006	S. Cuba
Laura Balsameda Puig	2007-2008	V. Clara
Maria Caridad Pi	2006-2007	Matanzas
Luis Angel Santos	2007-2008	I. juventud

Sección “Adicionar Estudiante”

Acción del Actor	Respuesta del Sistema
3. El técnico provincial selecciona la opción Adicionar.	4. El sistema muestra un formulario solicitando la información necesaria para insertar el estudiante:

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<ul style="list-style-type: none"> a) Nombre. b) Apellidos c) Carnet de identidad. d) Dirección particular. e) Curso FEA. f) Municipio. g) Provincia. h) Sexo. i) Categoría j) Nivel escolar k) Quiere SUM l) Fecha de matrícula <p>Y las opciones de</p> <ul style="list-style-type: none"> a) Aceptar b) Cancelar
5. El técnico provincial llena los campos y selecciona la opción "Aceptar"	6. El sistema verifica que los datos entrados sean correctos.
	7. El sistema inserta la ejecución y muestra un mensaje "Matrícula realizada con éxito".
Flujos Alternos 5a"El técnico provincial selecciona la opción Cancelar"	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	5a1.Retorna a 3.																
Flujos Alternos 6a”Los datos introducidos no son correctos”																	
Acción del Actor	Respuesta del Sistema																
	6a1. El sistema indica donde están los errores. Retorna a 5.																
Prototipo de Interfaz de Usuario																	
<div style="border: 1px solid gray; padding: 10px;"> <p>Matricular estudiantes en FEA</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Nombre <input style="width: 90%;" type="text"/></td> <td style="width: 50%;">Sexo <input style="width: 90%;" type="text" value="--Seleccione--"/></td> </tr> <tr> <td>Primer Apellido <input style="width: 90%;" type="text"/></td> <td>Nivel Escolar <input style="width: 90%;" type="text" value="--Seleccione--"/></td> </tr> <tr> <td>Segundo Apellido <input style="width: 90%;" type="text"/></td> <td>Quiere SUM <input style="width: 90%;" type="text" value="--Seleccione--"/></td> </tr> <tr> <td>C. Identidad <input style="width: 90%;" type="text"/></td> <td>Categoría</td> </tr> <tr> <td>Dirección <input style="width: 90%;" type="text"/></td> <td><input checked="" type="radio"/> Estudiante <input checked="" type="radio"/> Trabajador</td> </tr> <tr> <td>Curso FEA <input style="width: 90%;" type="text" value="--Seleccione--"/></td> <td>Politécnico <input style="width: 90%;" type="text"/></td> </tr> <tr> <td>Provincia <input style="width: 90%;" type="text" value="--Seleccione--"/></td> <td>Ocupación <input style="width: 90%;" type="text"/></td> </tr> <tr> <td>Municipio <input style="width: 90%;" type="text" value="--Seleccione--"/></td> <td style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </td> </tr> </table> </div>		Nombre <input style="width: 90%;" type="text"/>	Sexo <input style="width: 90%;" type="text" value="--Seleccione--"/>	Primer Apellido <input style="width: 90%;" type="text"/>	Nivel Escolar <input style="width: 90%;" type="text" value="--Seleccione--"/>	Segundo Apellido <input style="width: 90%;" type="text"/>	Quiere SUM <input style="width: 90%;" type="text" value="--Seleccione--"/>	C. Identidad <input style="width: 90%;" type="text"/>	Categoría	Dirección <input style="width: 90%;" type="text"/>	<input checked="" type="radio"/> Estudiante <input checked="" type="radio"/> Trabajador	Curso FEA <input style="width: 90%;" type="text" value="--Seleccione--"/>	Politécnico <input style="width: 90%;" type="text"/>	Provincia <input style="width: 90%;" type="text" value="--Seleccione--"/>	Ocupación <input style="width: 90%;" type="text"/>	Municipio <input style="width: 90%;" type="text" value="--Seleccione--"/>	<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>
Nombre <input style="width: 90%;" type="text"/>	Sexo <input style="width: 90%;" type="text" value="--Seleccione--"/>																
Primer Apellido <input style="width: 90%;" type="text"/>	Nivel Escolar <input style="width: 90%;" type="text" value="--Seleccione--"/>																
Segundo Apellido <input style="width: 90%;" type="text"/>	Quiere SUM <input style="width: 90%;" type="text" value="--Seleccione--"/>																
C. Identidad <input style="width: 90%;" type="text"/>	Categoría																
Dirección <input style="width: 90%;" type="text"/>	<input checked="" type="radio"/> Estudiante <input checked="" type="radio"/> Trabajador																
Curso FEA <input style="width: 90%;" type="text" value="--Seleccione--"/>	Politécnico <input style="width: 90%;" type="text"/>																
Provincia <input style="width: 90%;" type="text" value="--Seleccione--"/>	Ocupación <input style="width: 90%;" type="text"/>																
Municipio <input style="width: 90%;" type="text" value="--Seleccione--"/>	<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>																
Sección “Eliminar Estudiante”																	
3. El técnico provincial escoge los criterios de búsqueda o combinación de los mismos para buscar al estudiante que desea eliminar y selecciona la opción “Buscar”.	4. El sistema muestra los estudiantes matriculados que corresponden con el criterio de búsqueda.																
5. El técnico provincial escoge al estudiante al que	6. El sistema muestra un mensaje de confirmación																

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

desee eliminar y selecciona en el menú la opción - Eliminar.	para poder eliminar con la opción a) Aceptar. b) Cancelar.
7. El técnico provincial selecciona la opción "Aceptar".	8. El sistema elimina al estudiante y lo guarda en una tabla de bajas.
Flujos Alternos 4a" El sistema no encontró los estudiantes"	
Acción del Actor	Respuesta del Sistema
	4 ^a 1. El sistema muestra que no se encontraron estudiantes según los criterios de búsqueda. Retorna a 3.
Flujos Alternos 7a" El técnico provincial selecciona la opción Cancelar"	
Acción del Actor	Respuesta del Sistema
	7a1. El sistema cierra el mensaje y retorna a 3.
Flujos Alternos 8a" El sistema no pudo eliminar al estudiante "	
Acción del Actor	Respuesta del Sistema
	8a1. El sistema muestra los errores existentes y retorna a 3.
Prototipo de Interfaz de Usuario	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Gestionar Matrícula en FEA

Nombre: Provincia: --Seleccione-- v

Apellidos: Curso FEA: --Seleccione-- v

Buscar

	Apellidos	Curso Fea	Provincia
Eliminar	Julio Cesar Sanz Medrano	2007-2008	Las Tunas
	Pedro Perez Gil	2007-2008	Guantánamo
	Juan Marichal Pupo	2006-2007	S: Spiritus
	Margarita Garcia Gonzalez	2006-2007	Granma
	Martolome Ibarra Castillo	2006-2007	Holguín
	Miguel Barrientos González	2005-2006	S. Cuba
	Laura Balsameda Puig	2007-2008	V. Clara
	Maria Caridad Pi	2006-2007	Matanzas
	Luis Angel Santos	2007-2008	I. juventud

Anterior
<<
1,2,3,4,5,6,7,8,9,10
>>
Siguiete

Sección “Modificar datos del Estudiante”

Acción de autor	Respuesta del sistema
<p>3. El técnico provincial escoge los criterios de búsqueda o combinación de los mismos para buscar al estudiante y selecciona la opción “Buscar”</p>	<p>4. El sistema muestra los estudiantes matriculados que corresponden con el criterio de búsqueda.</p>
<p>5. El técnico provincial escoge al estudiante que desee modificarle los datos y selecciona la opción “Modificar”.</p>	<p>6. El sistema muestra un formulario que los datos del estudiante en forma editable:</p> <ul style="list-style-type: none"> a) Nombre b) Apellidos c) Carnet de identidad. d) Dirección particular.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	<ul style="list-style-type: none"> e) Curso FEA. f) Municipio. g) Provincia. h) Sexo. i) Categoría j) Nivel escolar k) Politécnico. <p>Con la opción</p> <ul style="list-style-type: none"> a) Aceptar. b) Cancelar.
7. El técnico llena los campos que desea modificar y selecciona la opción Aceptar.	8. El sistema verifica que los datos entrados sean correctos.
	9. El sistema modifica los datos del estudiante y actualiza la base de datos.
Flujos Alternos 4a” El sistema no encontró los estudiantes”	
Acción del Actor	Respuesta del Sistema
	4a1. El sistema muestra que no se encontraron estudiantes según los criterios de búsqueda. Retorna a 3.
Flujos Alternos 7a” El técnico provincial selecciona la opción Cancelar”	
Acción del Actor	Respuesta del Sistema
	7a 1. El sistema cierra el formulario y retorna a 3.
Flujos Alternos 9a” El sistema no pudo modificar los datos ”	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

9a1. El sistema muestra los errores existentes y retorna a 3.

Prototipo de Interfaz de Usuario

Gestionar Matrícula en FEA

Nombre: Provincia: --Seleccione-- 

Apellidos: Curso FEA: --Seleccione-- 

Nombre	Curso Fea	Provincia
Julio Cesar Sanz Medrano	2007-2008	Las Tunas
Pedro Perez Gil	2007-2008	Guantánamo
Juan Marichal Pupo	2006-2007	S: Spiritus
Margarita Garcia Gonzalez	2006-2007	Granma
Martolome Ibarra Castillo	2006-2007	Holguín
Miguel Barrientos González	2005-2006	S. Cuba
Laura Balsameda Puig	2007-2008	V. Clara
Maria Caridad Pi	2006-2007	Matanzas
Luis Angel Santos	2007-2008	I. juventud

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Modificar datos del estudiante

Nombre <input style="width: 90%;" type="text"/>	Sexo <input style="width: 90%;" type="text" value="--Seleccione--"/>
Primer Apellido <input style="width: 90%;" type="text"/>	Nivel Escolar <input style="width: 90%;" type="text" value="--Seleccione--"/>
Segundo Apellido <input style="width: 90%;" type="text"/>	Quiere SUM <input style="width: 90%;" type="text" value="--Seleccione--"/>
C. Identidad <input style="width: 90%;" type="text"/>	Categoría
Dirección <input style="width: 90%;" type="text"/>	<input checked="" type="radio"/> Estudiante <input checked="" type="radio"/> Trabajador
Curso FEA <input style="width: 90%;" type="text" value="--Seleccione--"/>	Politécnico <input style="width: 90%;" type="text"/>
Provincia <input style="width: 90%;" type="text" value="--Seleccione--"/>	Ocupación <input style="width: 90%;" type="text"/>
Municipio <input style="width: 90%;" type="text" value="--Seleccione--"/>	<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>

Sección "Detalles del estudiante"

Acción de autor	Respuesta del sistema
3. El técnico provincial escoge al estudiante al cual le desea ver los datos y selecciona en el menú la opción "Detalles".	4. El sistema muestra un formulario con los datos del estudiante seleccionado. Con la opción de a) Aceptar.
5. El técnico selecciona la opción Aceptar	6. El sistema cierra el formulario y retorna a 3.

Flujos Alternos 4a" El sistema no encontró los estudiantes"

Acción del Actor	Respuesta del Sistema
	4a1. El sistema muestra que no se encontraron estudiantes según los criterios de búsqueda.

Prototipo de Interfaz de Usuario

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Datos del estudiante

Nombre:	Adriana
Primer Apellido:	Mendez
Segundo Apellido:	García
Sexo:	Femenino
Carnet de Identidad:	85101324550
Dirección:	Calle 1ra e/ 12 y 25 No42
Curso FEA:	2005-2006
Provincia:	La Habana
Municipio:	San Jose de las lajas
Nivel Escolar:	Nivel técnico
Categoría:	Estudiante
Politécnico:	Frank Pais
Fecha de Matricula:	12 de septiembre del 2005

Aceptar

CU. Pasar estudiantes de año en SUM

Caso de Uso:	Pasar estudiantes de año en SUM
Actores:	Técnico de la delegación provincial
Resumen:	Este caso de uso se inicia cuando el técnico provincial solicita pasar los estudiantes de año.
Precondiciones:	Deben de existir estudiantes matriculados en SUM

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Referencias	RF14
Prioridad	Crítico
Flujo Básico	
Acción del Actor	Respuesta del Sistema
1. El técnico provincial selecciona la opción “Pasar estudiante de año”	2. El sistema muestra un formulario con criterios de búsquedas: - Año. Con la opción Buscar. Cancelar.
3.El técnico provincial selecciona el criterio de búsqueda o combinación de los mismos y selecciona la opción “Buscar”	4. El sistema muestra los estudiantes ya seleccionados permitiendo desmarcar aquellos que no pasarán de año. Con la opción a) Pasar de año. b) Cancelar.
5-El técnico provincial selecciona la opción “Pasar de año”	6- El sistema realiza la acción y muestra un mensaje “Estudiantes pasados de año con éxito”. Retorna a 1.
Flujos Alternos 4a”No existen estudiantes en ese año”	
Acción del Actor	Respuesta del sistema
	4a1. El sistema muestra una tabla vacía.
Flujos Alternos 5a”El técnico selecciona la opción Cancelar”	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Acción del Actor	Respuesta del sistema																																																																						
	5a1. Retorna a 1.																																																																						
Prototipo de Interfaz de Usuario																																																																							
<div style="border: 1px solid #ccc; padding: 10px;"> <p>Buscar año</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> Año <div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">1er Año</div> ▼ </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> Buscar Cancelar </div> </div> <div style="border: 1px solid #ccc; padding: 10px;"> <p>Pasar estudiantes de año</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Nombre</th> <th>Apellidos</th> <th>C.Identidad</th> <th>Sexo</th> <th>Provincia</th> <th>Municipio</th> <th></th> </tr> </thead> <tbody> <tr><td>Julio</td><td>Sanz Medrano</td><td>88040320423</td><td>M</td><td>Las Tunas</td><td>Amancio</td><td>✓</td></tr> <tr><td>Cesar</td><td>González</td><td>82040320423</td><td>M</td><td>Holguin</td><td>Mayarí</td><td>✓</td></tr> <tr><td>Rosa</td><td>Blanco</td><td>89120620423</td><td>F</td><td>Matanzas</td><td>Cárdenas</td><td>✓</td></tr> <tr><td>Margarita</td><td>Fernández</td><td>85120620423</td><td>F</td><td>Villa Clara</td><td>Remedios</td><td>✓</td></tr> <tr><td>Maura</td><td>Pérez</td><td>86120340423</td><td>F</td><td>Santiago</td><td>Santiago</td><td>✓</td></tr> <tr><td>Bertha</td><td>Hernández</td><td>87040354553</td><td>F</td><td>Guantánamo</td><td>Guantánamo</td><td>✓</td></tr> <tr><td>Gerardo</td><td>Castillo</td><td>87050324536</td><td>M</td><td>Granma</td><td>Manzanillo</td><td>✓</td></tr> <tr><td>Gonzalo</td><td>Batista</td><td>88020356656</td><td>M</td><td>C. Avila</td><td>Moron</td><td>✓</td></tr> <tr><td>Erick</td><td>Rivero</td><td>88050323563</td><td>M</td><td>Matanzas</td><td>Varadero</td><td>✓</td></tr> </tbody> </table> <div style="display: flex; justify-content: space-between; margin-top: 10px; align-items: center;"> Anterior << 1,2,3,4,5,6,7,8,9,10 >> Siguiente Pasar de Año Cancelar </div> </div>		Nombre	Apellidos	C.Identidad	Sexo	Provincia	Municipio		Julio	Sanz Medrano	88040320423	M	Las Tunas	Amancio	✓	Cesar	González	82040320423	M	Holguin	Mayarí	✓	Rosa	Blanco	89120620423	F	Matanzas	Cárdenas	✓	Margarita	Fernández	85120620423	F	Villa Clara	Remedios	✓	Maura	Pérez	86120340423	F	Santiago	Santiago	✓	Bertha	Hernández	87040354553	F	Guantánamo	Guantánamo	✓	Gerardo	Castillo	87050324536	M	Granma	Manzanillo	✓	Gonzalo	Batista	88020356656	M	C. Avila	Moron	✓	Erick	Rivero	88050323563	M	Matanzas	Varadero	✓
Nombre	Apellidos	C.Identidad	Sexo	Provincia	Municipio																																																																		
Julio	Sanz Medrano	88040320423	M	Las Tunas	Amancio	✓																																																																	
Cesar	González	82040320423	M	Holguin	Mayarí	✓																																																																	
Rosa	Blanco	89120620423	F	Matanzas	Cárdenas	✓																																																																	
Margarita	Fernández	85120620423	F	Villa Clara	Remedios	✓																																																																	
Maura	Pérez	86120340423	F	Santiago	Santiago	✓																																																																	
Bertha	Hernández	87040354553	F	Guantánamo	Guantánamo	✓																																																																	
Gerardo	Castillo	87050324536	M	Granma	Manzanillo	✓																																																																	
Gonzalo	Batista	88020356656	M	C. Avila	Moron	✓																																																																	
Erick	Rivero	88050323563	M	Matanzas	Varadero	✓																																																																	

Gestionar Informe general de matrícula en FEA.

Caso de Uso:	Gestionar Informe general de matrícula en FEA.
Actores:	Técnico DASNA
Resumen:	Este caso de uso se inicia cuando el técnico accede al sistema con la intención de conocer la matrícula general de estudiantes en FEA y

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	para darle al estudiante una pre ubicación laboral, el sistema muestra un informe y el técnico imprime dicho informe.
Precondiciones:	Deben de existir estudiantes matriculados en FEA.
Referencias	RF4, RF 8
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El técnico DASNA escoge la opción “Informe General”	<p>2. El sistema muestra un formulario con los estudiantes matriculados en FEA y tres opciones:</p> <p>2.1. Insertar Pre Ubicación Laboral se va a la sección “Insertar Pre Ubicación Laboral”</p> <p>2.2. Modificar Pre Ubicación Laboral se va a la sección de “Modificar pre ubicación Laboral”</p> <p>2.3. Ver cantidad por Provincia va a la sección “Cantidad por provincia”</p> <p>Muestra un formulario con criterios de búsquedas y la opción</p> <p style="padding-left: 40px;">- Buscar</p>
Sección “Insertar Pre Ubicación Laboral”	
Acción del Actor	Respuesta del Sistema
3. El técnico DASNA selecciona los criterios de o búsqueda o combinación de los mismo y escoge	4. El sistema muestra una lista con todos los matriculados en FEA según los criterios de

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

la opción Buscar	búsqueda seleccionado
5. El técnico DASNA escoge el estudiante al cual le va a insertar pre –ubicación laboral y selecciona la opción Insertar pre-ubicación laboral.	6. El sistema un formulario para insertar la pre ubicación laboral. 6.1 Lugar de ubicación. 6.1-1. Nombre de la entidad. 6.1-2. Nombre del organismo. 6.1-3. Plaza Con la opción 1. Aceptar 2. Cancelar
7. El técnico DASNA llena el campo pre ubicación laboral y selecciona la opción “Aceptar”	8. El sistema modifica los datos del estudiante y muestra un mensaje “Pre ubicación laboral insertada con éxito.”
Flujos Alternos 7a ”El técnico DASNA selecciona la opción Cancelar ”	
Acción del Actor	Respuesta del Sistema
	7a1. El sistema cierra la página actual y retorna a 3
Prototipo de Interfaz de Usuario	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Informe General

Nombre: Provincia:

Apellidos:

Insertar pre-ubicación	Apellidos	C.Identidad	D.Particular	Provincia	Municipio	Fecha Matrícula	U.Laboral
	Julio Sanz	86060378789	C12 #30	Las Tunas	Colombia	14/05/2007	---
	Cesar Castillo	86042023693	C 15 apto 67	Holguin	Sagua	14/05/2007	MAC
	Rosa Alvarez	84121756636	C 1ra #172	Granma	Rio Cauto	14/05/2007	MAC
	Angela Castro	85030325525	C7ma apto 234	Villa Clara	Remedio	14/05/2007	MAC
	Maria Costa	85050623235	C 2 #2	Las Tunas	Amancio	25/06/2008	MAC
	Caridad Blanco	87111123235	Babiney	Granma	Bayamo	25/06/2008	---
	Erick Soler	88040384523	San Lorenzo	Las Tunas	Amancio	25/06/2008	---
	Rafael Sanchez	84120620423	C 5ta #44	Granma	Manzanillo	25/06/2008	---
	Javier Acuña	88040320423	Lucas Ortiz	Las Tunas	Colombia	25/06/2008	---

Anterior << 1,2,3,4,5,6,7,8,9,10 >> Siguiente

Adicionar Ubicación Laboral

Nombre de la entidad

Nombre del organismo.

Plaza

Sección "Modificar Pre Ubicación Laboral"

Acción del Actor	Respuesta del Sistema
10. El técnico DASNA selecciona los criterios de búsqueda o combinación de los mismo y escoge la opción Buscar	4. El sistema muestra una lista con todos los matriculados en FEA según los criterios de búsqueda seleccionado.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

5. El técnico DASNA escoge el estudiante al cual le va a modificar su pre ubicación laboral y selecciona la opción Modificar pre-ubicación laboral.	6. El sistema muestra un formulario para modificar la pre ubicación laboral. 6.1 Lugar de ubicación. 6.1-1. Nombre de la entidad. 6.1-2. Nombre del organismo. 6.1-3. Plaza Con la opción 3. Aceptar 4. Cancelar
7. El técnico DASNA llena el campo pre ubicación laboral y selecciona la opción "Aceptar"	8. El sistema guarda los cambio en la base de datos y muestra un mensaje "Pre ubicación laboral modificada con éxito."
Flujos Alternos 6a "El técnico DASNA selecciona la opción Cancelar "	
Acción del Actor	Respuesta del sistema
	6a1. El sistema cierra la página actual y retorna a 3.
Prototipo de Interfaz de Usuario	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Informe General

Nombre: Provincia:

Apellidos:

No.	Modificar pre-ubicación	C.Identidad	D.Particular	Provincia	Municipio	Fecha Matrícula	U.Laboral
Julio	Sanz	86060378789	C12 #30	Las Tunas	Colombia	14/05/2007	---
Cesar	Castillo	86042023693	C 15 apto 67	Holguin	Sagua	14/05/2007	MAC
Rosa	Alvarez	84121756636	C 1ra #172	Granma	Rio Cauto	14/05/2007	MAC
Angela	Castro	85030325525	C7ma apto 234	Villa Clara	Remedio	14/05/2007	MAC
Maria	Costa	85050623235	C 2 #2	Las Tunas	Amancio	25/06/2008	MAC
Caridad	Blanco	87111123235	Babiney	Granma	Bayamo	25/06/2008	---
Erick	Soler	88040384523	San Lorenzo	Las Tunas	Amancio	25/06/2008	---
Rafael	Sanchez	84120620423	C 5ta #44	Granma	Manzanillo	25/06/2008	---
Javier	Acuña	88040320423	Lucas Ortiz	Las Tunas	Colombia	25/06/2008	---

Anterior << 1,2,3,4,5,6,7,8,9,10 >> Siguiete

Modificar Ubicación Laboral

Nombre de la entidad

Nombre del organismo.

Plaza

Sección "Cantidad por provincia"

Acción del Actor

Respuesta del sistema

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

<p>3. El técnico DASNA selecciona la opción Ver Cantidad por provincia.</p>	<p>3. El sistema muestra una lista con todos los matriculados en FEA y un criterio de búsqueda donde selecciona la provincia específica. Con la opción</p> <ol style="list-style-type: none"> 1. Buscar 2. Cancelar.
<p>4. El técnico DASNA selecciona la provincia deseada y selecciona la opción Buscar.</p>	<p>5. El sistema muestra los estudiantes en esa provincia y muestra la cantidad en cada municipio.</p>
<p>Flujos Alternos 4a "El técnico DASNA selecciona la opción Cancelar "</p>	
<p>Acción del Actor</p>	<p>Respuesta del sistema</p>
	<p>4a1. El sistema cierra la página actual y retorna a 3.</p>
<p>Prototipo de Interfaz de Usuario</p>	

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Seleccione la provincia

Provincia

Resultado de la búsqueda

Cantidad de estudiantes

Nombre	Apellidos	C.Identidad	D.Particular	Municipio
Julio	Sanz	86060378789	C12 #30	San Jose de las Lajas
Cesar	Castillo	86042023693	C 15 apto 67	Santa Cruz del norte
Rosa	Alvarez	84121756636	C 1ra #172	Caimito
Angela	Castro	85030325525	C7ma apto 234	Artemisa
Maria	Costa	85050623235	C 2 #2	San Jose de las Lajas
Caridad	Blanco	87111123235	Babiney	Guines
Erick	Soler	88040384523	San Lorenzo	San Antonio de los Baños
Rafael	Sanchez	84120620423	C 5ta #44	Artemisa
Javier	Acuña	88040320423	Lucas Ortiz	Bauta

<< 1,2,3,4,5,6,7,8,9,10 >>

2.5. Conclusiones

A través del análisis hecho en el presente capítulo de los procesos llevados a cabo en el negocio, se plantearon las ventajas de tener un sistema automatizado. Se demostró que el sistema propuesto plantea una mejora en el trabajo de los técnicos, además de la calidad proporcionada al trabajo de los mismos.

Se realizó una descripción detallada de los casos de usos definidos en el sistema y representados en el diagrama de casos de uso del sistema. Se desarrolló un listado de los requisitos funcionales y no funcionales con que debe de contar el sistema.

CAPÍTULO 3: DISEÑO DEL SISTEMA

El diseño se puede definir como el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física y producir un modelo o representación técnica del software que se va a desarrollar.

En este capítulo se abordará la descripción de la arquitectura propuesta así como la fundamentación del uso del framework symfony y los patrones de diseño que utiliza.

3.1. Descripción de la Arquitectura de Software Propuesta

Un sistema grande y complejo requiere de una arquitectura para que los desarrolladores puedan progresar hasta tener una visión común. La descripción de la arquitectura permite al arquitecto controlar el desarrollo del sistema desde la perspectiva técnica. Se centra tanto en los elementos estructurales significativos del sistema, como subsistemas, clases, componentes y nodos, como en las colaboraciones que tienen lugar entre estos elementos a través de las interfaces

La aplicación se fundamentará en un diseño modular gestionado a través de capas, que permitan un desarrollo incremental de la misma. Para esto la arquitectura del sistema estará basada en capas o layers, donde estará integrado por 4 capas:

- **Presentación:** Se encargará de interactuar con los usuarios que utilizarán la aplicación. Se implementará utilizando formularios, controles y componentes que permitan procesar y dar formato a los datos de los usuarios, así como obtener y validar los datos dados por estos.
- **Negocio:** Se implementan las reglas del negocio, es donde se gestionan los pedidos de la capa de presentación y respuestas hacia la capa de presentación.
- **Acceso a Datos:** Es la encargada de acceder a los datos que se encuentran en la base de datos y de convertir los datos en objetos persistentes.
- **SGBD:** Representa la base de datos como tal. Es donde se almacenan todos los datos de la aplicación.

En toda arquitectura de capa los elementos agrupados en una misma capa pueden comunicarse entre si; pero existen variantes en cuanto a las comunicaciones permitidas entre elementos de capas diferentes.

Una capa es un conjunto de subsistemas que comparten el mismo grado de generalidad y de volatilidad en las interfaces: las capas inferiores son de aplicación general a varias aplicaciones y deben poseer interfaces más estables, mientras que las capas altas son más dependientes de la aplicación y pueden tener interfaces menos estables.

3.2. Descripción del Framework.

Un framework es una estructura de software combinada de componentes personalizables e intercambiables para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Proporcionan una estructura definida la cual beneficia a crear aplicaciones con mayor rapidez. Ayudan en el desarrollo de software y a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación. Son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos y de acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Es por ello que se escogió el framework symfony para el desarrollo de este trabajo ya que es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web.

Symfony es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Está diseñado con el objetivo de optimizar la creación de las aplicaciones web, con el uso de sus características ya que automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Posee una librería de clases que permiten reducir el tiempo de desarrollo. Está desarrollado en PHP5, por lo que utiliza los mecanismos orientados a objetos disponibles en este.

Para acceder de forma efectiva a la base de datos desde un contexto orientado a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se nombre ORM (object- relational mapping) o “- mapeo de objetos a bases de datos”, y está formada por objetos que permiten acceder a los datos y que contienen en sí mismos el código necesario para hacerlo.

Propel es el ORM que utiliza Symfony por defecto, además de ser un proyecto de software libre, es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5.

Symfony presenta varias características que posibilita un mejor trabajo con este, es fácil de instalar y configurar en la mayoría de plataformas, es independiente del sistema gestor de bases de datos y a la misma vez es sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos. Sigue la mayoría de mejores prácticas y patrones de diseño para la web, esta preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Presenta un código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo, es fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Internamente presenta un conjunto de características web que automatiza la mayoría de elementos comunes de los proyectos web, como la capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos. La capa de presentación utiliza plantillas y layouts que pueden ser creados por diseñadores HTML sin ningún tipo de conocimiento del framework. Los helpers incluidos permiten minimizar el código utilizado en la presentación, ya que encapsulan grandes bloques de código en llamadas simples a funciones. Los datos incluyen mecanismos de escape que permiten una mejor protección contra los ataques producidos por datos corruptos.

Symfony incorpora por defecto varios entornos de desarrollo diferentes e incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del software. Las herramientas que generan automáticamente código han sido diseñadas para hacer prototipos de aplicaciones y para crear fácilmente la parte de gestión de las aplicaciones.

YAML es “un formato para serializar datos que es fácil de procesar por las máquinas, fácil de leer para las personas y fácil de interactuar con los lenguajes de script”. Dicho de otra forma, YAML es un lenguaje muy sencillo que permite describir los datos como en XML, pero con una sintaxis mucho más sencilla. Es un formato especialmente útil para describir datos que pueden ser transformados en arrays simples y asociativos.

YAML es mucho más rápido de escribir que XML (ya que no hacen falta las etiquetas de cierre y el uso continuo de las comillas) y es mucho más poderoso que los tradicionales archivos .ini (ya que estos últimos no soportan la herencia y las estructuras complejas). Por este motivo, Symfony utiliza el formato YAML como el lenguaje preferido para almacenar su configuración.

Patrón arquitectónico Modelo-Vista-Controlador (MVC).

Symfony está basado en el patrón de arquitectura MVC.

Model-View-Controller (MVC) es un patrón de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralelamente e independiente. MVC sugiere la separación del software en 3 estratos:

- *El modelo* representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Aquí se trata la comunicación a todos los niveles con la base de datos.
- *La vista* transforma el modelo en una página web que permite al usuario interactuar con ella.
- *El controlador* se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Aquí se produce la comunicación del usuario con la aplicación. Cada acción requerida por el usuario tiene su reflejo en un flujo de llamadas al modelo de datos desde la capa de control.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos,

haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

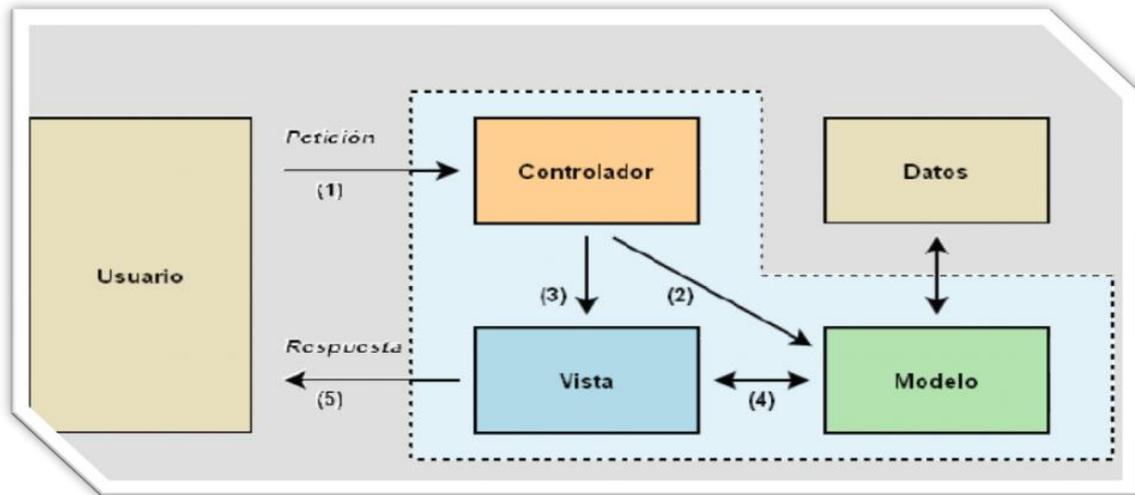


Fig. 3.1. Funcionamiento del patrón MVC.

El principio más importante de la arquitectura MVC es la separación del código del programa en tres capas, dependiendo de su naturaleza. La lógica relacionada con los datos se incluye en el modelo, el código de la presentación en la vista y la lógica de la aplicación en el controlador. La programación se puede simplificar si se utilizan otros patrones de diseño. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas.

La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos. La vista se separa en un layout y en una plantilla y el controlador normalmente se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página.

Las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas, para la realización de estas consultas a la base de datos Symfony emplea los patrones Active Record y Active Table:

- **Active Table:** Es un objeto que puede persistir otros objetos en medios no volátiles, como bases de datos. La función de una clase que implementa este patrón es de comunicarse con la base de datos y regresarnos objetos como los tenemos definidos para nuestra aplicación.

- **Active Record** : Una de las virtudes del patrón es representar de forma Orientada a Objetos los datos de una Base de Datos Relacional – modelo conocido también como ORM o “Object-Relational Mapping” - , definiendo interfaces sencillas para acceder y manipular esos datos.

El patrón Front Controller

El patrón Front Controller está pensado para centralizar las invocaciones de lógica de la aplicación en un único lugar. Para aplicaciones Web se utiliza este patrón ya que obliga a que todas las peticiones hechas a nuestra aplicación pasen por un servlet Controlador.

Este controlador proporciona un punto de entrada único que controla y gestiona las peticiones Web realizadas por los clientes. Teniendo este único punto de entrada se evita tener que repetir la misma lógica de control en todos los .jsp. Normalmente se utiliza junto con un Dispatcher que es el responsable de redirigir el flujo de ejecución hacia el jsp adecuado. Este Dispatcher puede ser realizado por el propio controlador o estar en una clase a parte.

Si nos situamos dentro del patrón MVC, el ‘Front Controller’ es el componente que recibe los requerimientos, los envía a los elementos encargados de procesar la lógica y luego lo envía nuevamente a la vista (esta vez incluyendo los datos obtenidos). Si no tuviéramos un punto de acceso único, tendríamos los siguientes problemas:

- a) Código duplicado (nuevamente).
- b) Manejo no centralizado de vistas.

En el caso de Symfony, el controlador se divide en varias capas entre ellas la capa del controlador frontal, que es el punto de entrada único de toda la aplicación en un entorno determinado. Carga la configuración y determina la acción a ejecutarse.

Symfony toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. Por esto la arquitectura MVC proporciona grandes ventajas, como la organización del código, la reutilización, la flexibilidad y una programación mucho más entretenida. Por si fuera poco, crear la aplicación con Symfony permite crear páginas XHTML válidas, depurar fácilmente las aplicaciones, crear una configuración sencilla, abstracción de la base de datos utilizada,

enrutamiento con URL limpias, varios entornos de desarrollo y muchas otras utilidades para el desarrollo de aplicaciones.

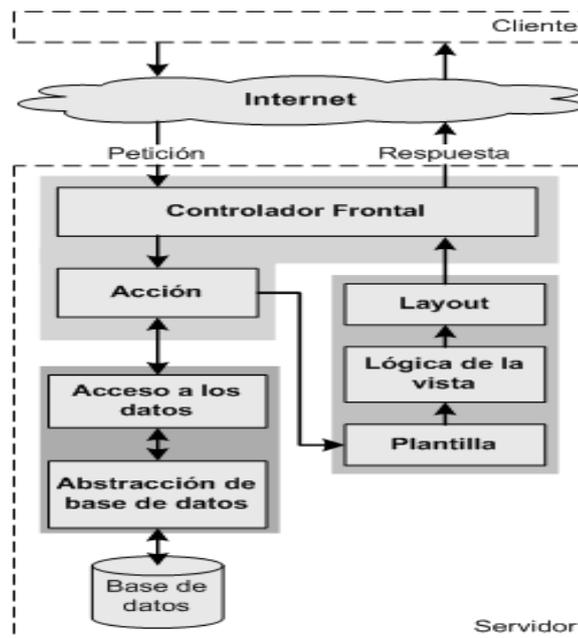


Fig. 3.2. Flujo de Trabajo del Symfony

Symfony implementa los patrones de diseños GOF. Estos son una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular.

Los patrones de diseño se clasifican según

- ✓ Su propósito.
 - Creacionales: Tratan la creación de instancias.
 - Estructurales: Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad.
 - De comportamiento: Tratan la interacción y cooperación entre clases.

- ✓ Su ámbito
 - De clase: Basados en la herencia de clases.
 - De objeto: Basados en la utilización dinámica de objetos.

El patrón Decorator

El patrón decorator se usa para adicionar responsabilidades a objetos individuales dinámicamente sin afectar otros objetos. Agrega responsabilidades que pueden ser retiradas. Cuando no es práctico adicionar responsabilidades por medio de la herencia.

En el Symfony la vista se separa en un archivo denominado layout.php que contiene el layout de la página. Este archivo, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este comportamiento es una implementación del patrón de diseño llamado “decorator”.

El patrón Singleton.

El patrón Singleton es uno de los más sencillos patrones de diseño, y es útil para limitar el máximo número de instancias de una clase en exactamente solo una.

Este patrón es efectivo para limitar el número máximo de instancias en exactamente una. En este caso, si más de un objeto necesita utilizar una instancia de una clase Singleton, esos objetos compartirán la misma instancia de la clase Singleton. Una clase que implementa el patrón de Singleton se conoce como una clase Singleton.

Uno de los usos habituales del Patrón de Diseño Singleton es concentrar en un único objeto todas las llamadas a la base de datos y que este se encargue de que el sistema use una única conexión, compartida por todos los que la necesiten.

En el Symfony, una de las clases que forma el núcleo es el singleton de contexto (que se obtiene mediante `sfContext :: getInstance ()`) almacena una referencia a todos los objetos que forman el núcleo de Symfony y puede ser accedido desde cualquier punto de la aplicación. Se trata de un objeto muy útil que guarda una referencia a todos los objetos del núcleo de Symfony relacionados con una petición dada, y ofrece un método accesor para cada uno de ellos.

El patrón Facade

El patrón fachada trata de simplificar la interface entre dos sistemas o componentes de software ocultando un sistema complejo detrás de una clase que hace las veces de pantalla o fachada.

La idea principal de este patrón es ocultar todo lo posible la complejidad de un sistema, el conjunto de clases o componentes que lo forman, de forma que solo se ofrezca un (o unos pocos) punto de entrada al sistema tapado por la fachada. Proporciona al programador una clase sencilla con un grupo de métodos, uno para cada operación permitida y de modo que sean estos métodos los que internamente hagan las operaciones con el fin de llevar a cabo la correcta lógica de la aplicación. A esta clase se le conoce como fachada (facade).

Una ventaja más de usar una clase fachada para comunicar las dos partes o componentes, es la de aislar los posibles cambios que se puedan producir en alguna de las partes.

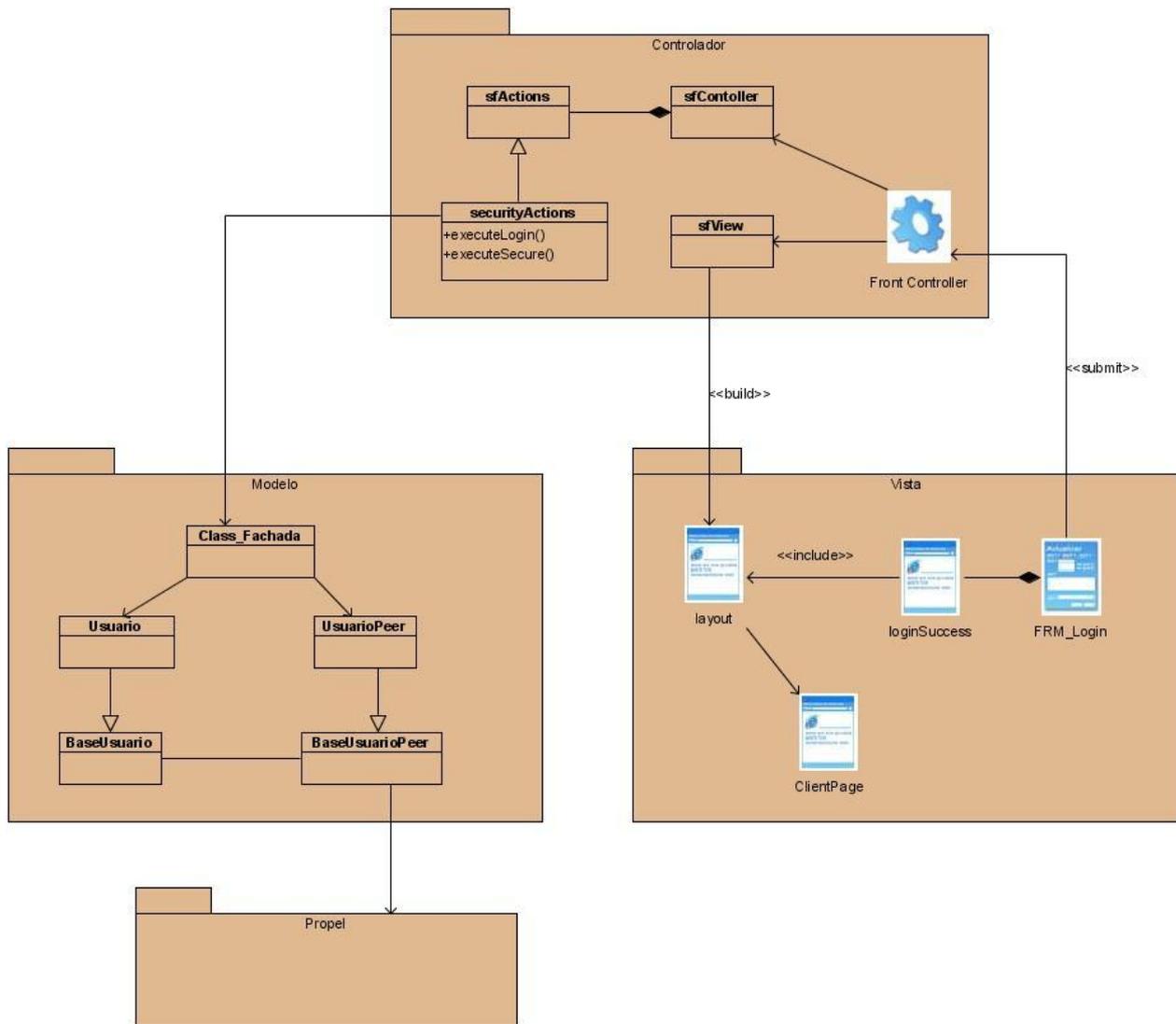


Fig. 3.3. Diagrama de Clases del Diseño con su integración con Symfony

3.3. Modelo de diseño

El diseño es un flujo de trabajo que se lleva a cabo para poder realizar una implementación segura y precisa, a través de este flujo se ajusta el sistema de tal forma que se incluyan los requisitos funcionales y no funcionales.

Durante el ciclo de desarrollo iterativo es posible pasar a la fase de diseño, una vez terminados los documentos del análisis. Durante este paso se logra una solución lógica que se funda en el paradigma orientado a objetos. Su esencia es la elaboración de diagramas de interacción, que muestran gráficamente como los objetos se comunican entre ellos a fin de cumplir con los requerimientos. Mediante estos diagramas se lleva a una buena preparación para el diseño. Con la creación de los diagramas de interacción se llega a la realización de los diagramas de diseño de clases que resumen la definición de las clases (e interfaces) implementables en el software.

3.3.1. Patrones de diseño.

Los patrones de diseño son herramientas que proveen facilidades para hacer software reutilizable y de buena calidad. Cada patrón describe un problema que ocurre repetidamente en nuestro entorno, y describe el núcleo de la solución a ese problema, de tal forma que ésta pueda ser usada un millón de veces, sin hacer el mismo trabajo dos veces.

Los patrones de diseño son soluciones concretas, proponen soluciones a problemas concretos, no son teorías genéricas. Se utilizan en situaciones frecuentes, ya que se basan en la experiencia acumulada para resolver problemas reiterativos. Favorecen la reutilización de código, ayudan a construir software basado en la reutilización, a construir clases reutilizables. Los propios patrones se reutilizan cada vez que se vuelven a aplicar y son soluciones técnicas, pues indican resoluciones técnicas basadas en Programación Orientada a Objetos (POO). En ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje.

3.3.1.1. Patrones GRAPS.

Los patrones GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades) que describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Para la realización de este trabajo se utilizaron los siguientes patrones de diseño GRAPS:

- **Experto:** Este patrón asigna una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a

objetos. En el symfony, el Propel es la librería externa que se utiliza para realizar la capa de abstracción en el modelo, encapsula toda la lógica de los datos y son generadas las clases con todas las funcionalidades comunes de las entidades.

- **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento. En las actions del symfony se crean los objetos de las clases que representan las entidades, evidenciando de este modo que la clase Actions es “creador” de dichas entidades.
- **Bajo acoplamiento:** Este patrón se encarga de asignar una responsabilidad para mantener bajo acoplamiento. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. La clase Action hereda solamente de sfActions para lograr un bajo acoplamiento de clases.
- **Alta cohesión:** Este patrón se basa en asignar responsabilidades de manera que la cohesión permanezca alta. Mantener la complejidad manejable. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Symfony permite asignar responsabilidades con una alta cohesión, por ejemplo la clase Actions tiene la responsabilidad de definir las acciones para las plantillas y colabora con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades, es decir, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas proporcionando que el software sea flexible frente a grandes cambios.
- **Controlador:** Su función es asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que representa el sistema global, dispositivo o subsistema (Controlador de fachada) y un escenario de casos de uso en el que tiene lugar un evento de sistema. El controlador es el responsable de gestionar un evento de entrada que delega el trabajo a otros objetos no hacen nada por si mismos. En el symfony todas las peticiones Web son manejadas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema

de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

3.3.1.2. Diagramas de clases del diseño.

Las clases controladoras tienen una relación de composición de uno a uno con la clase fachada.

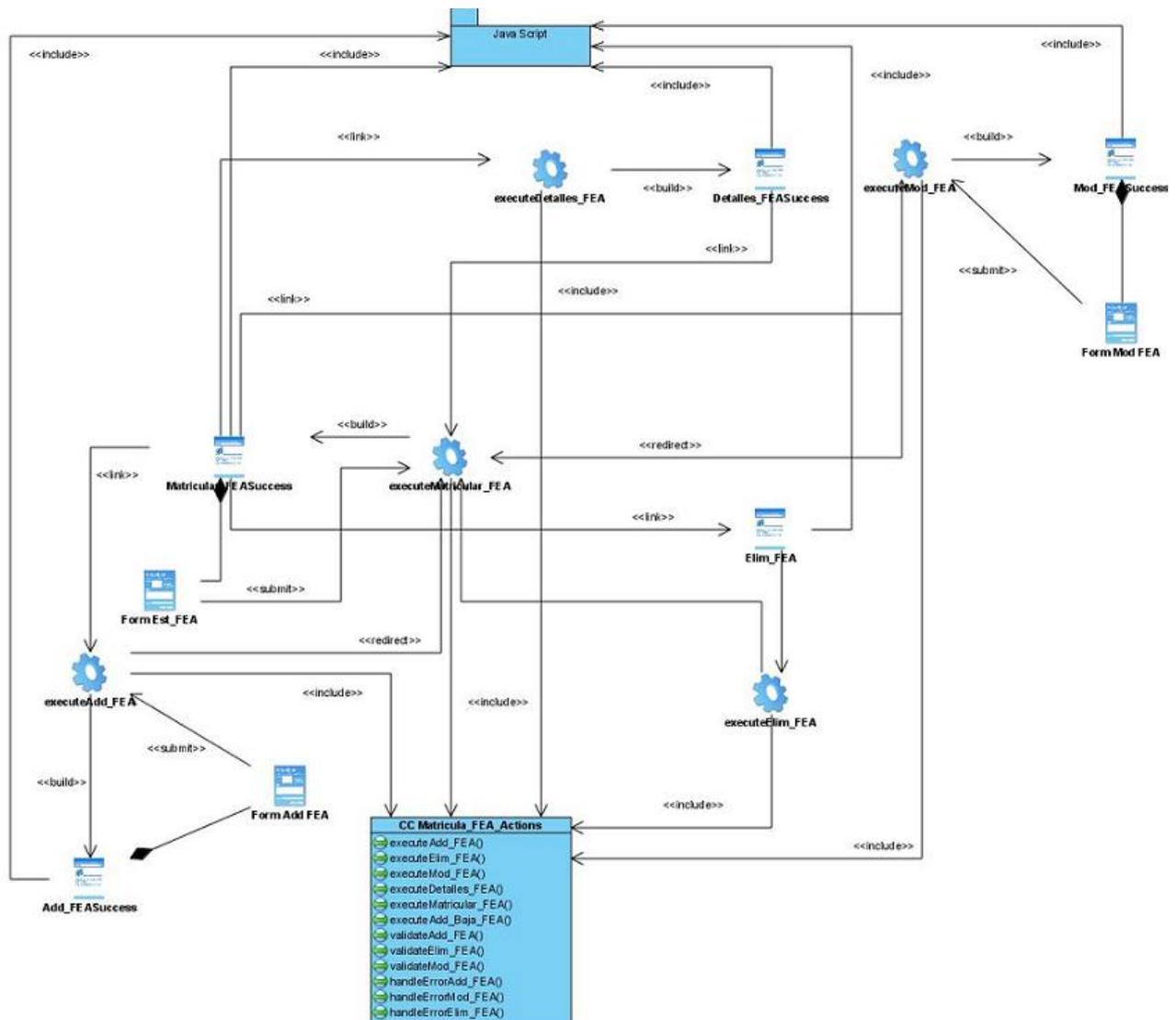


Fig.3.4. Diagrama CD Gestionar Matricula en FEA

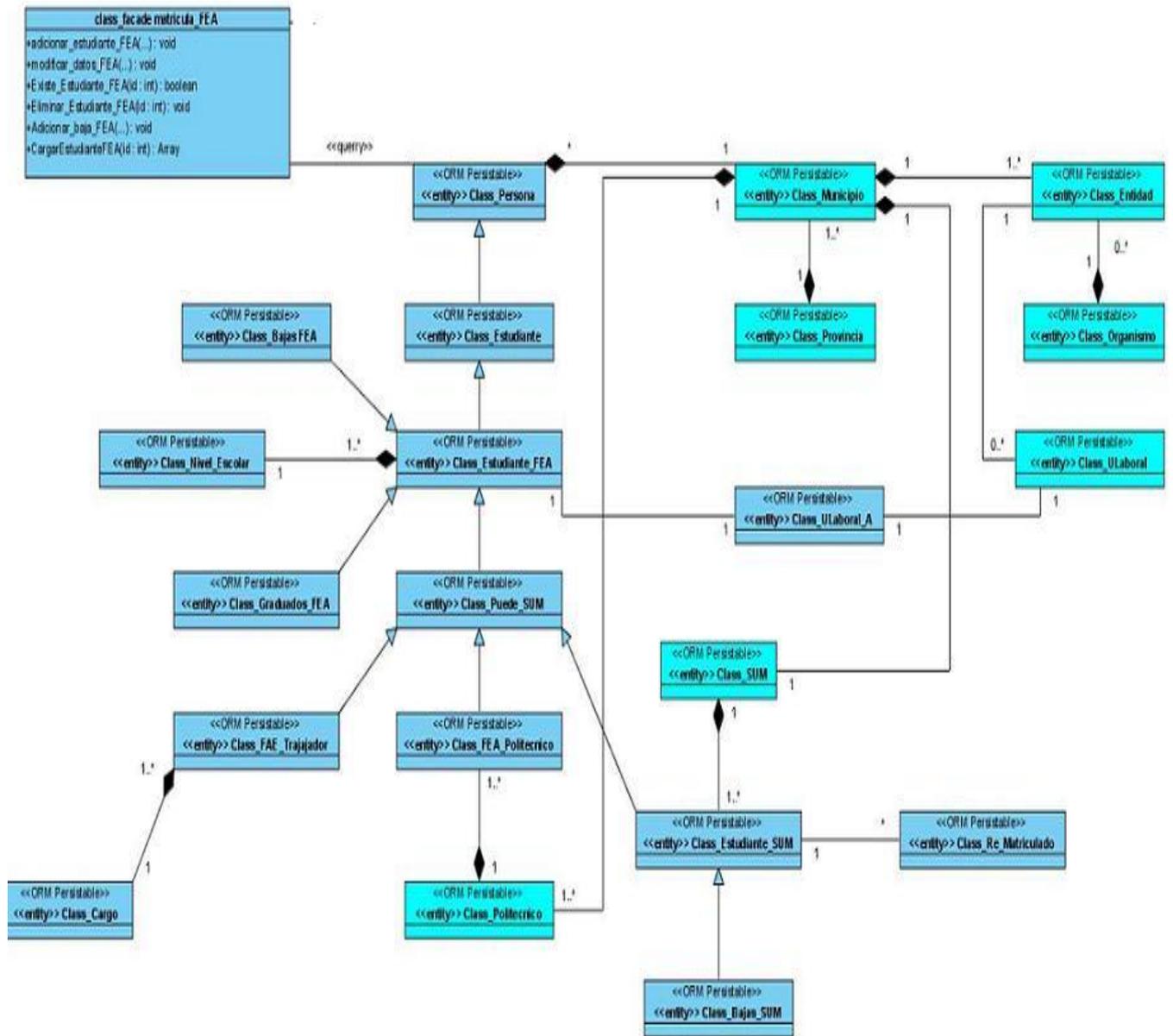


Fig.3.5. Diagrama CD Gestionar Matrícula en FEA (Continuación).

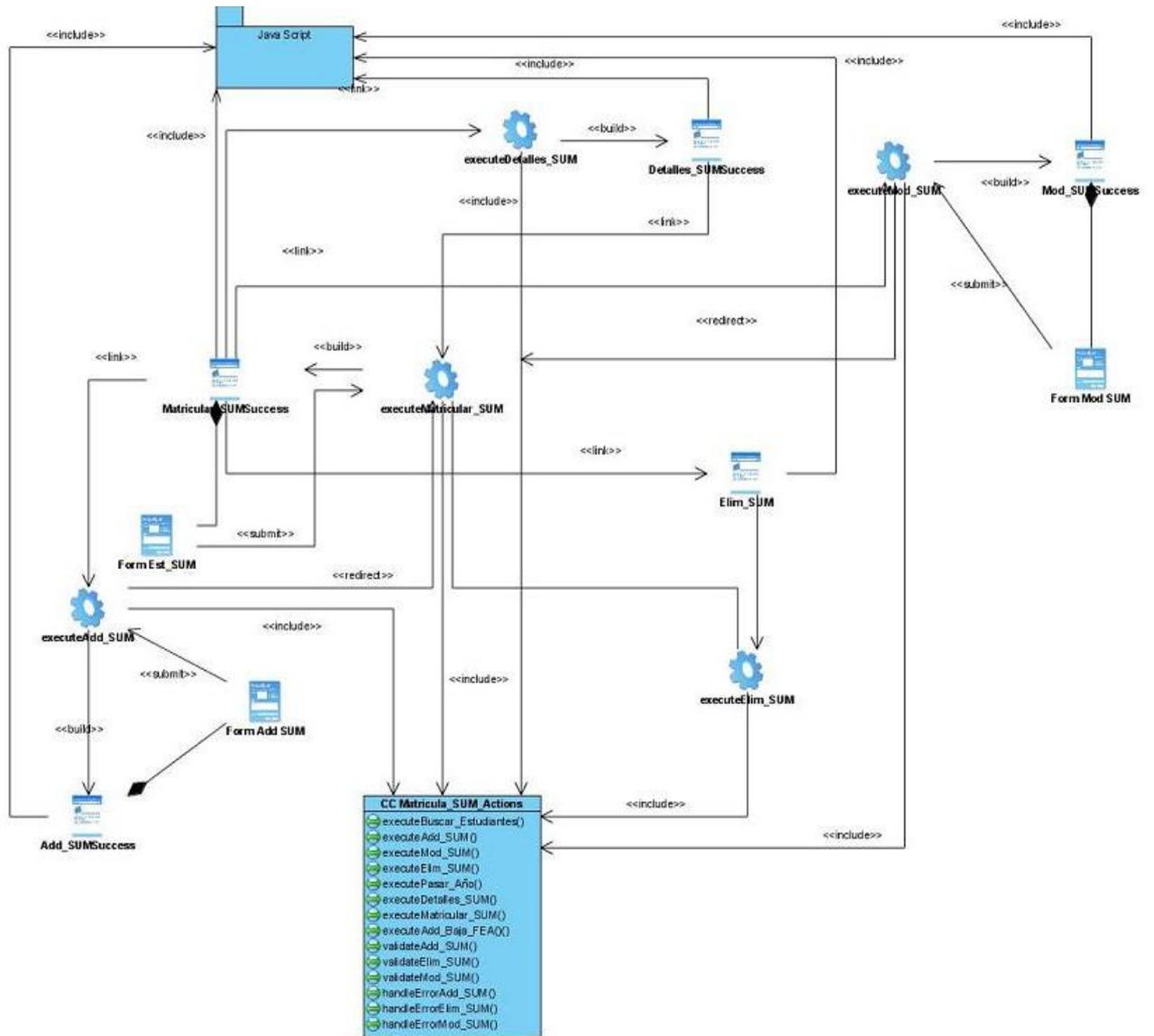


Fig. 3.6. Diagrama CD Gestionar Matrícula en SUM

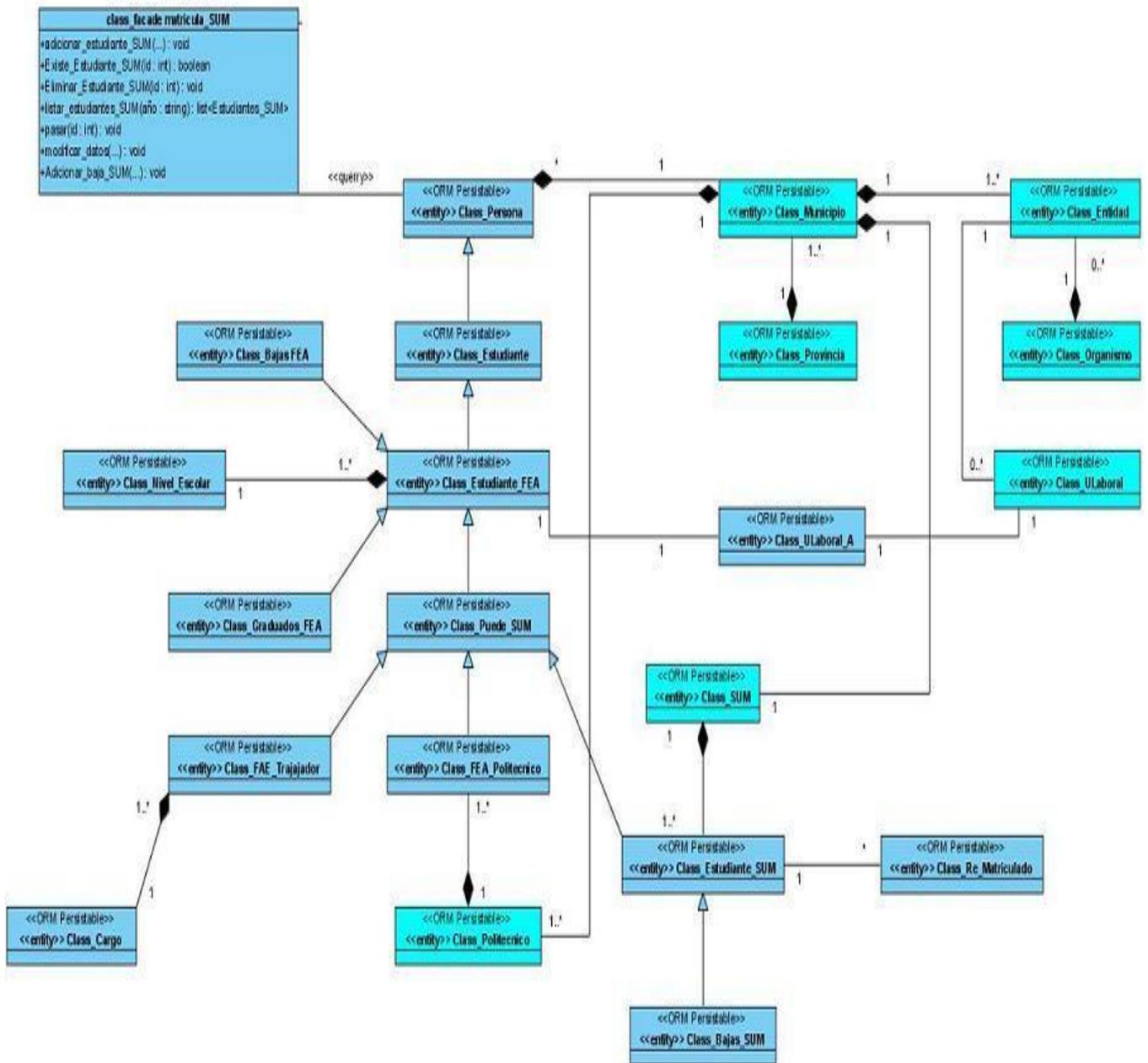


Fig. 3.7. Diagrama CD Gestionar Matrícula en SUM (Continuación).

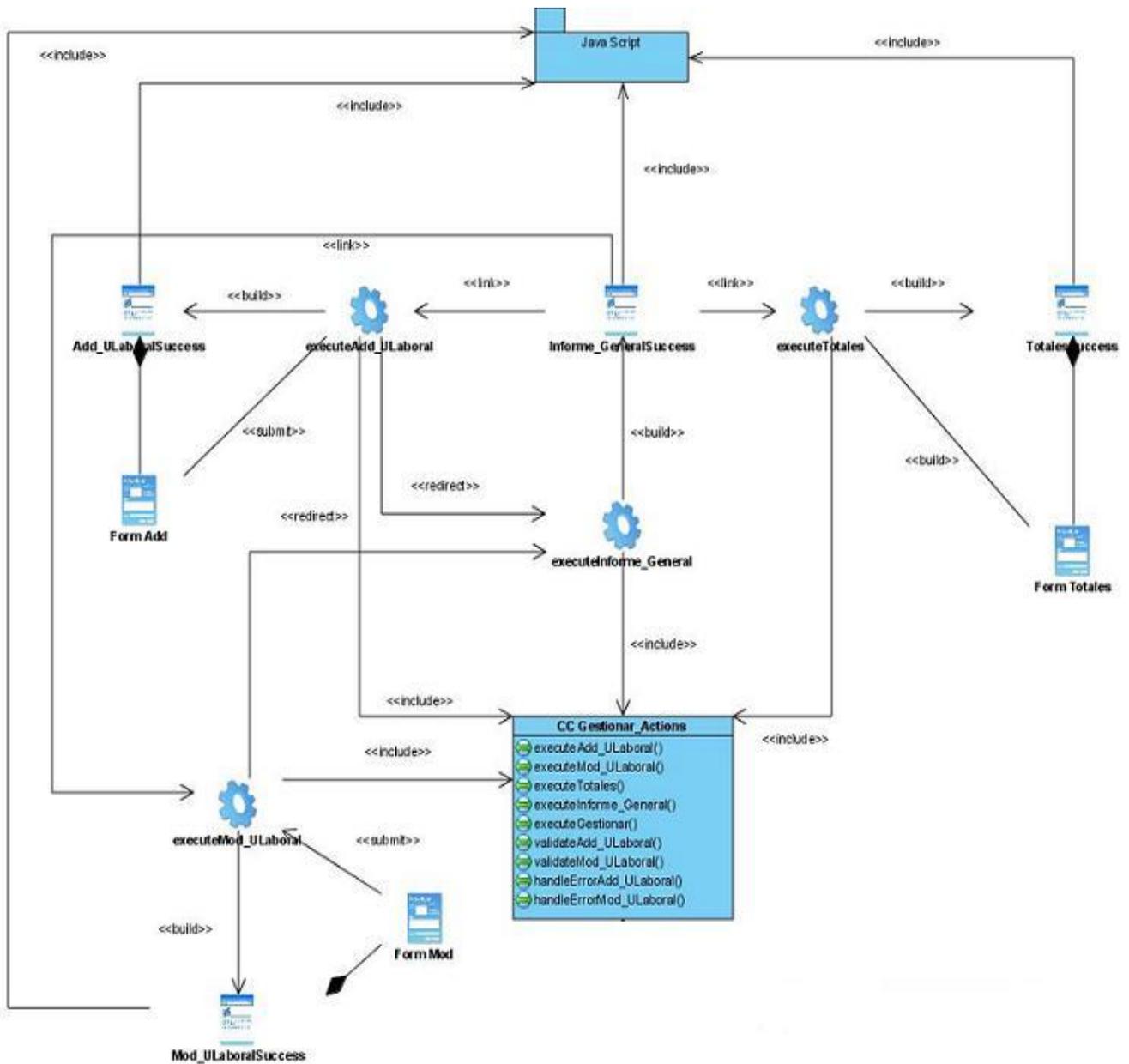


Fig.3.8. Diagrama CD Gestionar Informe general

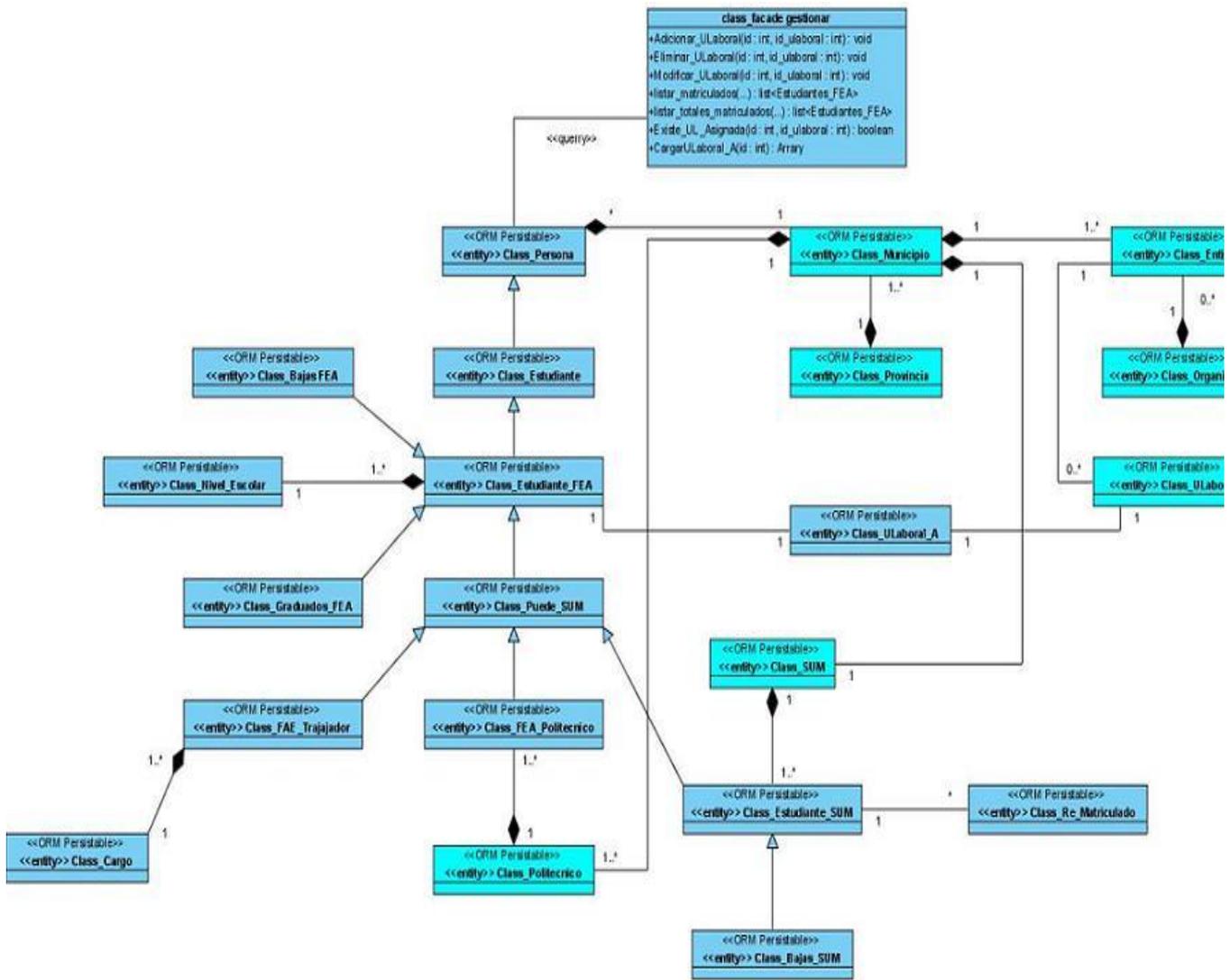


Fig.3.9. Diagrama CD Gestionar Informe general (Continuación).

3.3.1.3. Diagramas de secuencia.

Un diagrama de secuencia muestra las interacciones de objetos organizadas en la secuencia de tiempo. Ellos representan los objetos y clases involucradas en el escenario y la sucesión de mensajes intercambiados entre los objetos necesarios para llevar a cabo la funcionalidad del escenario. Los

diagramas de secuencia, generalmente, se asocian con una realización de casos de uso del sistema bajo desarrollo.

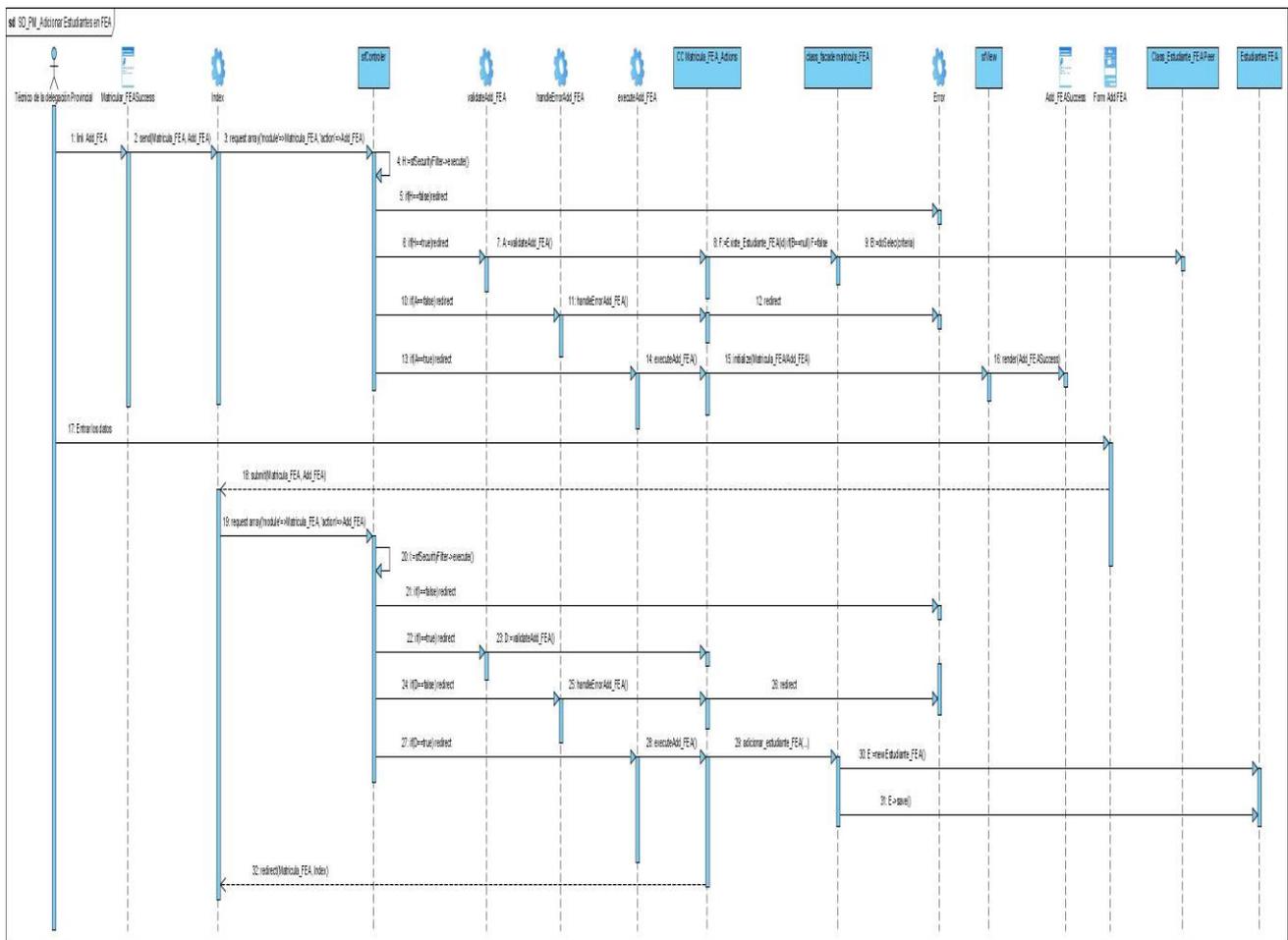


Fig. 3.8. DS Adicionar estudiante en FEA

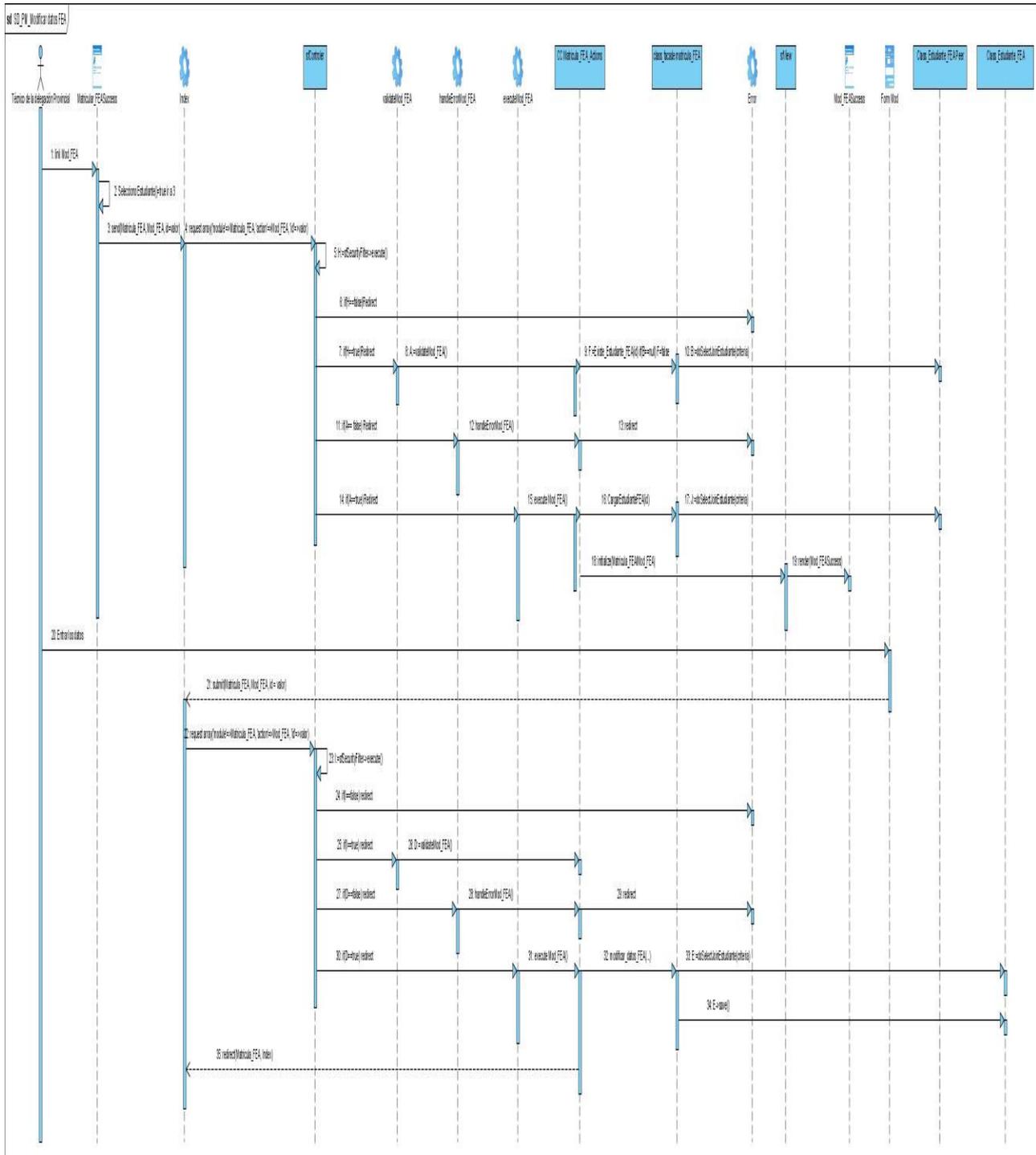


Fig.3.9. DS Modificar datos de estudiante en FEA

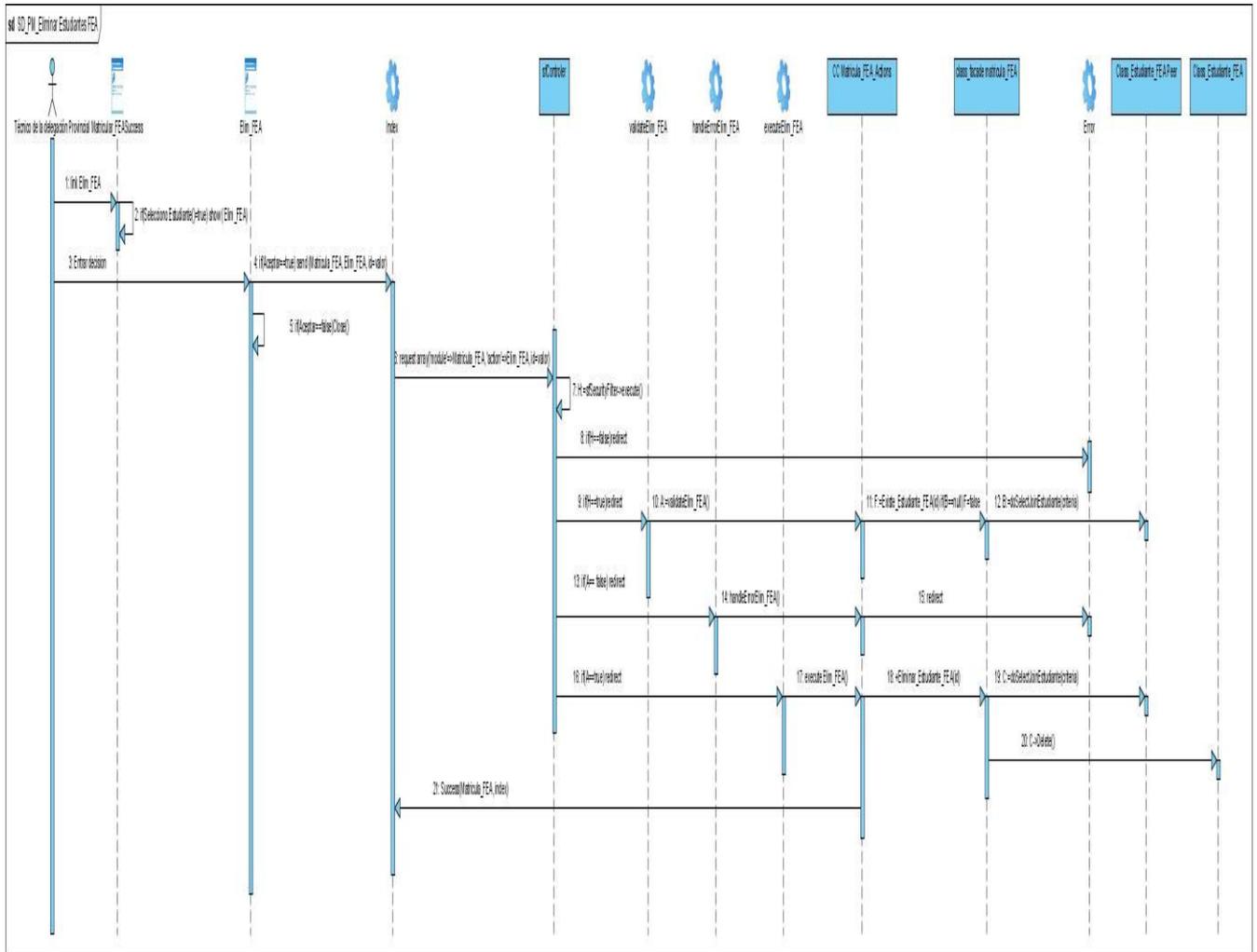
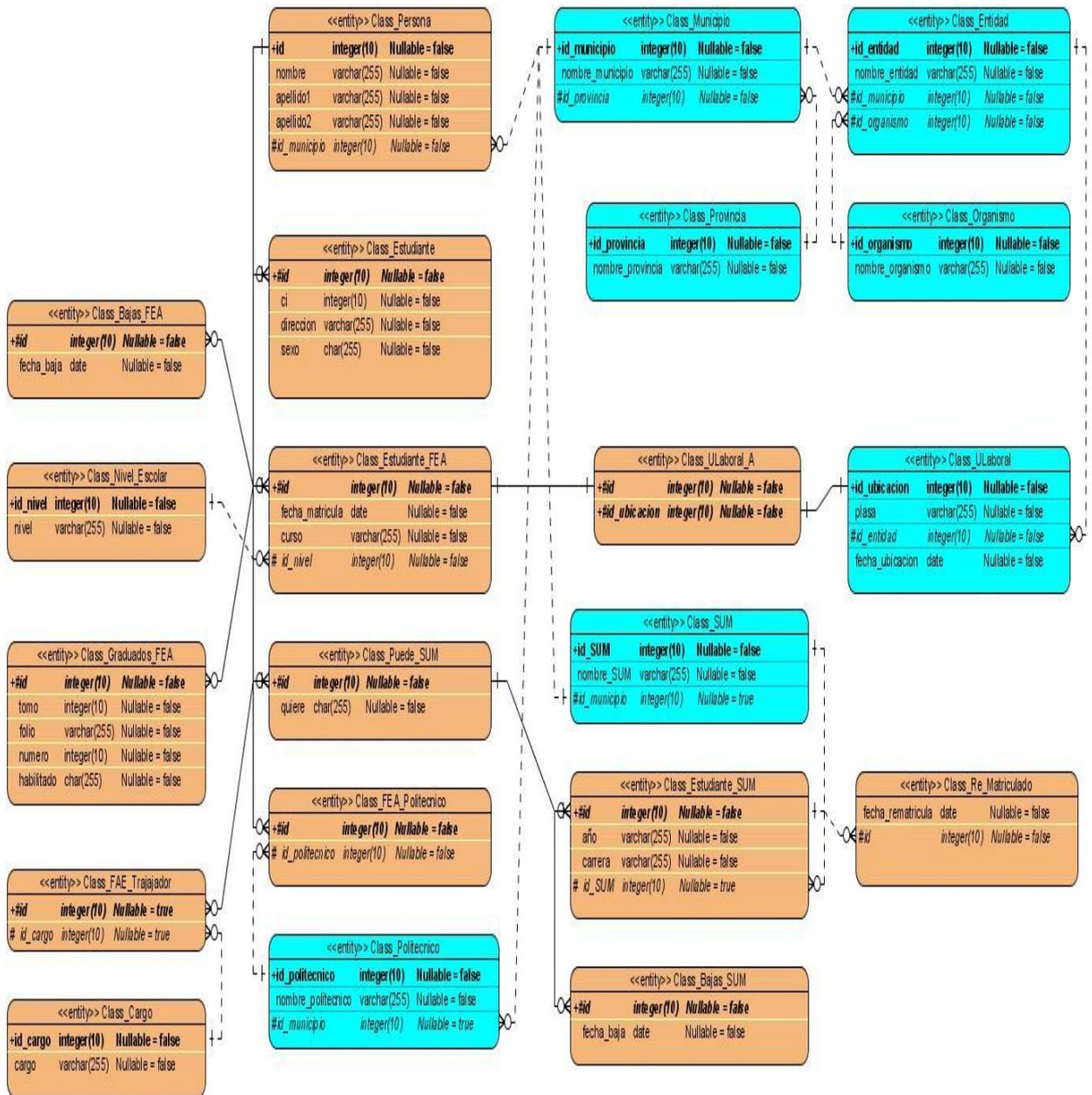


Fig.3.10. DS Eliminar estudiante en FEA

3.3.2. Diseño de la base de datos

Las bases de datos son de suma importancia para guardar datos organizados para un uso determinado que facilite el desarrollo de un sistema informático. Estos datos comparten características comunes con una misma estructura, además de que se pueden almacenar y recuperar en un futuro

3.3.2.1. Modelo Entidad Relación



3.4. Conclusiones

En este capítulo se definió la arquitectura del software. Se describieron las clases que contendrán el diseño de la aplicación web, los diagramas de clases del diseño y de interacción para tener un mejor entendimiento de la aplicación diseñada. Con el diseño de la aplicación propuesta, se obtuvo una traducción correcta de los requisitos funcionales y de esta forma se logró una especificación más detallada de las funcionalidades que debe de tener la aplicación. Esta contará con una interfaz sencilla y de fácil intercambio con el usuario. Se definieron además las clases persistentes.

CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD

Cuando se planifica un proyecto se obtienen las estimaciones del esfuerzo humano requerido, el tiempo de duración y el costo aproximado. En la mayoría de los casos las estimaciones se hacen valiéndose de la experiencia pasada como única guía. Debido a esto, antes de elaborar un proyecto, es importante conocer si es beneficioso seguir adelante con la realización del mismo.

El estudio de la factibilidad tiene como objetivo conocer la viabilidad de implementar un proyecto, ayudándonos a realizar un análisis profundo comparando costos, esfuerzos y beneficios.

En este capítulo se mostrará el estudio realizado para determinar si el software diseñado es factible, utilizando el método COCOMO II con salida de puntos de función.

4.1. Estimación: Cocomo II con salida de puntos de función.

Cuando se habla de estimación de esfuerzo, se refiere a la suma de los tiempos que le dedicarán los diferentes recursos a cierta actividad o al proyecto. Se mide en horas/hombre, días/hombre, semanas/hombre, etc. No importa que el trabajo se haga de forma secuencial por un solo recurso o en paralelo por diferentes personas. Se suman los tiempos de cada uno de ellos para obtener el esfuerzo total.

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema ya que permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso:

- Un Actor típicamente representa a un usuario humano o a otro sistema que interactúa con el sistema bajo análisis.
- Un Caso de Uso representa un gránulo funcional del sistema bajo análisis, relatado como una secuencia de acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo.

Si bien los Casos de Uso permiten especificar la funcionalidad de un sistema bajo análisis, no permiten por sí mismos efectuar una estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo.

Existen algunos métodos para calcular la estimación de proyectos de desarrollo del software. Para este trabajo se decidió utilizar el método llamado COCOMO II directamente sobre los Puntos de Función sin ajustar. Éste método es el preferido en la actualidad para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir.

COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (SLOC, Source Lines Of Code) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (cost drivers) que influyen en el esfuerzo requerido para el desarrollo del software.

Pasos para la realización del proceso

$$PM_{\text{nominal}} = A * (\text{Size})^B$$

Donde:

- **PM_{nominal}** es el esfuerzo necesario requerido en meses-hombre.
- **A** es una constante cuyo valor toma **2.94**.
- **B** es un factor escalar que su valor es **1.07** y está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.
- **Size** es el tamaño estimado del software en Miles de líneas de código (KSLOC) que se calcula como el producto de los puntos de función sin ajustar y el factor de conversión que varía según el lenguaje de programación a utilizar, en este caso utilizaremos PHP cuyo factor de conversión es 30.

$$\text{Size} = \text{factor de conversión} * UFP \text{ (Puntos de función sin ajustar)}$$

$$UFP = \sum A_{\text{aporte}}$$

CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD

	Complejidad			Aporte
	Baja	Media	Alta	
Entradas Externas	$5 * 4 = 20$	$2 * 5 = 10$	$3 * 7 = 21$	51
Salidas Externas				0
Consultas Externas	$4 * 3 = 12$	$2 * 4 = 8$	$9 * 6 = 54$	74
Archivos Lógicos Internos	$7 * 21 = 147$			147
Archivos de Interfaces Externos.				0
			TOTAL:	272

Ahora sustituimos en las ecuaciones y tenemos que:

$$\text{Size} = 30 * 272 = 8160 \text{ SLOC} = \mathbf{8.160 \text{ KLOC}}$$

Después

$$\mathbf{PM_{nominal} = 2.94 * (8.160)^{1.07} = 27.785 \text{ Meses} - \text{ Hombre}}$$

Posteriormente se calcula el ajuste del esfuerzo nominal:

$$\mathbf{PM_{ajustado} = PM_{nominal} * \pi(MEi)}$$

Donde los Mei son los multiplicadores de esfuerzo que varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura). En este caso se va aplicar el modelo de Diseño preliminar. Entonces, se cuantifica los multiplicadores de esfuerzo para éste modelo de la siguiente forma:

CAPÍTULO 4: ESTUDIO DE LA FACTIBILIDAD

<i>Multiplicador</i>	<i>Descripción</i>	<i>Ponderación</i>	<i>Valor</i>
<i>RCPX</i>	<i>Las exigencias de confiabilidad, documentación y volumen de datos son moderadas, y la complejidad del producto es baja</i>	<i>Nominal</i>	<i>1</i>
<i>RUSE</i>	<i>No se pretende reutilizar nada</i>	<i>Bajo</i>	<i>0.95</i>
<i>PDIF</i>	<i>No existen restricciones en cuando al tiempo de CPU o al consumo de memoria, la plataforma es muy estable</i>	<i>Bajo</i>	<i>0.87</i>
<i>PERS</i>	<i>Se tienen analistas y programadores con alta eficiencia y capacidad de trabajo en equipo. Dedicación full-time</i>	<i>Nominal</i>	<i>1</i>
<i>PREX</i>	<i>Tanto los analistas como los programadores tienen aproximadamente 6 meses de experiencia en la aplicación, la plataforma, el lenguaje y las herramientas utilizadas</i>	<i>Muy Bajo</i>	<i>1.33</i>
<i>FCIL</i>	<i>Se tienen herramientas CASE simples e infraestructura de comunicaciones básica</i>	<i>Bajo</i>	<i>1.10</i>
<i>SCED</i>	<i>Se requiere terminar el proyecto en el tiempo estimado</i>	<i>Nominal</i>	<i>1</i>
		Total:	<i>1.20</i>

Con los valores obtenidos de esta tabla se sustituye en la ecuación y se tiene que:

$$PM_{\text{ajustado}} = 27.785 * 1.20 = 33.34 \text{ Meses} - \text{Hombre} \approx \mathbf{33 \text{ Meses} - \text{Hombre}}$$

Redondeamos el número obtenido \approx **33 Meses – Hombre** lo que equivale a **2 años y 9 meses**, que entre 2 personas es **1 año y 4 meses**.

El valor obtenido esta expresado en meses – hombres, por lo que hay que expresarlo en Horas-hombre, y teniendo en cuenta que un mes es aproximadamente 160 horas, el esfuerzo resulta:

$$33 * 160 = 5440 \text{ Horas – Hombre}$$

4.2. Conclusiones.

En este capítulo se realizó el estudio de la factibilidad que propone y justifica el desarrollo del software, así como también el tiempo y costo estimados para su realización. Evidencia los beneficios que reportará la terminación del producto llegando a la conclusión de que el software es totalmente factible.

CONCLUSIONES GENERALES

Después de haber estudiado con profundidad los procesos que se llevan a cabo en el módulo de control de la Formación Emergente de Auditores, se llegó a la realización del diseño del sistema que facilitará la gestión de matrícula de los estudiantes vinculados a la FEA, mediante una propuesta de prototipo de interfaz de usuario no funcional.

En este informe se describieron las herramientas necesarias para llevar a cabo la realización de estos procesos, llegando a cumplir satisfactoriamente el objetivo general propuesto. Se modeló un sistema informático para la automatización de los procesos de control de la Formación Emergente de Auditores en la Dirección de Atención al Sistema Nacional de Auditoría del Ministerio de Auditoría y Control en el país.

El sistema diseñado es de vital importancia para la realización de las matrículas en la FEA debido a que se logrará con esto una mayor organización en el trabajo de los técnicos del DASNA. Por esto necesaria la puesta en funcionamiento del sistema.

RECOMENDACIONES

La investigación realizada consumó todos los objetivos propuestos, donde su objetivo fundamental era el análisis y diseño de la aplicación web por lo que se recomienda continuar con la etapa de implementación y de esta forma obtener un sistema que sea beneficioso para la Formación Emergente de Auditores.

BIBLIOGRAFÍA CONSULTADA

1. **2007.** [En línea] 2007. <http://www.facilnet.net/matriz/web2/apache.html>.
2. **2006.** CECAM. [En línea] 2006. <http://www.cecam.sld.cu/pages/desarrollo/gestionuniv.htm>.
3. **2008.** Grupo CF developer. [En línea] 2008. <http://www.grupocfdeveloper.com/modulos.htm>.
4. **2007.** Grupo CF Developer. [En línea] 2007. <http://www.grupocfdeveloper.com/productos.htm>.
5. **2007.** Grupo de Usuario Rational del Perú. [En línea] 2007. <http://prug.solucionesracionales.com/node/12>.
6. **2004.** Informatizate. [En línea] María A. Mendoza Sanchez, 7 de Junio de 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
7. **2008.** IRCA. [En línea] Enero de 2008. <http://spain.irca.org/auditortrain.html>.
8. libros web. [En línea] http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
9. **Moreno, Gerardo Martínez. 2002.** *Ingeniería de Software UML*. 2002.
10. **Rambaugh, Ivar Jacobson, Grady, James y Grady, Booch. 2000.** *El Lenguaje Unificado de Modelado*. Madrid : Addison Wesley, 2000.
11. **Robles Prado, Javier Thomas y Turienzo Fernández, Raul.** <http://www.programacion.com>. <http://www.programacion.com>. [En línea] [Citado el: 8 de Febrero de 2008.] <http://www.programacion.com>.
12. **Thomas, Alex. 2005.** <http://www.noticias.com>. <http://www.noticias.com>. [En línea] 12 de Junio de 2005. [Citado el: 30 de Noviembre de 2007.] <http://www.noticias.com/articulo/12-06-2005/alex-tomas/por-que-elegir-php-4i92.html>.

BIBLIOGRAFÍA

1. **2007.** Desarrollo Web. [En línea] 2007. <http://vargasti.wordpress.com/2007/08/06/librería-extjs/>.
2. **2007.** El código. [En línea] 2007. <http://www.elcodigo.net/tutoriales/javascript/javascript1.html#punto1>.
3. **2006.** Jordisan.net. [En línea] 2006 Jordisan. <http://jordisan.net/modules/wordpress/2006/que-es-un-framework/>.
4. libros web. [En línea] http://www.librosweb.es/symfony/capitulo2/el_patron_mvc.html.
5. **2007.** Maestros del Web. [En línea] 6 de septiembre de 2007. <http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
6. **2007.** Maestros del Web. [En línea] 31 de julio de 2007. <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>.
7. **2004.** Master Magazine. [En línea] 2004. <http://www.mastermagazine.info/termino/4184.php>.
8. **2008.** Miller-group. [En línea] 2008. <http://www.miller-group.net/>.
9. **2000.** Proyecto S.B.O.L. [En línea] 2000. <http://www.sobl.org/traduccion/practical-postgres/node19.html>.
10. Usabilidadweb.com.ar. [En línea] <http://www.usabilidadweb.com.ar/postgre.php>.
11. **2005.** World Wide Ewb. [En línea] 2005. <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>.
12. **2006.** BIBLIOGRAPHY **González, Diego Andrés Asenjo.** *Patrones de Diseño, Marco Teórico Introdutorio.* 2006.
13. **2006.** Patrones de diseño. [En línea] 2006. http://www.infor.uva.es/~felix/datos/priiii/tr_patrones-2x4.pdf.
14. **2006.** **Carlos E. Cuesta.** Patrones de Diseño GoF. [En línea] 2006. <http://www.kybele.etsii.urjc.es/docencia/IS4/2007-2008/Material/Patrones%20de%20Diseno.pdf>.
15. **2008.** **León Welicki .** Microsoft. [En línea] 2008. http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_4081/default.aspx.
16. **2005.** Estado Beta. [En línea] 2005. <http://www.estadobeta.com/2006/05/02/active-record>.

17. **2007**. Ewb 2.0 Development. [En línea] 16 de Mayo de 2007.
<http://web2development.blogspot.com/2007/05/patron-activetable-y-activerrecord.html>.
18. **2007**. Libros Web. [En línea] 2007.
http://www.librosweb.es/symfony/capitulo1/conceptos_basicos.html#mapeo_de_objetos_a_bas_es_de_datos_orm.

ANEXOS

Anexo #1 Descripción de Casos de Uso del Sistema:

CU. Gestionar Confirmación de Matrícula

Caso de Uso:	Gestionar confirmación de matrícula	
Actores:	Técnico DASNA	
Resumen:	Este caso de uso se inicia cuando el técnico accede al sistema con la intención de confirmar la matrícula en FEA y realizar la re-matrícula en SUM. Cuando selecciona la opción de matrícula es para los estudiantes que terminan el curso de FEA y entran a la SUM. Si selecciona la opción de re-matrícula es para pasar de año a los que están en la SUM.	
Precondiciones:	El estudiante debe de estar matriculado en FEA o en SUM.	
Referencias	RF21, RF22.	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El técnico DASNA escoge la opción Confirmar matrícula.	2. El sistema muestra 2 opciones: 2.1 Confirmar matrícula en FEA se va a la sección "Confirmar matrícula". 2.2 Confirmar re-matrícula en SUM se va a la sección "Confirmar re-matrícula". Muestra criterios de búsquedas con la opción - Buscar.	
Sección "Confirmar matrícula"		
Acción del Actor	Respuesta del Sistema	

3. El técnico DASNA selecciona la opción Confirmar matrícula en FEA.	4. El sistema muestra una lista de todos los estudiantes matriculados en FEA y la opción - Confirmar matrícula
5. El técnico provincial busca al estudiante según los criterios de búsqueda.	6. El sistema muestra los estudiantes que correspondan con el criterio de búsqueda seleccionado.
7. El técnico DASNA escoge el estudiante al cual desea confirmar su matrícula en la FEA y la opción de Confirmar matrícula.	8. El sistema muestra un formulario con los datos del estudiante seleccionado y las opciones - Confirmar matrícula - Cancelar.
9. El técnico DASNA selecciona la opción Confirmar matrícula.	10. El sistema realiza la acción y actualiza la matrícula del estudiante.
Flujos Alternos 6a” El sistema no encontró los estudiantes”	
Acción del Actor	Respuesta del Sistema
	6a1. El sistema muestra que no se encontraron estudiantes según los criterios de búsqueda. Retorna a 3.
Flujos Alternos 8a ”El técnico selecciona la opción Cancelar ”	
Acción del Actor	Respuesta del Sistema
	8a1. El sistema cierra la página actual y retorna a 3.
Prototipo de Interfaz de Usuario	

Confirmación de matrículas

Nombre

Provincia

Apellido

Curso FEA

Buscar

Confirmar Matricula en FEA

Confirmar Matricula

Nombre	Apellidos	C.Identidad	D.Particular	Provincia	Municipio
Pedro Luis	Lazo Lazo	78120745654	Ave 22 #44	C. Habana	Cerro
Angel	Rodriguez	4561237891	Ave 25 No 2546	C. Habana	Playa
Marieta	Pozo Alegre	80042012124	San Bartolomé	Granma	Manzanillo
Yulieski	Betancourt	84062589523	Babiney	Cienfuegos	Abreus
Mael Angel	Rodriguez	77102052014	C 1ra edif B4	Cienfuegos	Rodas
Raul	Blanco Blanco	80081278489	C principal #4	Cienfuegos	Rodas
Ana Luisa	Kumicova	79100266594	San Rafael	Granma	Bayamo
Ana del Carmen	Guevarra	84020224474	Las Casitas #6	C. Habana	Habana Vieja
Mark	Guerra Guerra	84010120356	Ave 23 apto 44D	C. Habana	Plaza

Anterior << 1,2,3,4,5,6,7,8,9,10 >> Siguiente

Confirmación de Matrícula	
Nombre:	Angel
Primer Apellido:	Rodriguez
Segundo Apellido:	Medina
C.Identity:	4561237891
Dirección:	Ave 25 No 2546
Curso FEA:	2006 – 2007
Sexo:	Masculino
Nivel Escolar:	Medio Superior
Quiere SUM:	Si
<input type="button" value="Confirmar matrícula"/> <input type="button" value="Cancelar"/>	

Sección “Confirmar re-matrícula”

Acción del Actor	Respuesta del Sistema
3. El técnico DASNA selecciona la opción Confirmar re-matrícula en SUM	4. El sistema muestra una lista con todos los matriculados en SUM. Y la opción - Confirmar re-matrícula
5. El técnico provincial busca al estudiante según los criterios de búsqueda.	6. El sistema muestra los estudiantes que correspondan con el criterio de búsqueda seleccionado.
7. El técnico DASNA escoge el estudiante al cual desea re-matricular en SUM y la opción Confirmar re-matrícula.	8. El sistema muestra los detalles del estudiante para verificar sus datos con la opción de - Confirmar re-matrícula

	- Cancelar.
9. El técnico DASNA selecciona la opción Confirmar re-matrícula.	10. El sistema realiza la acción y guarda los datos del estudiante seleccionado.
Flujos Alternos 6a” El sistema no encontró los estudiantes”	
Acción del Actor	Respuesta del Sistema
	6a1. El sistema muestra que no se encontraron estudiantes según los criterios de búsqueda. Retorna a 3.
Flujos Alternos 8a ”El técnico DASNA selecciona la opción Cancelar ”	
Acción del Actor	Respuesta del Sistema
	8a1. El sistema cierra la página actual y retorna 3
Prototipo de Interfaz de Usuario	

Confirmación de Re-matricula

Nombre

Provincia

Apellido

Año

Buscar

Confirmar Re-Matricula en FEA

Confirmar Re-Matricula

Nombre	Apellidos	C.Identidad	D.Particular	Provincia	Municipio
Pedro Luis	Lazo Lazo	78120745654	Ave 22 #44	C. Habana	Cerro
Angel	Rodriguez	4561237891	Ave 25 No 2546	C. Habana	Playa
Marieta	Pozo Alegre	80042012124	San Bartolomé	Granma	Manzanillo
Yulieski	Betancourt	84062589523	Babiney	Cienfuegos	Abreus
Mael Angel	Rodríguez	77102052014	C 1ra edif B4	Cienfuegos	Rodas
Raul	Blanco Blanco	80081278489	C principal #4	Cienfuegos	Rodas
Ana Luisa	Kumicova	79100266594	San Rafael	Granma	Bayamo
Ana del Carmen	Guevarra	84020224474	Las Casitas #6	C. Habana	Habana Vieja
Mark	Guerra Guerra	84010120356	Ave 23 apto 44D	C. Habana	Plaza

Anterior << 1,2,3,4,5,6,7,8,9,10 >> Siguiente

Datos del estudiante

Nombre:	Luis
Primer Apellido:	Rodriguez
Segundo Apellido:	Castro
C. Identidad:	123456789
Dirección:	Ave 25 No2546
Provincia:	C. Habana
Sexo:	Masculino
Nivel Escolar:	Medio Superior
Carrera:	Contabilidad
Año:	Primero

Anexo #2. Prototipos de Interfaz de Usuario

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL

Sábado 24 de Mayo del 2008 Inicio Ayuda Salir

MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

Matrícula en FEA

Principal de FEA

Informe general

Matrícula en SUM

Principal de SUM

Pasar estudiante de año

Estudiantes por año estudio

Confirmación de Matrícula

Confirmación de matrícula en FEA

Confirmación de re-matrícula en SUM

Buscar Informes

Histórico de Graduados

General de Habilitados

Gestionar matrícula en FEA






No.	Nombre y apellidos	Carnet de identidad	Curso FEA	Provincia	Categoría	Fecha de matrícula
1	daiamna pi	85654233	2005-2006	Habana	Estudiante	20/09/2006
2	Edgar Blanco	45678958	2005-2006	Matanzas	Estudiante	20/09/2005a
3	daiamna pi	85654233	2005-2006	Habana	Estudiante	20/09/2006
4	Edgar Blanco	45678958	2005-2006	Matanzas	Estudiante	20/09/2005a
5	daiamna pi	85654233	2005-2006	Habana	Estudiante	20/09/2006

<< Anterior Siguiente >>

Fig. A-1. Gestionar Matrícula en FEA

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL

Sábado 24 de Mayo del 2008 Inicio Ayuda Salir

MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

Matrícula en FEA

Principal de FEA

Informe general

Matrícula en SUM

Confirmación de Matrícula

Buscar Informes

Histórico de Graduados

General de Habilitados

Matricular estudiante

Datos del estudiante

Nombre:	<input type="text"/>	Provincia:	--Seleccione--
Apellidos:	<input type="text"/>	Municipio:	--Seleccione--
Carnet de identidad:	<input type="text"/>	Nivel escolar:	--Seleccione--
Sexo:	--Seleccione--	Quiere SUM:	--Seleccione--
Dirección:	<input type="text"/>	Fecha de matrícula:	2008-06-18 ...
Curso FEA:	<input type="text"/>		
Categoría:	<input type="radio"/> Estudiante <input type="radio"/> Profesor		

Aceptar Cancelar

Fig. A-2. Matricular estudiante en FEA.

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL

Sábado 24 de Mayo del 2008 Inicio? Ayuda? Salir

MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

- Matrícula en FEA
- Matrícula en SUM
 - Principal de SUM
 - Pasar estudiante de año
 - Estudiantes por año estudio
- Confirmación de Matrícula
 - Confirmación de matrícula en FEA
 - Confirmación de re-matrícula en SUM
- Buscar Informes
- Histórico de Graduados
- General de Habilitados

Pasar estudiante de año

Seleccione el año

Año:

Resultado de la búsqueda

No.	Nombre y apellidos	Carnet de identidad	Sexo	Provincia	Municipio	Seleccionar
1	Daiamna Pi	85654233	F	Habana	San jose	<input checked="" type="checkbox"/>
2	Vladimir Rodriguez	1234568	M	Matanzas	Cárdenas	<input checked="" type="checkbox"/>
3	Leticia Martinez	4567893	F	Santiago de Cuba	Palma Soriano	<input checked="" type="checkbox"/>
4	Daiamna Pi	85654233	F	Habana	San jose	<input checked="" type="checkbox"/>
5	Vladimir Rodriguez	1234568	M	Matanzas	Cárdenas	<input checked="" type="checkbox"/>

<< Anterior >>

Fig. A-3. Pasar estudiante de año en SUM

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL

Sábado 24 de Mayo del 2008 Inicio Ayuda Salir

MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

- Matrícula en FEA
 - Principal de FEA
 - Informe general**
- Matrícula en SUM
- Confirmación de Matrícula
- Buscar Informes
- Histórico de Graduados
- General de Habilitados

Informe general

No.	Nombre y apellidos	Carnet de identidad	Sexo	Dir particular	Curso FEA	Provincia	Municipio	Categoría	Nivel escolar	Fecha de matrícula
1	daiamna pi	85654233	F	ave 25 n 3456a	2005/2006	Habana	San jose	estudiante	tecnico	20/09/2006
2	daiamna pi	85654233	F	ave 25 n 3456a	2005/2006	Habana	San jose	estudiante	tecnico	20/09/2006
3	daiamna pi	85654233	F	ave 25 n 3456a	2005/2006	Habana	San jose	estudiante	tecnico	20/09/2006
4	daiamna pi	85654233	F	ave 25 n 3456a	2005/2006	Habana	San jose	estudiante	tecnico	20/09/2006
5	daiamna pi	85654233	F	ave 25 n 3456a	2005/2006	Habana	San jose	estudiante	tecnico	20/09/2006

<< Anterior Siguiente >>

Pasar de año Cancelar

Fig. A-4. Informe General de matrículas en FEA.

The screenshot shows the SIGAC (SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL) web application. The header includes the date 'Sábado 24 de Mayo del 2008' and navigation links for 'Inicio', 'Ayuda', and 'Salir'. The left sidebar contains a menu with options like 'DASNA', 'Matrícula en FEA', 'Matrícula en SUM', and 'General de Habilitados'. The main content area is titled 'Insertar Pre ubicación laboral' and contains a form with three input fields: 'Nombre de la entidad', 'Nombre del organismo', and 'Plaza'. Below the form are two buttons: 'Aceptar' and 'Eliminar'.

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL

Sábado 24 de Mayo del 2008 Inicio Ayuda Salir

MINISTERIO DE AUDITORÍA Y CONTROL

Insertar Pre ubicación laboral

Datos de la ubicación laboral

Nombre de la entidad:

Nombre del organismo:

Plaza:

Fig. A-5. Insertar pre ubicación laboral

Sábado 24 de Mayo del 2008 Inicio? Ayuda? Salir

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

- ▶ Matrícula en FEA
- ▶ Matrícula en SUM
- ▼ Confirmación de Matrícula
 - Confirmación de matrícula en FEA
 - Confirmación de re-matrícula en SUM
- ▼ Buscar Informes
 - Baja en FEA
 - Baja por año en SUM
 - Confirmación de matrícula
 - Confirmación de re-matrícula
 - Ubicación Laboral Nacional
 - ▼ Histórico de Graduados
 - Graduados por Curso
 - Graduados por organismo
 - ▶ General de Habilitados

Graduados por cursos

Seleccione Curso FEA

Curso FEA:

Resultado de la búsqueda

No.	Nombre y apellidos	Carnet de identidad	SUM donde curso	Ubicación laboral	Provincia	Municipio
1	Daiamna Pi	85654233	2005-2006	Habana	Estudiante	20/09/2005
2	Jorge Moro	45696321	2005-2006	Las Tunas	Estudiante	20/09/2005
3	Eduardo García	5698322	2005-2006	Holguín	Estudiante	20/09/2005
4	Daiamna Pi	85654233	2005-2006	Habana	Estudiante	20/09/2005
5	Jorge Moro	45696321	2005-2006	Las Tunas	Estudiante	20/09/2005

Fig. A-6. Estudiantes Graduados por Curso FEA.

SIGAC SISTEMA INFORMÁTICO DE GESTIÓN DE AUDITORÍA Y CONTROL Sábado 24 de Mayo del 2008 Inicio Ayuda Salir

MINISTERIO DE AUDITORÍA Y CONTROL

DASNA

- Matrícula en FEA
 - Principal de FEA
 - Informe general
- Matrícula en SUM
 - Principal de SUM
 - Pasar estudiante de año
 - Estudiantes por año estudio
- Confirmación de Matrícula
- Buscar Informes
- Histórico de Graduados
 - Graduados por Curso
 - Graduados por organismo
- General de Habilitados

Estudiantes por año de estudio en SUM

Seleccione el año

Año:

Resultado de la búsqueda

No.	Nombre y apellidos	Carnet de identidad	Provincia	Municipio	Categoría	Fecha de matrícula
1	Daiamna Pi	85654233	2005-2006	La Habana	Estudiante	20/09/2005
2	Rafael Garcia	54487454132	2005-2006	Matanzas	Estudiante	20/09/2005
3	Oscar Rodriguez	4569874	2005-2006	Ciudad Habana	Estudiante	20/09/2005
4	Daiamna Pi	85654233	2005-2006	La Habana	Estudiante	20/09/2005
5	Rafael Garcia	54487454132	2005-2006	Matanzas	Estudiante	20/09/2005

<< Anterior >>

Fig. A-7. Mostrar estudiantes por año de estudio en SUM

GLOSARIO DE TÉRMINOS.

- ❖ **MAC:** Ministerio de Auditoría y Control.
- ❖ **DASNA:** Dirección de atención al Sistema Nacional de Auditoría.
- ❖ **FEA:** Formación Emergente de Auditores.
- ❖ **SUM:** Sede Universitaria Municipal
- ❖ **CU:** Caso de Uso.
- ❖ **RF:** Requisito funcional
- ❖ **RNF:** Requisito no funcional
- ❖ **DS:** Diagrama de secuencia.
- ❖ **Técnicos del DASNA:** empleado del Ministerio de Auditoría y Control encargado de realizar los informes y reportes del mismo.
- ❖ **Técnico de la delegación provincial:** empleado del Ministerio de Auditoría y Control encargado de realizar los procesos de matrícula en las sedes provinciales universitarias y en el programa de formación emergente de auditores.
- ❖ **Auditor:** persona capacitada y experimentada que se designa por una autoridad competente, para revisar, examinar y evaluar los resultados de la gestión administrativa y financiera de una dependencia o entidad con el propósito de informar o dictaminar acerca de ellas, realizando las observaciones y recomendaciones pertinentes para mejorar su eficacia y eficiencia en su desempeño.
- ❖ **Modelo CH1:** informe que recoge los datos de los estudiantes que realizan su matrícula en FEA por provincia.
- ❖ **Modelo CH5:** informe general de los estudiantes que realizan matrícula en SUM
- ❖ **Modelo CH5A:** informe de matrícula en SUM actualizado con los datos de los estudiantes que han causado baja.
- ❖ **Modelo CH8:** Informe que contiene los datos de los estudiantes que se encuentran trabajando.
- ❖ **Modelo CH9:** Informe que contiene los datos de los de graduados por organismo en FEA.
- ❖ **Modelo CH7:** Informe con los datos y la cantidad de estudiantes graduados en una provincia.
- ❖ **MVC:** Patrón Modelo Vista Controlador.
- ❖ **Estudiantes habilitados:** estudiantes que ocupan plaza de auditor y se encuentran trabajando.
- ❖ **CECAM:** Centro de Cibernética Aplicado a la Medicina.

- ❖ **IRCA:** Registro Internacional de Auditores Certificados.
- ❖ **IBM:** empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.
- ❖ **OMG:** estándar de descripción de esquemas de software.
- ❖ **LDAP:** es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
- ❖ **CVS:** Sistema Concurrente de Versiones.
- ❖ **KDE:** Es un entorno de Escritorio contemporáneo para estaciones de trabajo Unix.
- ❖ **ERD:** Diagramas entidad relación.
- ❖ **ORM:** Mapeo objeto-relacional.
- ❖ **API:** Interfaz de Programación de Aplicaciones.
- ❖ **CD:** Clases del diseño