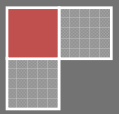
SISTEMA DE GESTIÓN DE LA INFORMACIÓN DE LOS ESTUDIANTES GUYANESES EN CUBA.

Trabajo de Diploma para optar para el Título de Ingeniero de Informática.

Universidad de Cienfuegos “Carlos Raphael Rodríguez”
Facultad de Informática.

Autor: Alaric Abraham
Tutor: Msc. Ivan Fuentes Jova
Cienfuegos, Cuba. 2008



DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Oficina de Los Asuntos Estudiantiles de la Embajada de Guyana y al Departamento de Informática de la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que hagan el uso que estimen pertinente con el trabajo de diploma.

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

___Alaric Abraham___

Nombre del autor

__Msc: Ivan Fuentes Jova_____

Nombre del tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Tutor

Nombre y Firma de ICT

Firma Vicedecano

OPINIÓN DEL USUARIO.

El Trabajo de Diploma, “Sistema de Gestión de la información de los estudiantes guyaneses en Cuba” fue realizada en nuestro entidad Oficina de Asuntos Estudiantiles de la Embajada de Guyana. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado nos satisface:

- ☐ Totalmente
- ☐ Parcialmente en un ____ %

EL resultado obtenido permite a esta entidad contar con una herramienta que beneficia el desarrollo del proceso de gestión de la información de los estudiantes guyaneses, facilitando la labor de los oficiales de esta oficina. Con la implementación del sistema se logran mejores en la comunicación, accesibilidad y gestión de la información referente a los estudiantes guyaneses en Cuba.

Para que así conste, se firma la presente a los __ días del mes de Junio del año 2008.

_____ _____

Nombre del representante de la entidad Cargo

_____

Firma Cuño

DEDICATORIA

A mis padres.

AGRADECIMIENTOS

- ✓ *A Dios, por su misericordia durante los años, pues sin él nada es posible.*
- ✓ *A mi familia, a quien todo le debo, por su constante preocupación y amor infinitos. Ningún agradecimiento será suficiente.*
- ✓ *A mis amigos.*
- ✓ *A la Revolución Cubana, le agradezco mucho por darme la oportunidad de realizar mis estudios.*
- ✓ *A mi tutor, MSc. Iván Fuentes Jova.*

A todos, muchísimas gracias.

--- Alaric---

RESUMEN

La oficina de los asuntos estudiantiles (Student Affairs Office) es una entidad que se encarga de los asuntos de estudiantes guyaneses que cursan estudios en Cuba, responde a la Embajada de Guyana en La Habana, Cuba y al Ministerio de Servicios Públicos en Guyana.

El software que se implementa surgió dada la necesidad de facilitar la gestión y control de los datos de los estudiantes guyaneses que estudian en Cuba en la Oficina de Asuntos Estudiantiles (SAO por sus siglas en ingles), perteneciente a la Embajada de la República Cooperativa de Guyana en la República de Cuba, debido a que en la actualidad este proceso se lleva a cabo de forma manual, lo que dificulta en gran medida el trabajo y la agilidad de los procedimientos que se llevan a cabo en la SAO. El sistema pretende, además, brindar a todos los estudiantes una vía de comunicación, tanto entre ellos como con su embajada y conservar un índice histórico de los datos de todos los jóvenes guyaneses que estudian en Cuba con vistas al posterior análisis de los beneficios que reporta la cooperación educacional entre ambas repúblicas con vistas al fortalecimiento futuro de dichos vínculos.

Atendiendo a las peticiones de los usuarios, se decidió resolver el problema mediante la implementación de un sitio web dinámico que centralice toda la información del curso de estudios en Cuba.

Se espera que el sitio web sea accesible por todos a través de Internet, que publique y gestione toda la información y datos referentes a los estudiantes guyaneses en Cuba.

Para la confección del sitio se empleó como gestor de base de datos MYSQL, como un servidor se empleo Apache, como sistema de autor Adobe Dreamweaver CS3, para el diseño visual y como lenguaje de programación PHP con un editor de código PHP se utilizo ZEND Studio 5.5.

Se empleó el lenguaje de modelación unificado UML (Unified Modeling Language) como metodología y la herramienta Rational Rose para el modelado visual de la documentación del software.

Índice

INTRODUCCIÓN ---- **1**

CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA ---- **4**

1.1 INTRODUCCIÓN ---- 4
1.2 DESCRIPCIÓN DEL PROBLEMA. ---- 5
1.2.1 Conceptos asociados al dominio del problema ---- *5*
1.3 DESCRIPCIÓN DEL OBJETO DE ESTUDIO ---- 6
1.3.1 Objetivos estratégicos de la organización. ---- *6*
1.3.2 Flujo actual de los procesos ---- *6*
1.3.3 Análisis crítico de la ejecución de los procesos ---- *7*
1.3.4 Procesos objeto de automatización ---- *7*
1.4 DESCRIPCIÓN DE LOS SISTEMAS EXISTENTES ---- 8
1.5 TENDENCIAS, METODOLOGÍAS Y/O TECNOLOGÍAS ACTUALES ---- 9
Tendencias y tecnologías actuales. ---- *9*
Tecnologías del lado del cliente. ---- *12*
Tecnologías del lado del servidor. ---- *17*
Scripts del lado del servidor. ---- *19*
Gestión de Datos. ---- *22*
Seguridad ---- *25*
Metodología. ---- *26*
Herramientas de desarrollo. ---- *27*
1.6 CONCLUSIONES ---- 29

CAPITULO – 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA ---- **30**

2.1 INTRODUCCIÓN. ---- 30
2.2 DESCRIPCIÓN DEL MODELO DEL NEGOCIO. ---- 30
2.3 REGLAS DEL NEGOCIO A CONSIDERAR. ---- 32
Generales. ---- *32*
Particulares. ---- *32*
2.4 – MODELO DE CASOS DE USO DEL NEGOCIO ---- 33
2.4.1 Actores del negocio. ---- *33*
2.4.2 Trabajadores del Negocio. ---- *34*
2.4.3 Diagramas de los Casos de Uso del Negocio. ---- *35*
2.4.4 Descripción de los Casos de Uso. ---- *36*
2.5 DIAGRAMAS DE ACTIVIDADES ---- 47
2.6 DIAGRAMA DE OBJETOS DEL NEGOCIO ---- 47
2.7 REQUERIMIENTOS FUNCIONALES ---- 48
2.7.1 – INTRODUCCIÓN. ---- 48
2.7.2 – DESCRIPCIÓN DEL SISTEMA PROPUESTO. ---- 49
2.7.2 – Concepción general del sistema. ---- *49*
2.7.3 – Requerimientos funcionales. ---- *49*
2.7.4– Requerimientos no funcionales. ---- *51*
2.8 MODELO DE CASOS DE UO DEL SISTEMA. ---- 55
2.8.1 Actores del sistema. ---- *55*
2.8.2 Casos de Uso del sistema. ---- *57*
2.8.3 Paquetes y sus relaciones. ---- *59*

2.8.4 Diagramas de casos de uso del sistema. ---- *59*
2.8.5 Descripción de los casos de uso del sistema. ---- *60*
2.9 CONCLUSIONES ---- 81

CAPÍTULO 3 – CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA ---- **82**

3.1 – INTRODUCCIÓN ---- 82
3.2 – DIAGRAMA DE CLASES DEL DISEÑO ---- 83
3.3 – DISEÑO DE LOS DATOS ---- 84
3.3.1 – Modelo lógico de datos ---- *84*
3.3.2 – Modelo físico de datos ---- *84*
3.4 – MODELO DE IMPLEMENTACIÓN ---- 85
3.5 – DIAGRAMA DE DESPLIEGUE ---- 86
3.6 – PRINCIPIOS DE DISEÑO ---- 87
3.6.1 – Estándares en la interfaz de la aplicación ---- *87*
3.6.2 – Tratamiento de errores ---- *88*
3.7 – CONCLUSIONES ---- 88

CAPITULO IV – ESTUDIO DE FACTIBILIDAD ---- **89**

4.1 – INTRODUCCIÓN ---- 89
4.2 – PLANIFICACIÓN POR CASOS DE USOS. ---- 90
4.3 – DETERMINACIÓN DE LOS COSTOS. ---- 96
4.4 – BENEFICIOS TANGIBLES E INTANGIBLES. ---- 98
4.5 – ANÁLISIS DE COSTOS Y BENEFICIOS ---- 98
4.6 – CONCLUSIONES ---- 99

CONCLUSIONES ---- **100**

RECOMENDACIONES ---- **101**

REFERENCIAS BIBLIOGRAFÍAS ---- **102**

BIBLIOGRAFÍA ---- **105**

GLOSARIO DE TÉRMINOS ---- **106**

ANEXOS ---- **108**

Diagramas de actividades ---- *108*
Diagramas de los casos del uso del sistema ---- *113*
Prototipos del sistema ---- *116*
Diagramas de clases web ---- *133*
Modelo lógico de datos ---- *140*
Modelo físico de datos ---- *141*

INTRODUCCIÓN

El gran crecimiento de Internet, ha incrementado la cantidad de personas en el mundo que utilizan este recurso para mejorar la comunicación, facilitando el acceso a la información existente. Actualmente Internet se ha convertido en el medio más utilizado para difundir todo tipo de información en cualquier lugar del mundo. Esta potencialidad es aprovechada cada vez más por las organizaciones, pues tener publicado todo lo referente a sus actividades, estructura y relaciones mediante sitios Web, constituye un elemento trascendental para mejorar su nivel de promoción.

La República de Cuba, como parte de su programa integral de cooperación internacional, en especial con los países subdesarrollados del mundo, ha llevado a cabo un programa de formación general integral y gratuita de jóvenes en sus instituciones, en todas las carreras universitarias que se imparten en el país, incluyendo la formación de médicos y deportistas para todo el mundo. Dicho programa tuvo desde sus inicios cientos de países vinculados a él, los cuales sienten su apoyo y participación en el programa, basados en el hecho de que la educación cubana es una de las que más calidad posee a nivel internacional y en que se obtienen múltiples beneficios de este proyecto tanto para Cuba, que fortalece así sus relaciones internacionales, como para los demás países, que logran con esto la formación de profesionales totalmente capacitados para contribuir al desarrollo económico, político y social de sus naciones.

Uno de los países que desde el inicio de sus relaciones con la República de Cuba ha sido beneficiado con este programa integral de cooperación educacional, es la República Cooperativa de Guyana, la cual es una de las naciones subdesarrolladas del mundo, que como Cuba, luchan por alcanzar cada día mejores posiciones en la economía mundial, para lo cual, se necesita de profesionales preparados y dispuestos a colocar a su país en los mejores lugares a nivel mundial de los mercados internacionales en los que participa.

Los estudiantes que llegan a Cuba, de Guyana poseen entonces la posibilidad de tener una sede en La Habana, Cuba, para establecer el vínculo directo con su país y sus familiares, ya que existe en la Embajada de la República Cooperativa de Guyana en Cuba, la oficina de Asuntos Estudiantiles (SAO por sus siglas en ingles) que se ocupa

de coordinar y gestionar todos aquellos asuntos que tienen que ver con los estudiantes, además de controlar sus datos personales y académicos.

El trabajo de la oficina, ha resultado exitoso, no obstante sus directivos se han dado cuenta de que existen problemas que afectan el buen funcionamiento de la organización, tales como:

1. Poca divulgación de la información del programa de estudios en Cuba a nivel internacional.
2. Limitada comunicación entre los estudiantes y los funcionarios de la oficina de asuntos estudiantiles.
3. La información referente a los estudiantes está almacenada en ficheros Excel poco confiables.
4. La búsqueda de los datos de los estudiantes resulta trabajosa por la manera en que está almacenada la información.
5. La creación de diferentes reportes acapara gran parte del tiempo del oficial de asuntos estudiantiles por lo que no dispone suficiente tiempo para cumplir el resto de sus funciones.

De manera general resulta evidente que se necesita implementar un mecanismo que pueda resolver estos problemas. Partiendo de ello, se consideró adecuado estudiar temas relacionados con la divulgación y algunos elementos de soporte técnico utilizando soluciones Web como medio ampliamente difundido para lograr mayor nivel divulgativo de información a nivel mundial.

Anteriormente la oficina de asuntos estudiantiles no contaba con ninguna herramienta que solucionara los problemas mencionados.

Para solucionar el problema planteado se define el siguiente Objetivo General:

Diseñar e implementar un sitio web dinámico para la Oficina de Asuntos Estudiantiles que permita manejar de forma rápida y eficiente toda la información de esa oficina.

Del objetivo general se derivan los siguientes objetivos específicos:

1. Realizar un estudio de las tendencias y tecnologías actuales del campo de la Informática, determinando cuáles utilizar en la solución del problema.
2. Construir el sistema propuesto.

3. Crear una base de datos que garantice el almacenamiento, seguridad e integridad de la información.

Se espera que la implementación de la solución Web contribuya a la divulgación de las funciones y actividades de la oficina de asuntos estudiantiles.

El documento se estructura en 4 capítulos, representándose todo lo relacionado con el trabajo investigativo realizado, así como la solución al problema planteado.

El Primer capítulo, “Fundamentación Teórica”, está dedicado al estudio de los sitios Web, su estructura, la tecnología usada tanto por el cliente, como por el servidor para su programación, etc. Los aspectos tratados en ese capítulo ayudarán a comprender el por qué de la selección de un sitio Web dinámico como solución al problema planteado.

El Segundo capítulo, Descripción de la solución propuesta. Se describe el modelo del negocio, el entorno de trabajo en que se desarrolla el sistema, mostrando los casos de uso, los diagramas de actividad, diagrama de clase del modelo de objetos, requerimientos funcionales y no funcionales, descripción del sistema propuesto y los casos de uso del sistema. Todo esto se hizo con el objetivo de entender la estructura y la dinámica de la organización, así como los problemas existentes en el negocio.

También el segundo capítulo está dedicado a enumerar los requerimientos no funcionales. Define las características del software propuesto mediante la definición de los actores del sistema, los paquetes y sus relaciones y la explicación y descripción del los casos de uso, enunciándose en qué ciclo se implementará cada uno.

El Tercero capítulo, “Descripción de la Solución Propuesta”, muestra el diagrama de clases correspondiente al sitio Web y define también el modelo lógico y físico de los datos. Son especificados aspectos relacionados con el diseño de la interfaz, estándares utilizados y el tratamiento de errores.

El Cuarto capítulo, “Estudio de Factibilidad”, hace un análisis de las variables tiempo, costo y esfuerzo a través de su estimación mediante la variante “Análisis de Puntos de Casos de Uso”, se enuncian además los beneficios tangibles e intangibles que aporta el sistema.

CAPÍTULO I – FUNDAMENTACIÓN TEÓRICA

1.1 INTRODUCCIÓN

En el presente capítulo se brinda una visión general de los aspectos relacionados con un sistema de gestión de gastos y una visión específica de los conceptos asociados al dominio de este.

Se abordarán temas tales como: la definición de un sitio web, su estructura, la tecnología usada tanto por el cliente como por el servidor para su programación, etc. Los aspectos tratados en este capítulo ayudarán a comprender el por qué de la selección de un sitio web dinámico como solución al problema planteado.

1.2 DESCRIPCIÓN DEL PROBLEMA.

1.2.1 Conceptos asociados al dominio del problema

Gestión

Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener. [1]

Gestión de la información

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y, posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. [1]

Los estudiantes extranjeros como parte del Sistema de Educación Superior Cubano.

A lo largo de muchos años al Sistema Educativo Cubano se incorporan de manera progresiva una cantidad elevada de estudiantes de diversos países, que proceden de culturas diferentes, sustentado por el prestigio del cual goza la educación cubana a nivel internacional. [2]

El programa especialista de becas de los gobiernos de Guyana/Cuba

El programa especialista de becas de los gobiernos de Guyana/Cuba se desarrolla como resultado de una visita a Cuba efectuada por el Presidente Bharrat Jagdeo en el año 2001, en la que el Presidente Jagdeo consolidó la concesión de becas en varias disciplinas, después de sus discusiones con el Presidente Fidel Castro.

Las becas se ofrecen en las ramas de Medicina, Ingeniería, Tecnología, Telecomunicaciones, Agricultura, Deportes y Cultura.

Actualmente, 843 estudiantes guyaneses estudian en Cuba, incluidos algunos que lo hacían con anterioridad al programa de especialistas de becas de los gobiernos de Guyana/Cuba.

1.3 DESCRIPCIÓN DEL OBJETO DE ESTUDIO

1.3.1 Objetivos estratégicos de la organización.

La entidad para la cual se lleva a cabo este trabajo es la Embajada de la República Cooperativa de Guyana en Cuba, la cual tiene como objetivo fundamental fomentar las relaciones existentes entre ambas naciones, así como gestionar todos los asuntos referentes a aquellos ciudadanos guyaneses que por una causa u otra se encuentren en la República de Cuba, ya sea como turistas, visitantes, residentes temporales o permanentes. La embajada se estructura de modo jerárquico, teniendo como máximo exponente al Sr. Embajador de la República Cooperativa de Guyana en Cuba, al que se subordinan varios departamentos o sectores que se ocupan de los asuntos antes mencionados.

El departamento que orientó y supervisó fundamentalmente la realización de este trabajo es la Oficina de Asuntos Estudiantiles (SAO por sus siglas en ingles), la cual está integrada por la Agregada para Asuntos Estudiantiles y el Asistente de esa oficina. Este departamento trata específicamente todo lo referente a los estudiantes guyaneses que estudian en Cuba, o sea, el control de su información, la gestión de permisos, la recogida de sus resultados académicos, etc.

1.3.2 Flujo actual de los procesos

Procedimientos del Sistema Actual:

1. Recogida de los datos personales de los estudiantes.
    - Después que los estudiantes se encuentran en la preparatoria, estos llenan una planilla con todos los datos que solicita la embajada, entre los cuales se cuenta un vínculo en su país que se pueda contactar en caso de emergencia.
    - El responsable de los asuntos estudiantiles de los guyaneses que cursan estudios en Cuba, recorre los centros donde reciben los cursos de preparatoria, y recoge las planillas.
    - El responsable de los asuntos estudiantiles de los guyaneses en Cuba, introduce todos sus datos en un documento de Excel.

2. Solicitud y gestión de permisos de los estudiantes guyaneses a la embajada.
    - El estudiante solicita permiso a la oficina de asuntos estudiantiles.
    - En caso de solicitud de salida por la ocurrencia de algún problema grave:
        - La embajada localiza al contacto de emergencia del estudiante en Guyana.
        - En caso de tratarse de un hecho grave (fallecimientos, accidentes, etc.) la embajada autoriza automáticamente al estudiante y paga su pasaje.
    - En caso de solicitud de salida por vacaciones:
        - La embajada gestiona con el centro estudiantil si el estudiante puede salir o no.
        - Si es autorizado, el estudiante paga su pasaje y sale del país.
        - Si no es autorizado y el estudiante se va, causa baja automáticamente del centro.
3. Generación de reportes estadísticos.

1.3.3 Análisis crítico de la ejecución de los procesos

No existe un sistema informático eficiente que facilite la gestión y control de los datos de los estudiantes guyaneses que estudian en Cuba. En la actualidad, la mayoría de este proceso se lleva a cabo de forma manual, lo que dificulta en gran medida el trabajo y la agilidad de los procedimientos que se llevan a cabo en la Oficina de Asuntos Estudiantiles.

La información referente a los estudiantes guyaneses en Cuba está almacenada en ficheros Excel poco confiables.

1.3.4 Procesos objetos de automatización

Los procesos objetos de automatización, son la gestión de los datos de los estudiantes, así como la actualización de la información referente a los estudiantes, eventos, noticias, etc. de manera que muestre la información lo más actualizada posible.

Así como generar reportes que son solicitados posteriormente por diversas entidades como es el caso del Ministerio de Servicios Públicos de Guyana (PSM por sus siglas en

ingles), Oficina del Presidente de Guyana (OP por sus siglas en ingles), y Ministerio de Educación Superior (MES), entre otras.

1.4 DESCRIPCIÓN DE LOS SISTEMAS EXISTENTES

A continuación se presentan dos sistemas informáticos que han sido desarrollados hasta la actualidad y están relacionados con este campo de acción:

SACBEXT: Sistema Automatizado de control de Becarios Extranjeros.

Fue desarrollado durante los años 1998-2000, para la manipulación de información de Estudiantes Extranjeros en la ONABE (Oficina Nacional de Atención a Becarios Extranjeros). El sistema gestiona en un alcance general la información de los estudiantes a nivel nacional, permitiendo así la eficiencia en la gestión de esta información). SACBEXT logró mejorar el proceso de control y la rapidez en la toma de decisiones por la ONABE. El sistema se destaca por las posibilidades brindadas para la obtención de resúmenes y reportes de la información almacenada y el envío por correo de los informes generados por el sistema. Las exigencias presentes en ese período, requirieron una gestión generalizada y resumida de la información, pues SACBEXT no profundizó en los parámetros que podrían diferenciar, y a la vez identificar con más precisión a los estudiantes extranjeros. La accesibilidad al sistema es limitada, ya que los Centros de Estudio donde radican los estudiantes y donde son más conocidos, no tienen acceso directo a esta herramienta por cuestiones de seguridad que no fueron implementadas, aunque en general se puede afirmar que SACBEXT ha dado buenos resultados y ha estado en uso hasta la actualidad.

BDEDUCACION: Base de Datos Educación.

Proyecto de reciente apertura que surgió de la intención de ofrecer mejoras a SACBEXT y que comenzó en el año 2003 con el objetivo de perfeccionar el proceso de gestión. BDEDUCACION rediseñó la estructura de la Base de Datos de Estudiantes Extranjeros (BDEE), introduciendo nuevos campos de información y algunos módulos. Supuestamente, esas modificaciones perfeccionarían el funcionamiento del sistema y darían nuevos resultados que permitirían una mejoría en la manipulación de la información. Sin embargo, no se logró terminar el proyecto pues se estancó el proceso de corrección de los problemas existentes en SACBEXT y el sistema no fue culminado.

1.5 TENDENCIAS, METODOLOGÍAS Y/O TECNOLOGÍAS ACTUALES

El estudio de las tecnologías actuales se ha convertido en uno de los factores claves en el desarrollo de cualquier sistema informático por muy sencillo que este resulte. Representa uno de los aspectos determinantes en el funcionamiento de las empresas e instituciones. En la medida en que el concepto calidad tecnológica sea mayor, se logrará alcanzar los objetivos trazados de una forma eficiente y efectiva. Se necesita realizar una correcta selección de las tecnologías a emplear, en dependencia de las necesidades y recursos propios con los que se cuenta. Una correcta selección permitirá lograr un producto final con calidad y eficiencia. [3]

En los epígrafes que vienen a continuación, se pretende brindar un recorrido informativo a través de las tecnologías que se encuentran alrededor de las requeridas para el desarrollo del sistema propuesto, analizando las ventajas y desventajas que las caracterizan, justificando así el por qué del lenguaje, gestor de base de datos y otros software utilizados. Además, se realiza un estudio de la metodología utilizada para el análisis y el diseño del sistema, entre otros aspectos.

Tendencias y tecnologías actuales.

¿Qué diferencias existen entre las aplicaciones de escritorio y las aplicaciones Web?

Una aplicación offline o aplicación de escritorio se ejecuta en el cliente, es decir en tu PC. Para iniciarla y hacerla funcionar requiere estar presente delante del ordenador que tiene instalada dicha aplicación. Un ejemplo seria programas como los de Microsoft Office (Excel, Word). Cualquier programa que instales en tu ordenador son aplicaciones offline.

Ventajas: Su ejecución no requiere habitualmente de comunicaciones con el exterior, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales.

Desventajas: Su acceso se limita al PC donde se instalan, son dependientes del sistema operativo que utilice tu ordenador y sus capacidades (video, memoria, etc.).

Una aplicación en línea o aplicación Web por el contrario reside en un servidor, y su ejecución requiere disponer de un PC con conexión a internet, un navegador como

Internet Explorer, Mozilla Firefox, Opera, etc.; y por supuesto que la aplicación funciona en el servidor que la aloja. Un ejemplo simple sería este Blog, donde usuarios presentan y actualizan información de forma remota a través de unos administradores Web, y otros usuarios acceden a ella a través de una interfaz Web: la aplicación reside en los servidores de WordPress, nosotros sólo accedemos a ella a través de un navegador.

Ventajas: proporcionan movilidad, dado que puedes ejecutarlas desde cualquier ordenador con conexión a internet. La información que manejan se accede a través de internet, motivo por el cual son especialmente interesantes para desarrollar aplicaciones multiusuario basadas en la compartición de información. El cliente o usuario que utiliza la aplicación no necesita tener un ordenador de grandes prestaciones para trabajar con ella.

Desventajas: La comunicación constante con el servidor que ejecuta la aplicación establece una dependencia con una buena conexión a internet. Además, el servidor debe tener las prestaciones necesarias para ejecutar la aplicación de manera fluida, no sólo para un usuario sino para todos los que la utilicen de forma concurrente.

Arquitectura Cliente-Servidor.

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

En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. El programa cliente gestiona la comunicación con el servidor y ofrece las herramientas necesarias para poder trabajar con este, mientras que el programa servidor se encarga de transmitir la información en la forma más adecuada para el usuario o usuarios, ya que un servidor admite múltiples accesos simultáneos. Los programas cliente y servidor pueden ser muy variados y funcionar sobre sistemas operativos diversos (UNIX, Windows NT, MS-DOS, OS/2, etc.). [4]

Los clientes interactúan con el servidor, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido y recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad. [4]

Entre las principales características de la arquitectura cliente / servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interface única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interface externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Arquitectura de N capas.

A lo largo de la historia del análisis y diseño de sistemas, ha quedado demostrado que subdividir un software en varias partes lógicas llámense módulos, paquetes o capas, resulta muy provechoso sobre todo a la hora de entender su filosofía y distribuir las tareas que ejecuta. Para enfrentarse a estos temas, la comunidad de software desarrolló la noción de una arquitectura de varios niveles y entre las que más se destacan se encuentra el modelo de tres capas.

La aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definido. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación. [5]

Sitio web.

Un Sitio Web es un conjunto de archivos electrónicos y páginas Web que son accesibles mediante el protocolo HTTP de Internet. Este conjunto de páginas tiene como objetivo el desarrollo de algún tema, prestar servicios o entregar contenidos asociados al entorno de la página. Un sitio está compuesto por una o más páginas, del mismo modo que en una revista, un artículo puede tener una o más páginas.

Aplicaciones Web.

Una aplicación Web es un sistema Web donde la entrada o acción del usuario afecta el estado del negocio. Estas se consideran una especialización y concreción de las aplicaciones cliente-servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP) son estándar, y no han de ser creados por el desarrollador, aunque del lado del servidor pueden estar varias capas (por ejemplo la capa de negocio y capa de datos).

La parte de cliente de las aplicaciones Web está formada por el código HTML que forma la página Web, con opción a código ejecutable mediante los lenguajes script de los navegadores (Java Script, VBScript, Perl Script) o mediante pequeños programas (applets) en Java. La parte de servidor está formada por un programa o script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente. [6]

Tecnologías del lado del cliente.

Tecnologías que brindan interactividad a los contenidos en las páginas Web que se ejecutan en el navegador del cliente.

Lenguaje de Marcado de Hipertexto.

El Lenguaje de Marcado de Hipertexto (*Hiper Text Markup Language HTML*), fue desarrollado con el objetivo de crear una interfaz gráfica de mayor calidad para visualizar documentos en el WWW, a través de los navegadores, siendo hoy en día la interfaz más extendida en la red.

Este lenguaje se basa en etiquetas o comandos encerrados entre signos " < > " que indican la inserción de textos, imágenes, enlaces, y otros en la página Web. [7]

Permite aglutinación y combinación de estos al gusto de los desarrolladores. Además, es aquí donde reside su ventaja con respecto a libros o revistas, el HTML proporciona la introducción de referencias a otras páginas por medio de los enlaces hipertexto pero se limita a describir la estructura y el contenido de un documento, y no el formato de la página y su apariencia. [8]

El lenguaje para describir páginas HTML, queda limitado a la hora de definir cualquier tipo de interactividad tales como acceder a una Base de Datos o animar gráficos y textos.

Otros de los problemas que han acompañado al HTML es la diversidad de navegadores presentes en el mercado, los cuales no son capaces de interpretar un mismo código de una manera unificada. Esto obliga al autor, una vez creada su página, a comprobar que ésta puede ser leída satisfactoriamente por todos los navegadores, o al menos, los más utilizados. Se considera, que a pesar de las ventajas que trajo el HTML, no logró una completa aceptación, debido a las limitaciones que tiene para la creación de páginas Web, fundamentalmente desde el punto de vista de que sólo permite contenidos estáticos. Es por esto, que posteriormente se introducen otras tecnologías que extienden sus potencialidades y permiten organizar, optimizar y automatizar el funcionamiento de las Web. Ejemplos: las CSS, *Java Script*, *VBScript* y otros.

Estos lenguajes capaces de recrear a partir de ciertos *script* un sinfín de páginas automatizadas son los protagonistas del concepto de páginas dinámicas. [9]

Dinámicas vs. HTML.

La plataforma WWW ha evolucionado progresivamente y, pasó de ser una aglomeración de documentos con información estática, a un ambiente donde se pueden implementar potentes aplicaciones, como cliente y servidor, accesibles desde un cliente Web o *browser* como el Netscape Navigator o el Microsoft Internet Explorer.

Cuando los contenidos de las páginas no necesitan modificarse continuamente sino que son perennes, el empleo de páginas dinámicas resulta a todas luces improductivo. Por el contrario, si el sitio es extenso y sus contenidos cambian rápidamente, es importante automatizar en la medida de lo posible todas las tareas de forma tal que se pueda gestionar su explotación de la manera más óptima.

El HTML Dinámico es un término colectivo para una combinación de nuevas etiquetas y opciones, hojas de estilo y programación HTML que permite crear páginas más animadas y que respondan mucho más a la interacción del usuario.

Permite que las páginas Web se vean y se comporten como aplicaciones del escritorio o producciones multimedia.

El mayor obstáculo al uso del HTML dinámico es que, como muchos usuarios aún están usando navegadores antiguos, los sitios Web deben crear dos versiones de cada página y ofrecer las apropiadas para la versión de navegador del usuario.

Cada elemento de una página (división o sección, encabezado, párrafo, fotografía, imagen, lista y demás) se ve como un "objeto" (Microsoft llama a esto el "Modelo de Objeto de HTML Dinámico"; Netscape lo llama el "Modelo de Objeto HTML"; y W3C lo llama el "Modelo de Objeto de Documento"). Por ejemplo, cada encabezado de una página puede tener un nombre y es posible darle atributos de estilo y color de texto, y llamarlo por su nombre en un pequeño programa o *script* incluido en la página.

Estos programas (entre los cuales se encuentran el *Java Script* y el *VBScript* son simplemente incluidos en el código HTML y al igual que este pueden ser directamente interpretados por el navegador [6]. A continuación se relata en qué consisten algunos de ellos.

Lenguajes de Script.

Han surgido para extender las capacidades del Lenguaje HTML. Son lenguajes de programación utilizados para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado. Esto significa que no sufre ninguna transformación previa a su ejecución. El navegador del cliente es el encargado de interpretar las instrucciones *script* y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. [10]

Permite crear efectos especiales en páginas Web, ayuda a la creación de contenidos dinámicos y elementos de la página que tengan movimiento a través del manejo de los distintos componentes de la página Web. Por otro lado permite ejecutar instrucciones como respuesta a las acciones del usuario para crear páginas interactivas gracias a ser orientado a eventos. [10]

Dos son los lenguajes de *script* que en la actualidad se encuentran mayormente estandarizados: el *VBScript* (derivado de Visual Basic) y el *JavaScript* (derivado de Java). El primero fue desarrollado por Microsoft e implementado en su navegador Internet Explorer y el segundo por Netscape en su navegador Navigator. [12]

ActiveX.

ActiveX es el nombre que Microsoft ha dado a un grupo de tecnologías y herramientas "estratégicas" orientadas a objetos. Su principal tecnología es el Modelo de Objeto Componente (*Component Object Model*, COM).

Un control *ActiveX* es un objeto (un fragmento modular de software) que provee una interfaz de programación configurable, de manera que realiza diversas tareas o cálculos, se comunica con otros programas, con otros módulos y con Internet, siendo fácilmente reutilizable por cualquier programa o entorno de programación que pueda contener controles *ActiveX*, como Internet Explorer.

Constituyen una extensión de los controles OLE (*Object Linking and Embedding*) optimizados para su descarga por Internet. [13]

ActiveX permite incrustar programas plenamente funcionales dentro de las páginas Web, volviéndolas más dinámicas e interactivas. Introduce serios problemas de seguridad, ya que una vez cargadas en el ordenador se ejecutan con control total sobre nuestros recursos, pudiendo perpetrar cualquier tipo de fechoría, desde destruir datos hasta inocular virus.

ActiveX es la respuesta de Microsoft a la tecnología Java de *Sun Microsystems*.

Un control *ActiveX* es aproximadamente el equivalente a un *Applet Java*. [14]

Applets – Java.

Un *applet* es un pequeño programa ejecutable escrito en lenguaje Java, que se coloca en un servidor Internet, junto con el resto de ficheros que componen un sitio Web para lograr múltiples efectos con texto, imágenes, sonidos, etc. Se transporta por la red, se instala automáticamente en la maquina cliente y requiere de un navegador para su ejecución.

Dado que Java es multiplataforma y los *applets* son soportados por ambos navegadores (Internet Explorer y Nestcape), su uso se ha popularizado bastante. Pueden ejecutar animaciones interactivas, cálculos inmediatos u otras tareas sencillas sin tener que enviar una solicitud del usuario al servidor. Se interpretan ejecutándose en la máquina local sin correr el riesgo de dañar los equipos. Esto ocurre porque ningún *applet* puede eliminar información o dañar un archivo local de ninguna manera. [15][16]

Hoja de Estilos en Cascada.

Hoja de Estilos en Cascada (*Cascading Style Sheets* CSS) es un mecanismo simple para adicionar estilo a los documentos HTML. Con CSS se puede especificar el tamaño, color y espacio del texto, así como el lugar donde estarán ubicados los textos e imágenes en la página.

Las clases son una nueva posibilidad que introdujo el HTML v 4.0, que permite definir estilos para cada una de las marcas del HTML e incluso hacer varios estilos para una determinada marca. En una clase se pueden definir tipos de letras, imágenes de fondo, así como otra buena cantidad de atributos provechosos. Hay que tener en cuenta que no todos los navegadores soportan todas las posibilidades de los CSS, por lo que es bueno asegurarse de que la página generada se vea lo más parecida posible en los distintos navegadores, aunque la mayor parte de los clientes esté utilizando los navegadores de Microsoft. [7]

Otra de las bondades de los CSS es que permiten de una forma fácil y centralizada, definir los estilos de las páginas y asegurar la consistencia en todas las páginas (o un grupo de ellas) en un documento o un sitio Web. Es por esto que se considera importante su utilización en la creación de las páginas Web.

Document Object Model (DOM).

El Modelo de Objetos de Documento o DOM, es la interfaz que permite acceder y manipular, mediante la programación, los contenidos de una página Web (o documento). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las propiedades de dichos objetos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiéndote crear contenido dinámico. [17]

El DOM proporciona también una interfaz para trabajar con eventos, permitiéndote capturar y responder a las acciones del usuario o del navegador. Siendo una especificación del W3C, uno de los objetivos importantes del Modelo de Objetos del Documento es proporcionar un interfaz estándar de programación que pueda utilizarse en una amplia variedad de entornos y aplicaciones. El DOM se ha diseñado para ser utilizado en cualquier lenguaje de programación. [17]

Tecnologías del lado del servidor.

Servidores Web.

Un servidor Web es un programa que implementa el protocolo HTTP (Hypertext Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Entre los servidores más utilizados en Internet se encuentran Apache e Internet Information Services (IIS). [19]

CGI.

Common Gateway Interface, es la más antigua de las conocidas y de hecho sentó las bases para las demás. Se basa en la ejecución de programas en el servidor Web. Esto ocurre cuando se hace una petición específica, obteniendo, en la mayoría de los casos un documento HTML generado en tiempo real, a partir de los parámetros definidos por el cliente. La tecnología CGI es muy útil para trabajar con bases de datos y trasladar la información almacenada en este medio hacia el formato HTML. Una de las ventajas principales que ofrece este tipo de aplicaciones es la independencia del hardware y del Sistema Operativo que utiliza el cliente al ser diseñadas para ejecutarse remotamente, y dependen de manera exclusiva de las capacidades del servidor WEB. Por tanto, siempre que los programas CGI devuelvan elementos HTML estándares, sus resultados podrán ser accesibles para la mayoría de los usuarios de WWW. Una desventaja es que necesitan crear un nuevo proceso en el servidor para atender cada solicitud del cliente, por lo que hacen un uso intensivo de los recursos del Servidor Web.

Una evolución inmediata de CGI sobre la plataforma Windows, para resolver este inconveniente lo constituyeron las API. [18]

ISAPI / NSAPI.

Son una variante de la anterior pero tienen la peculiaridad de que utilizan las *Aplication Program Interfaces* (APIs) de los servidores Web, ISAPI para *Internet Information Server* (IIS) y NSAPI para el servidor Web de Netscape. Las ISAPIs son bibliotecas de enlaces dinámicos *(DLL)* que corren en el mismo espacio de direcciones del servidor,

por lo que se obtienen mejorías en términos de velocidad y aprovechamiento óptimo de los recursos. El principal beneficio que poseen es que cuando se invocan la primera vez, se cargan en memoria y se quedan allí esperando a volver a ser invocadas además de abrir un hilo de ejecución por pedidos, esto posibilita que no se pierda tiempo adicional en la carga después de la primera vez y que aumente el desempeño del servidor.

Tanto ISAPI como CGI son para un servidor Web determinado aunque las primeras han desplazado a las segundas. [18]

Servidor Apache.

El servidor Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras. Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración. En la actualidad (2007), Apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios Web en el mundo y creciendo aún su cuota de mercado.

Apache además de constituir el servidor más utilizado, forma parte del proyecto de "código abierto", implicando esto que puede obtenerse libremente a través de Internet de forma gratuita la aplicación y su código fuente, lo que ha permitido el desarrollo de funciones del Apache por muchos programadores con carácter voluntario, dando lugar a rápidas mejoras en su funcionalidad y posibilitando la rápida subsanación de errores o fallos de seguridad.

Aunque Apache es un servidor inicialmente pensado para ser ejecutado en servidores Unix, se ha realizado un verdadero esfuerzo de migración, de modo que es un software capaz de funcionar en una gran cantidad de plataformas: aix, aux, beos, bs2000-osd, bsdi, cygwin, darwin, dgux, digitalunix, freebsd, hpux, irix, linux, macosx, macosxserver, netbsd, netware, openbsd, os2, os390, osf1, qnx, reliantunix, hapsody, sinix, solaris, sunos, unixware y win32. Pero sin dudas, Linux es su plataforma favorita y en la que más servidores de Internet están trabajando. [20].

Servidor Internet Information Services (IIS).

IIS es una serie de servicios para los ordenadores que funcionan con Windows.

Originalmente solo era parte del Windows. Luego fue distribuido en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS.

Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. [19]

Este servicio convierte a una computadora en un servidor de Internet o Intranet, es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web). Si se quiere para usar local es más recomendable utilizar el PWS (*Personal Web Service*). El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de *Active Server Pages* (ASP) y ASP.NET.

También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

La principal diferencia que muestra con respecto al anterior es que IIS está concebido para entornos Windows y Apache para entornos Unix, sin embargo vale la pena destacar el esfuerzo de Apache para correr sobre plataformas no nativas como Windows, mostrando valores de rendimiento muy favorables. Apache obtiene sus valores de rendimiento más elevados sobre entornos Unix. Otro punto importante a destacar es que el IIS presenta interfaces gráficas para la interacción con los usuarios, mientras que Apache carece de la misma. [19]

Scripts del lado del servidor.

Una peculiaridad de este grupo es que se apoya en las anteriores. De hecho, los Server-Side Script embebidos en el HTML utilizan a las tecnologías antes descritas como una interfaz o una capa entre el servidor Web y la maquinaria encargada de ejecutarlo. Debido a esto, no se puede pretender que estas tecnologías sean más rápidas que las 3 primeras. Ahora se ilustrarán las distintas tecnologías más usadas.

Active Server Pages.

Active Server Pages (ASP) es la tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft, asociadas también al *Internet Information Server*. Evidentemente su empleo se realiza sobre plataformas Windows. [21]

Una aplicación ASP es un conjunto de ficheros con extensión *.asp* interpretados por una ISAPI específica. Estos documentos son una mezcla de HTML con código en lenguaje *script*, generalmente *Visual Basic Script* o *Java Script*. Las etiquetas del HTML son devueltas con inmediatez al servidor WEB por parte de la ISAPI, mientras que el *script* es interpretado. Es posible que surja una confusión cuando se mencionan estos dos lenguajes, pues ambos son implementados por el Internet Explorer para ejecutar acciones dentro de la página Web, y debe quedar bien claro que se habla de código a ejecutarse en el servidor, no en el cliente Web. [22]

Una de las principales ventajas que presenta el trabajar con páginas dinámicas es el poder almacenar los contenidos en bases de datos. De esta forma, se puede organizarlos, actualizarlos y buscarlos de una manera mucho más simple.

ASP ofrece una forma muy eficaz de interaccionar con las bases de datos gracias al uso del componente ADO *(ActiveX Data Objects)* el cual permite acceder a dichas bases de una forma sencilla. ADO no es más que un grupo de objetos que, utilizados en conjunto permiten explotar de una forma muy versátil las bases de datos de la aplicación.

Por otra parte, los *scripts* ASP deben establecer un diálogo con la base de datos. Este diálogo se lleva a cabo a partir de un idioma universal: el SQL *(Structured Query Language)* el cual es común a todas las bases de datos. [23]

Professional Home Pages.

Professional Home Pages (PHP) podría ser considerado como el lenguaje análogo al ASP utilizado en plataformas Unix y Linux. A continuación se mencionan algunas de sus características.

Es un lenguaje de programación de estilo clásico con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como HTML.

Está más cercano a *JavaScript* o a *C*. Es una tecnología multiplataforma con un lenguaje propietario derivado del *Perl*.

A diferencia de *Java* o *JavaScript* que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo una base de datos. El programa en PHP es ejecutado en el servidor y el resultado enviado al navegador, normalmente una página HTML. [24]

El objetivo final es conseguir la integración de las paginas HTML con aplicaciones que corran en el servidor como procesos integrados en el mismo, y no como un proceso separado, como ocurría con los CGI.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte. Es independiente del navegador, sin embargo para que sus páginas funcionen, el servidor donde están alojadas debe soportar PHP.

PHP es uno de los lenguajes de lado servidor más extendidos en la Web debido sobre todo a la potencia y simplicidad que lo caracterizan. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para los CGI y las APIs, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera llana y sin complicaciones. [25] Tanto PHP como ASP resultan bastante útiles para la explotación de bases de datos y su aprendizaje no resulta muy engorroso. Cualquiera de ellos resultaría la opción ideal a la hora de hacer evolucionar un sitio Web realizado en HTML.

Se podría efectuar una comparación entre el lenguaje PHP y la tecnología ASP a la hora de generar páginas dinámicas estableciéndose algunas diferencias:

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones.
- ASP, siendo una tecnología Microsoft, está orientado hacia sistemas Windows, especialmente NT.
- Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente uno y otros procedimientos. ASP realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados (o programados) por el servidor a determinadas empresas especializadas.
- PHP presenta una filosofía totalmente diferente y, con un espíritu más generoso, es progresivamente construido por colaboradores desinteresados que implementan nuevas funciones en nuevas versiones del lenguaje.

Elementos para tener en cuenta en un lenguaje de scripts.

Las cuatro grandes características: Velocidad, estabilidad, seguridad y simplicidad. [26]

- Velocidad: No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado.
- Estabilidad: La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de fallas, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para estas fallas sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- Seguridad: El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo.ini.
- Simplicidad: Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.
- Otra característica a tener en cuenta seria la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil. Esto le permite al PHP ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos, XML y otras.

Gestión de Datos.

La mayoría de los sistemas informáticos de gestión operan el almacenamiento de la información externa de la capa de datos a través de Sistemas Gestores de Base de Datos (SGBD).

Un Sistema de Gestión de Base de Datos (SGBD) es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. [28]

En la actualidad existen un sinnúmero de SGBD, muchos de ellos muy potentes, entre los que se encuentran Microsoft SQL Server, Oracle, Postgree SQL, MySQL, entre otros.

Lenguaje de consultas estructurado.

El *Structured Query Language (SQL)* no es más que un lenguaje estándar de comunicación con bases de datos. SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. En muchos casos, el esquema básico de operación es una serie de *scripts* que rigen el comportamiento de una base de datos. [29]

El hecho de que sea estándar no quiere decir que sea idéntico para cada base de datos. En efecto, determinadas bases de datos implementan funciones específicas que no tienen necesariamente que funcionar en otras.

Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una parte, presenta una potencia y versatilidad notables que contrasta, por otra, con su accesibilidad de aprendizaje.

Microsoft SQL Server.

Microsoft SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor. Es uno de los sistemas más rápidos disponibles. Soporta disparadores: *script* que se ejecutan al ocurrir determinados eventos y procedimientos almacenados: serie de sentencias SQL, que se almacenan dentro de la BD y pueden ser usados para consultar o actualizar datos. Permite dar contenido por XML (Lenguaje de Marcado Extendido). Además tiene mecanismo de réplica y publicación, las herramientas de administración son muy completas, soporta autenticación integrada con NT o autenticación SQL. [30]

MYSQL.

MYSQL es un proyecto *open source*. Es un gestor de bases de datos bastante rápido. Sólo es más rápido en las consultas simples, creando y eliminando índices principalmente. La licencia es completamente libre. Está soportado por la gran mayoría de los SO como Solarix, Linux, Windows, Mac entre otros. Para su conexión se utiliza *Open Database Connectivity* (ODBC), programa de interfaz de aplicaciones (API) para

acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello SQL (lenguaje de consulta estructurado). Un punto débil de este gestor de bases de datos es que no implementa todo el estándar ANSI SQL 92. No soporta disparadores ni procedimientos almacenados. [32]

Inicialmente, MYSQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan. Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar: [31]

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Dentro de las Ventajas que MYSQL ofrece se encuentran:

- Alto rendimiento.
- Elevada velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- Excelentes utilidades de administración (backup, recuperación de errores, etc.).
- Aunque se bloquee, no suele perder información ni corromper los datos.
- Excelente integración con PHP.
- No tiene límites en el tamaño de los registros.
- Significativo control de acceso, es decir, qué usuarios tienen acceso a que tablas y con qué permisos.

- MYSQL se comporta mejor que otros motores de Base de Datos como SQL Server y Oracle a la hora de modificar ó añadir campos a una tabla.
- Es multi-hilo, o sea, maneja muchas conexiones al mismo tiempo, donde cada una tiene su propio hilo, de modo que ningún hilo tiene que esperar por otro, a menos que uno esté modificando una tabla que otro quiera acceder.

Seguridad

MD5

Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta.

Para comprobar la integridad de un archivo descargado de Internet se puede utilizar una herramienta MD5 para comparar la suma MD5 de dicho archivo con un archivo MD5SUM con el resumen MD5 del primer archivo. En los sistemas UNIX, el comando de *md5sum* es un ejemplo de tal herramienta. Además, también está implementado en el lenguaje de *scripting* PHP como MD5("") entre otros.

En sistemas UNIX y GNU/Linux se utiliza el algoritmo MD5 para cifrar las claves de los usuarios. En el disco se guarda el resultado del MD5 de la clave que se introduce al dar de alta un usuario, y cuando éste quiere entrar en el sistema se compara la entrada con la que hay guardada en el disco duro, si coinciden, es la misma clave y el usuario será autenticado. He ahí el problema de encontrar y generar colisiones de *hash* a voluntad.

El MD5 también se puede usar para comprobar que los correos electrónicos no han sido alterados usando claves públicas y privadas.

Metodología.

Lenguaje Unificado de Modelación (UML).

UML (por sus siglas en inglés, *Unified Modelling Language*) es el lenguaje de modelado de sistemas de software más conocido en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. UML cuenta con varios tipos de modelos, los cuales muestran diferentes aspectos de las entidades representadas.

Rational Unified Process (RUP).

El Proceso Unificado de Desarrollo o RUP (Rational Unified Process), es un proceso de desarrollo de software que utilizando UML, constituye una metodología estándar muy utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Tiene las siguientes características: [33]

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué,
- cuándo y cómo)
- Pretende implementar las mejores prácticas en ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

RUP es un producto de Rational. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (productos tangibles del proceso) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al concluir cada uno, y cada ciclo se divide en las siguientes fases:

- Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- Transición: se implementa el producto en el cliente y se entrena a los usuarios.
- Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados.

Herramientas de desarrollo.

Rational Rose 2003.

Para apoyar el trabajo con esta metodología ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose desde el año 2000. Esta herramienta integra todos los elementos que propone la metodología para cubrir el ciclo de vida de un proyecto. Esta herramienta CASE propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. [34]

Rational Rose proporciona también mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño. [34]

ADOBE Dreamweaver CS3.

Con esa herramienta se podrá desarrollar cualquier sitio Web personal con características de sitio profesional y utilizar casi todos los recursos de la Web, así como realizar aplicaciones que se ejecuten en servidor y vinculaciones dinámicas de datos; además de contar con un soporte para aplicaciones PHP y utilización de bases MySQL. También cuenta con un amplio soporte para la creación y utilización de CSS (*Cascading Style Sheets*) para lograr un diseño fácil y óptimo.

Zend Studio 5.5.

Zend Studio 5 se ha diseñado para una amplia gama de programadores y existen dos ediciones:

Standard y Professional. Zend Studio 5, concebido con el fin de crear aplicaciones altamente fiables, proporciona una facilidad de uso inigualable, escalabilidad, fiabilidad, y la extensión que los programadores profesionales y de empresas requieren para desarrollar, distribuir, depurar y administrar aplicaciones PHP críticas de negocios.

- Integra Java fácilmente en su código utilizando las características del completado de Código y define/especifica Jars adicionales o carpetas de clase que pueden utilizarse para el completado de códigos.
- Integración del uso y completado de código personalizado de Zend Framework y vista de la lista de las funciones del framework desde la Visualización de Funciones PHP.
- Visualización los eventos de Zend Plataforma en una ventana de lista de eventos personalizada y dedicada. Clic en cada evento para ver el detalle completo del evento en la ventana del navegador.
- Aumentar la productividad con: Soporte PHP 5 completo, Analizador de Código, carpeta de Código, completado de Código, coloreado de Sintaxis, Administrador de Proyecto, Editor de Código, Depurador de gráficos y asistentes.
- Documentación del código de forma más sencilla, aplicaciones, y proyectos con PHP Documentor, la herramienta de documentación estándar para PHP.
- Simplificar el despliegue con la integración FTP y SFTP de forma tal que permita a los programadores en forma segura subir y descargar archivos de proyectos de modo transparente hacia y desde servidores remotos.

1.6 CONCLUSIONES

Después de realizar un estudio de los procesos que tienen lugar en la oficina de asuntos estudiantiles y de los sistemas vinculados a estos; se evidencia la necesidad de crear una aplicación que facilite la gestión de los estudiantes guyaneses. Para ello, se propone un sistema basado en tecnología Web y base de datos.

Para el desarrollo del sistema propuesto se realizó un análisis de las tecnologías actuales a considerar, pudiendo concluirse que: el lenguaje que será utilizado para la implementación de la solución propuesta es PHP y MYSQL como servidor de Base de Datos, teniendo en cuenta que ambas tecnologías se han convertido es estándares para el desarrollo de aplicaciones web. PHP se ha convertido en un poderoso lenguaje de programación que se ejecuta del lado del servidor y que ha sido creado específicamente para el desarrollo de aplicaciones de este tipo, y MYSQL como un eficaz motor de bases de datos relacionales que proporciona el depósito perfecto para aplicaciones que requieran acceso rápido y confiable a la información.

Se utilizará Apache como servidor web, como resultado de ser el servidor más utilizado y eficiente en la red.

Se realizará el análisis, diseño e implementación del sistema utilizando la metodología RUP, basada en el lenguaje unificado de modelación UML.

CAPITULO – 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 INTRODUCCIÓN.

Modelar e identificar el flujo de los procesos que serán objeto de automatización de un sistema informático, es un elemento clave para lograr un desarrollo exitoso del producto y una buena comunicación entre los desarrolladores, los clientes y el usuario final. A este flujo de trabajo se le denomina: Modelación del Negocio.

El modelado del negocio es una técnica que permite comprender los procesos de negocio de la organización y se desarrolla en dos pasos: [34]

1. Confección de un modelo de casos de uso del negocio que identifique los actores y casos de uso del negocio que utilicen los actores.
2. Desarrollo de un modelo de objetos del negocio compuesto por trabajadores y entidades del negocio que juntos realizan los casos de uso del negocio.

El objetivo fundamental de este capítulo es modelar los procesos de negocio que tienen lugar en la Oficina de Asuntos Estudiantiles (SAO) de la Embajada de Guyana, a través del uso de los artefactos que propone la metodología RUP y el lenguaje UML para este flujo de trabajo.

2.2 DESCRIPCIÓN DEL MODELO DEL NEGOCIO.

El primer paso del modelado del negocio consiste en capturar y definir los procesos de negocio de la organización bajo estudio, tarea crucial que define los límites del proceso de modelado posterior. [35]

Un proceso de negocio se entiende como un grupo de tareas relacionadas de manera lógica que se llevan a cabo en determinada secuencia, y producen o manipulan una colección de datos empleando recursos de la organización para dar resultados que apoyan sus objetivos. [35]

A partir del planteamiento anterior se realizó un estudio de la organización en cuestión y los procesos identificados para ser modelado fueron:

- Gestión de la información de los estudiantes

- Gestión de los permisos de los estudiantes
- Recogida de las notas de los estudiantes
- Generación de reportes

Gestión de la información de los estudiantes.

Después que los estudiantes se encuentran en la preparatoria, estos llenan una planilla con todos los datos que solicita la embajada, entre los cuales se cuenta un vínculo en su país que se pueda contactar en caso de emergencia.

El responsable de los asuntos estudiantiles de los guyaneses en Cuba recorre los centros donde realizan su preparatoria y recoge las planillas.

El responsable de los asuntos estudiantiles de los guyaneses en Cuba introduce todos sus datos en un documento de Excel.

Gestión del permiso de salida de los estudiantes.

El estudiante solicita permiso a la SAO.

En caso de que la solicitud de salida se deba a algún problema grave y repentino:

- La embajada localiza al contacto de emergencia del estudiante en Guyana.
- De tratarse de un hecho grave (fallecimientos, accidentes, etc.) la embajada autoriza automáticamente al estudiante y paga su pasaje.

En caso de solicitud de salida por vacaciones:

- La embajada gestiona con el centro estudiantil si el estudiante puede salir o no.
- Si es autorizado, el estudiante paga su pasaje y sale del país.
- Si no es autorizado y el estudiante se va, causa baja automáticamente del centro.

Recogida de las notas de los estudiantes.

El responsable de los asuntos estudiantiles recorre los centros donde los guyaneses cursan estudios y recoge los resultados académicos de cada uno de ellos al concluir cada semestre.

Generación de reportes.

La Oficina de Asuntos Estudiantiles (SAO por sus siglas en ingles) crea reportes de los datos de los estudiantes de forma manual.

2.3 REGLAS DEL NEGOCIO A CONSIDERAR.

Generales.

- Si un estudiante viola por cualquiera razón el contrato que firmó para estudiar en Cuba, automáticamente pierde la beca. El contrato se viola si se desobedecen las reglas del centro de educación o las leyes del país, entre otras acciones.
- Si una estudiante presenta un embarazo, causará automáticamente baja de la carrera y de la universidad; y el gobierno pagara su pasaje de regreso a Guyana.

Particulares.

Caso de Uso Solicitar Permiso.

- Las solicitudes de los estudiantes para cambiar su carrera, repetir un año, cambiar de universidad, salir de la provincia y país, o pedir licencia sólo pueden ser realizadas mediante una carta de solicitud de permiso.

- Los estudiantes sólo deben recibir la respuesta a sus solicitudes mediante una carta de permiso emitida por el encargado de asuntos estudiantiles, el cual a su vez será el único documento que tendrá validez para autorizar a cualquier nivel la puesta en práctica de la solicitud hecha por el estudiante.

- Si el ministerio no le otorga permiso al estudiante para viajar, el encargado de asuntos estudiantiles no podrá emitir la carta de permiso de salida.

- Si un estudiante solicita repetir un año de la carrera por deficiencia académica se toma en cuenta para la autorización de su solicitud el criterio del Encargado de Asuntos Estudiantil

Caso de uso Colectar y Archivar datos Personales de los Estudiantes.

- El encargado de asuntos estudiantiles sólo podrá archivar los datos personales de un estudiante si este ha revisado y aprobado los datos a archivarse.

2.4 – MODELO DE CASOS DE USO DEL NEGOCIO

2.4.1 Actores del negocio.

Actor	Descripción
Estudiante	El estudiante es el que cursa estudios en Cuba, por lo que es el que brinda sus datos personales para ser almacenados, y solicita permisos.
Oficial	El oficial es una persona afuera del negocio que solicita reportes de diferentes criterios de la oficina de asuntos estudiantiles de la embajada.
Organismo de Educación	El Organismo de Educación es una entidad externa que se encarga de gestionar la información referente a los estudiantes que estudian carreras en la República de Cuba que le son afines.

Tabla 1: Actores del Negocio. Justificación

2.4.2 Trabajadores del Negocio.

Trabajador	Descripción
Encargado de asuntos estudiantiles de la Embajada de Guyana.	Se encarga de responder las solicitudes de las estudiantes. Revisa la trayectoria del estudiante para determina si el estudiante satisface el criterio de repetir un año. Compra el boleto de regreso para una estudiante que tiene una situación de emergencia.
Encargado de asuntos estudiantiles del Ministerio de Servicio Público.	Verifica si un estudiante puede salir. Determina si un estudiante puede hacer un cambio, o pedir licencia. Establece las condiciones para un estudiante repite un año.

Tabla 2: Trabajadores del Negocio. Justificación

2.4.3 Diagramas de los Casos de Uso del Negocio.

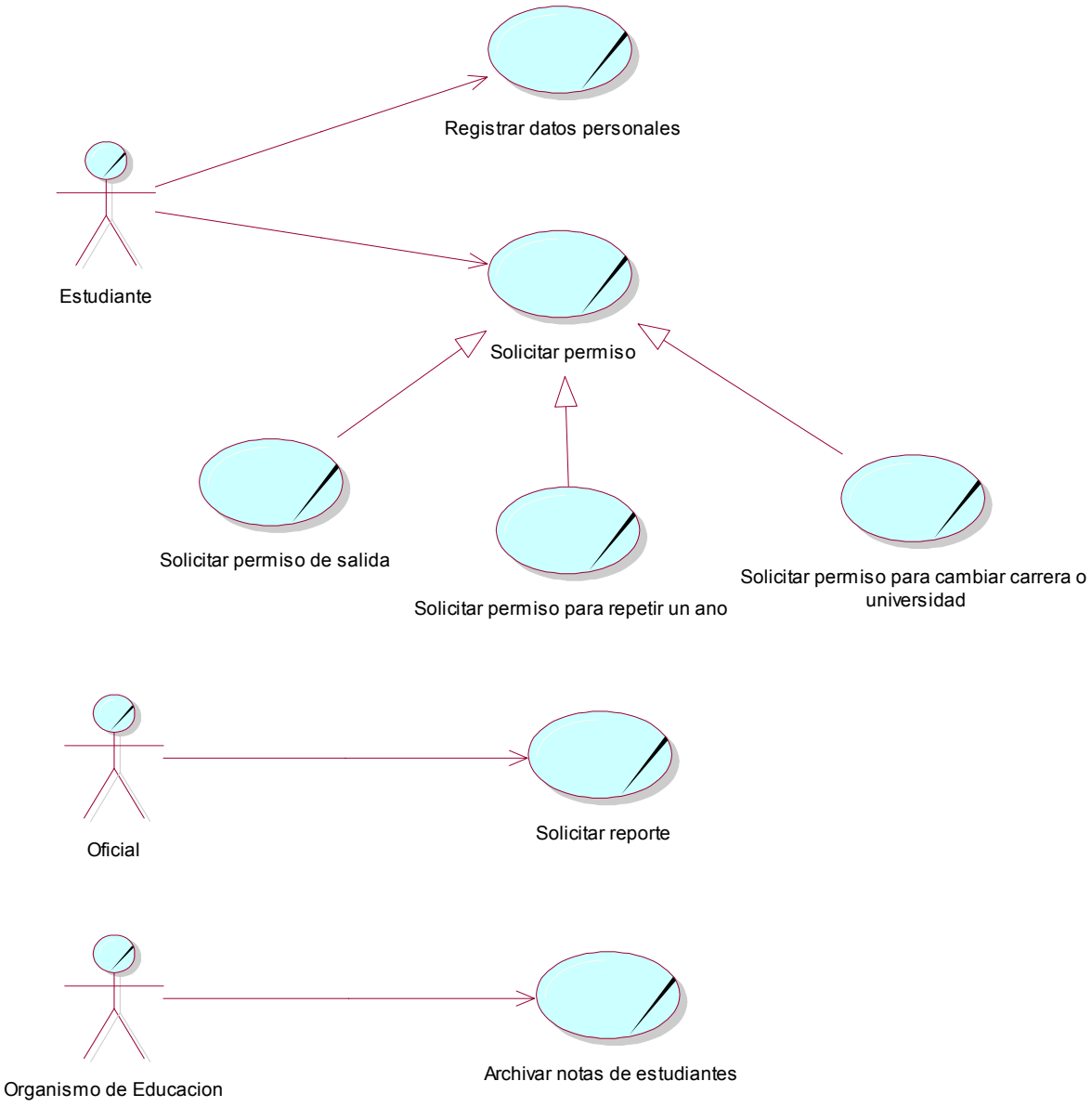


Figura 1: Diagramas de los Casos de Uso del Negocio.

2.4.4 Descripción de los Casos de Uso.

Caso de uso del negocio	Registrar Datos Personales
Actores	Estudiante Guyanés. (inicia)
Propósito	Registrar Datos Personales
Casos de uso asociados	

Resumen

El Caso de Uso se inicia cuando el estudiante llega a la República de Cuba para iniciar sus estudios. El encargado de asuntos estudiantiles solicita que el estudiante llene sus datos. El caso de uso concluye cuando se han archivado los datos personales de los estudiantes.

Acción del Actor		Respuesta del Proceso de Negocio	
1	El estudiante guyanés arriba a la institución de preparatoria y se presenta en la oficina de relaciones internacionales.	2	El Encargado de Asuntos Estudiantiles solicita que el estudiante llene sus datos personales y le entrega a este una planilla vacía.
3	El estudiante llena sus datos personales y entrega le planilla llena al encargado de asuntos estudiantiles.	4	El encargado de asuntos estudiantiles revisa los datos llenados por el estudiante. Si están

			correctos se ejecuta el paso #5, sino se ejecuta la Sección *Corregir Datos.*
		5	El encargado de asuntos estudiantiles comprueba que todos los datos están correctos.
		6	El encargado de asuntos estudiantiles recoge la planilla.
Otras secciones		*Sección Corregir Datos*	
Acciones del Actor		Respuesta del Proceso de Negocio	
		1	El encargado de asuntos estudiantiles detecta un error en la planilla entregada por el estudiante.
3	El estudiante rectifica la planilla de sus datos personales y la entrega al encargado de asuntos estudiantiles.	2	El encargado de asuntos estudiantiles entrega la planilla al estudiante para que este la rectifique.
		4	El encargado de asuntos estudiantiles revisa la planilla entregada por el estudiante. Si esta está correcta se archiva la planilla, sino se ejecuta el paso Sección *Corregir Datos*.
Mejoras propuestas			

Tabla 3: Descripción del Caso de Uso “Registrar Datos Personales”.

Caso de uso del negocio	Solicitar Permiso
Actores	Estudiante
Propósito	Solicitar a la embajada un permiso de cualquier tipo.
Casos de uso asociados	Solicitar permiso de salida, Solicitar Permiso de Repetir un año, Solicitar Permiso para Hacer un Cambio o Pedir Licencia

Resumen

El Caso de Uso se inicia cuando el estudiante solicita a la embajada un permiso de cualquier tipo. La embajada gestiona el permiso solicitado por el estudiante. El caso de uso concluye cuando los se le informa al estudiante la respuesta a su solicitud.

Acción del actor		Respuesta del proceso de negocio	
1	El estudiante solicita a la embajada un permiso mediante una carta de solicitud.	2	La embajada recibe la carta de solicitud del estudiante.
		3	El encargado de asuntos estudiantiles gestiona el tipo de permiso solicitado por el estudiante.
		4	El encargado de asuntos estudiantiles informa al estudiante la respuesta a su solicitud.
5	El estudiante recibe la respuesta a su solicitud.		
Otras secciones			

Mejoras propuestas	

Tabla 4: Descripción del Caso de Uso “Solicitar Permiso”.

Caso de Uso del Negocio		Solicitar Permiso de Salida	
Actores		Estudiante (Inicia)	
Propósito		Obtener un permiso de salida solicitado por un estudiante	
Caso de Uso Asociado			
Resumen: El caso de uso se inicia cuando el estudiante solicita a la embajada un permiso y solicita al oficial de asuntos estudiantiles (SAO) y determina si un estudiante tiene una situación de emergencia. Un oficial del Ministerio de Servicio Público (PSM) verifica si el estudiante en realidad tiene una emergencia. En caso que no tenga una emergencia y quiera salir para otra razón, como vacaciones, el PSM verifica si el estudiante puede salir. El caso de uso concluye cuando el estudiante recibe la información de si ha sido autorizado o no.			
Acciones de Actor		Respuesta del proceso de negocio	
1	El estudiante solicita permiso de salida.		
		2	El SAO evalúa la solicitud y determina que es una situación de emergencia.
		3	El SAO solicita al PSM que verifique si en realidad el estudiante tiene una emergencia.
		4	El PSM notifica al encargado el resultado de la verificación realizada.
		5	El SAO recibe el resultado de la verificación que corrobora que el estudiante en realidad presenta una

			situación de emergencia. (Ver Flujo #1)
		6	El SAO estudiantiles compra el pasaje de salida e informa al estudiante que puede salir del país.
		7	El SAO emite una carta de permiso de salida del país.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta a su solicitud.		
Flujo Alterno a la Sección Permiso de Salida			
Flujo #1			
Acciones del Actor		Respuesta del proceso de negocio	
		1	El SAO estudiantiles recibe el resultado de la verificación que informa que el estudiante no presenta una situación de emergencia.
		2	El SAO pregunta al PSM si el estudiante puede o no salir del país.
		3	El PSM verifica si el estudiante puede o no salir del país.
		4	El PSM notifica al SAO el resultado de la verificación.
		5	El SAO recibe el resultado de la verificación que corrobra que el estudiante puede salir del país.

			(Ver Flujo #2)
		7	El SAO emite una carta de permiso de salida del país.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta a su solicitud.		
Flujo # 2			
Acciones del Actor		Respuesta del proceso de negocio	
		1	El SAO recibe el resultado de la verificación que informa que el estudiante no puede salir del país.
		2	El SAO rechaza la solicitud del estudiante.
		3	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
4	El estudiante recibe la respuesta a su solicitud.		
		7	El SAO emite una carta de permiso de salida del país.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta de respuesta a su solicitud.		
Mejoras Propuestas			

Tabla 5: Descripción del Caso de Uso Solicitar Permiso de Salida.

Caso de Uso del Negocio	Solicitar Permiso de Repetir un año
Actores	Estudiante (Inicia)
Propósito	Obtener un permiso solicitado por un estudiante
Caso de Uso Asociado	

Resumen:

El caso de uso se inicia cuando el estudiante solicita a la embajada un permiso y solicita al oficial de asuntos estudiantiles (SAO) y revisa la trayectoria de una estudiante para determinar si un estudiante puede repetir un año. El caso de uso concluye cuando el estudiante recibe la información de si ha sido autorizado o no.

Acciones de Actor		Respuesta del proceso de negocio	
1	El estudiante solicita permiso de repetir un año.		
		2	El SAO revisa la trayectoria del estudiante.
		3	La trayectoria del estudiante es satisfactoria. (Ver Flujo Alterno)
		4	El SAO emite una carta de permiso Repetir un año.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta a su solicitud.		

Flujo Alterno de la Sección Permiso para Repetir un año.

Acciones del Actor		Respuesta del proceso del negocio	
		1	La trayectoria del estudiante no es satisfactoria.
		2	El SOA rechaza la solicitud estudiante.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta de respuesta a su solicitud.		
Mejoras Propuestas			

Tabla 6: Descripción del Caso de Uso Solicitar Permiso de Repetir un año.

Caso de Uso del Negocio	Solicitar Permiso de hacer un cambio o pedir licencia
Actores	Estudiante (Inicia)
Propósito	Obtener un permiso solicitado por un estudiante
Caso de Uso Asociado	

Resumen:

El caso de uso se inicia cuando el estudiante solicita a la embajada un permiso y solicita al oficial de asuntos estudiantiles (SAO) y pregunta si un estudiante puede hacer un cambio o pedir licencia. Un oficial del Ministerio de Servicio Público (PSM) verifica si el estudiante en puede hacer un cambio o pedir licencia. El caso de uso concluye cuando el estudiante recibe la información de si ha sido autorizado o no.

Acciones de Actor		Respuesta del proceso de negocio	
1	El estudiante solicita permiso de salida.		

		2	El SAO pregunta al PSM si el estudiante está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia.
		3	El PSM verifica si el estudiante está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia
		4	El PSM notifica al SAO el resultado de la verificación
		5	El SAO recibe el resultado de la verificación que corrobra que el estudiante está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia. (Ver Flujo #4)
		6	El SAO emite una carta de permiso de que el estudiante puede está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta de respuesta a su solicitud.		
Flujo Alterno de la Sección Permiso de que el estudiante puede está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia.			
Acciones del Actor		Respuesta del negocio	
		1	El estudiante no está autorizado a cambiar la carrera, cambiar la universidad o pedir licencia.

		2	El SAO rechaza la solicitud estudiante.
		8	El SAO informa al estudiante mediante una carta la respuesta a su solicitud.
9	El estudiante recibe la carta de respuesta a su solicitud.		
Mejoras Propuestas			

Tabla 7: Descripción del Caso de Uso Solicitar Permiso para Hacer un Cambio o Pedir Licencia.

Caso de uso del negocio		Colectar y archivar Notas	
Actores		Organismo de Educación	
Propósito		Archivar las notas de todos los estudiantes correspondientes a un período de clases.	
Casos de uso asociados			
Resumen: El Caso de Uso se inicia cuando el organismo de educación informa a la embajada que finalizó un curso escolar y envía las notas de todos sus estudiantes. El encargado de asuntos estudiantiles archiva las notas. El caso de uso concluye cuando se han archivado las notas de los estudiantes.			
Acción del actor		Respuesta del proceso de negocio	
1	El Organismo de Educación informa a la embajada que finalizó el curso.	2	El encargado de asuntos estudiantiles recibe la información de que ya se termino el curso escolar.

4	El Organismo de Educación conforma el listado con las notas de sus estudiantes.	3	El encargado de asuntos estudiantiles solicita al Organismo de Educación las notas de sus estudiantes.
5	El Organismo de Educación envía el listado a la embajada.	6	El encargado de asuntos estudiantiles recibe el listado con las notas de los estudiantes pertenecientes al Organismo de Educación.
		7	El encargado de asuntos estudiantiles archiva las notas.
Otras secciones			
Mejoras propuestas			

Tabla 9: Descripción del Caso de Uso Solicitar reporte

Caso de uso del negocio	Solicitar Reporte
Actores	Oficial
Propósito	Generar un reporte según parámetros determinados.
Casos de uso asociados	
Resumen El caso de uso se inicia cuando un oficial solicita al Encargado de Asuntos Estudiantiles (SAO) un reporte según parámetros determinados. El Encargado crea el reporte y el caso de uso concluye cuando el Ministerio de Servicio Público recibe el reporte pedido.	

Acción del actor		Respuesta del proceso de negocio	
1	El oficial solicita un reporte según parámetros determinados.		
		2	El SAO crea el reporte.
3	El oficial recibe el reporte		
Mejoras propuestas:			

Tabla 8: Descripción del Caso de Colectar y archivar Notas.

2.5 DIAGRAMAS DE ACTIVIDADES

Diagrama de Actividades de Registrar datos personales Ver Anexo 1.1.

Diagrama de actividades del Caso de Uso Solicitar Permiso de Salida Ver Anexo 1.2.

Diagrama de actividades del Caso de Uso Solicitar Permiso de repetir un año Ver Anexo 1.3.

Diagrama de Actividades de Uso Solicitar Permiso para Hacer un Cambio o Pedir Licencia Ver Anexo 1.4.

Diagrama de Actividades de Archivar notas de los estudiantes Ver Anexo 1.5.

Diagrama de Actividades de Uso Solicitar Reporte Ver Anexo 1.6.

2.6 DIAGRAMA DE OBJETOS DEL NEGOCIO

Un modelo de objetos del negocio es un modelo interno de un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo

Una entidad del negocio representa algo, que los trabajadores toman, inspeccionan, manipulan, producen o utilizan en un caso de uso del negocio.

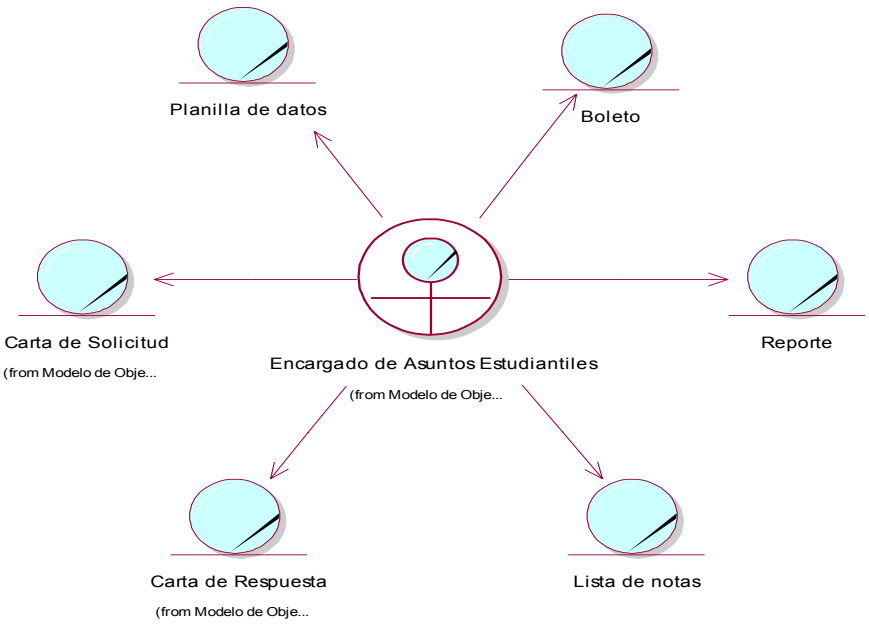


Figura 2: Diagrama de los objetos del Negocio.

2.7 REQUERIMIENTOS FUNCIONALES

2.7.1 – Introducción.

Una vez realizado el estudio del negocio e identificado el flujo de procesos que se desarrolla dentro del mismo, atendiendo a un análisis crítico de la situación problémica y proponiendo las mejoras correspondientes; los desarrolladores pasan a un nuevo flujo de trabajo conocido como requerimientos.

Los requerimientos no son más que las necesidades de los clientes y los usuarios finales expresadas a través de políticas que deben ser respetadas seriamente al llevar a cabo el análisis, diseño e implementación del sistema informático.

Se describen aquellas herramientas que son usadas en la modelación de los casos de uso del sistema, tales como los diagramas de dichos casos de uso, la descripción de los actores del sistema, la descripción de los casos de uso del sistema, la vista de arquitectura de los casos de uso y la especificación de algunos de ellos, para la realización de tales herramientas se tuvieron en cuenta los requerimientos que debe

cumplir el software para facilitar su fácil manejo y la interacción con él de la forma más eficiente posible.

2.7.2 – DESCRIPCIÓN DEL SISTEMA PROPUESTO.

2.7.2 – Concepción general del sistema.

El sistema que se propone consiste en la creación de un sistema para facilitar la gestión de datos de estudiantes, información de las carreras que estudian, las instituciones donde estudian y otros datos pertinentes. El sistema va a tener usuarios de diferentes niveles de acceso a los datos. El sistema puede crear reportes de diferentes criterios depende del criterio de un administrador. También el sistema va a brindar un servicio de mensajería por usuarios dentro el sistema.

Con su creación y el consiguiente cumplimiento de sus objetivos se logrará tener al alcance de los estudiantes y del encargado de asuntos estudiantiles una herramienta muy útil para la manipulación de datos, así para todo aquel que lo precise.

La aplicación se desarrollara en un ambiente amigable con el cliente, cuya implementación estará sustentada en los lenguajes de programación PHP, Javascript y HTML asociada a una Base de Datos de tipo mySQL. Los procesos que serán automatizados y que constituyen los casos de uso del sistema son todos los que tienen que ver con los siguientes aspectos:

1. Manipular los datos generales del estudiante
2. Manipular los datos relevantes del estudiante, como sus notas, los permisos solicitados y otra información pertinente al curso de estudios en Cuba.
3. Crear reportes pertinentes a los estudiantes

2.7.3 – Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo

Los requisitos funcionales del sistema son los siguientes:

1. Realizar autenticación de los usuarios al sistema.
2. Permitir que los usuarios del sistema cambien su contraseña
3. Insertar usuario
4. Modificar usuario
5. Eliminar usuario
6. Registrar usuario
7. Insertar estudiante
8. Modificar estudiante
9. Eliminar estudiante
10. Insertar nota del estudiante
11. Modificar nota del estudiante
12. Eliminar nota del estudiante
13. Insertar permiso del estudiante
14. Modificar permiso del estudiante
15. Eliminar permiso del estudiante
16. Insertar campo de estudio
17. Modificar campo de estudio
18. Eliminar campo de estudio
19. Insertar institución
20. Modificar institución
21. Eliminar institución
22. Insertar carrera
23. Modificar carrera
24. Eliminar carrera
25. Insertar asignatura
26. Modificar asignatura
27. Eliminar asignatura
28. Insertar provincia
29. Modificar provincia
30. Eliminar provincia

31. Insertar noticia
32. Modificar noticia
33. Eliminar noticia
34. Enviar mensaje
35. Leer mensaje
36. Eliminar mensaje
37. Insertar imagen
38. Eliminar imagen
39. Búsqueda de estudiante
40. Visualizar información particular de estudiante
41. Visualizar notas de estudiante
42. Visualizar permisos de estudiante
43. Visualizar campos de estudios
44. Visualizar institución
45. Visualizar carrera
46. Visualizar asignatura
47. Visualizar noticias
48. Visualizar imágenes
49. Crear reportes según varios criterios
50. Crear reportes estadísticos
51. Imprimir reportes

2.7.4– Requerimientos no funcionales.

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, etc.

Para el sistema propuesto se han definido los siguientes requisitos no funcionales:

Apariencia o interfaz externa.

El software brindará una interfaz sencilla que facilite la interacción del usuario con el mismo. La interfaz del sistema se ajustará a los estándares establecidos para el

desarrollo de un buen diseño. Tendrá consistencia con el mundo real de manera que la mayoría de los conceptos manejados son conocidos y les resultan familiares a los usuarios, lo que hace relativamente fácil su uso y aprendizaje.

Requisitos de Usabilidad.

Los usuarios del sistema quedarán definidos por algún visitante que quiera saber alguna información del curso de estudios en Cuba, un oficial, el encargado de los asuntos estudiantiles, y un estudiante.

No obstante a lo anterior, será el encargado de los asuntos estudiantiles quien use con mayor frecuencia el producto.

El encargado de los asuntos estudiantiles será la máxima responsable del control de los datos. El sistema, al automatizar las tareas fundamentales de la gestión de la tecnología, posibilitará un aumento del rendimiento de la actividad de la SAO. Es por ello que el sistema tendrá un buen nivel de uso una vez instalado.

El sistema sólo podrá ser utilizado por personas que posean los conocimientos básicos correspondientes a los asuntos de estudiantes guyaneses. Tendrá una documentación básica que comprenda los aspectos generales a tener en cuenta para llevar a cabo esta tarea, aunque la sencillez de su interfaz permitirá que la aplicación pueda ser usada por todas aquellas personas que no tengan experiencia en el uso de la computadora, luego de un ligero entrenamiento sobre el funcionamiento de los principales elementos de la interfaz.

Requisitos de Rendimiento.

Aunque no se requiere una velocidad de respuesta comparada con los sistemas de tiempo real, se debe garantizar la rapidez de respuesta del sistema ante las solicitudes de los usuarios.

Como la aplicación está concebida para 3 capas, debe garantizarse que los tiempos de respuestas sean generalmente rápidos al igual que la velocidad de procesamiento de la información. Se seguirá una lógica de negocios en la comunicación entre el cliente y el

servidor web de modo tal que cada cual procesará lo que le corresponde, puesto que es posible mencionar que existen funciones que son más óptimas ejecutarlas en el cliente, otras por cuestiones de seguridad, o de acceso a los datos, deben realizarse en el servidor. Se realizará una parte del proceso en el cliente y en dependencia de esto se concluirá en el servidor, lo que facilitará un tiempo de respuesta más rápido, una mayor velocidad de procesamiento, y un mayor aprovechamiento de los recursos.

Requisitos de Soporte.

- Del lado del Servidor:
  Se requiere una computadora que cuente con un servidor web Apache, con soporte para PHP versión 4.0 o superior. Además, se requiere de un servidor de base de datos MySQL, preferentemente superior a las versiones 4.0. Todo lo anterior para una eficiencia óptima, aunque todo el conjunto puede estar en una sola máquina.
- Del lado del cliente:
  Por parte del cliente se requiere un navegador que interprete HTML y las funciones básicas de Java Script, con cualquier sistema operativo.

Requisitos de Portabilidad.

El producto podrá ser utilizado en diferentes plataformas. La estandarización del protocolo de Internet permite la interacción del lado del cliente para los sistemas operativos más difundidos, entre ellos Windows, Linux, Unix, Solaris y otros. No obstante, hay que señalar que la plataforma seleccionada para desarrollar la aplicación fue Windows. La herramienta de desarrollo seleccionada permite además, construir aplicaciones clientes multiplataforma. Del lado del cliente se utilizó Java Script como lenguaje script.

Requerimientos políticos, culturales y legales.

La herramienta propuesta deberá responder a los intereses de la Constitución de la República de Guyana, la Constitución de la República de Cuba, asimismo no existirán prioridades en el servicio según el nivel social, cultural o étnico.

Requerimientos de Software.

- Servidor:
  Para la implantación del sistema se requiere de un servidor con sistema operativo Linux o Windows, a partir de la versión Windows Advanced Server 2000. Además, se necesita tener un servidor web Apache con PHP (versión4.0) o superior y como servidor de base de datos MySQL (versión 4 o superior).
- Cliente:
  Los requerimientos en el lado del cliente para la utilización del sistema, sólo se limitan a tener disponible un navegador web compatible con Java Script.

Requerimientos de Hardware.

- Servidor:
  La máquina servidora debe tener como mínimo las siguientes características de hardware: Procesador Pentium III 450 MHz o superior, 256 Mb de memoria RAM (incluye la utilizada por el Sistema Operativo) y 5Gb de capacidad en disco duro.
- Cliente:
  Las computadoras situadas en los puestos de trabajo de los usuarios requerirán como mínimo un procesador Pentium II, 64 Mb de memoria RAM.
  Estas máquinas deberán estar conectadas en red con el servidor.

Requisitos de Seguridad.

La aplicación deberá garantizar un control estricto sobre la seguridad de la información teniendo en cuenta el establecimiento de niveles de acceso. Se permitirá un acceso sin autenticación solamente para aquellos visitantes que necesiten ver información básica de un estudiante u otra información del programa de estudios, del cual solo podrán obtener información muy general. No se permite otro tipo de acceso sin autorización al sistema.

Se definirá una política de usuarios con privilegios de acuerdo a su rol, lo que asegurará que la información pueda ser consultada de acuerdo a su nivel de acceso.

La información almacenada será consistente y se utilizarán validaciones que limiten la entrada de datos erróneos.

El sistema garantizará que la información esté disponible a los usuarios en todo momento siempre que no existan fallas de fuerza mayor.

2.8 MODELO DE CASOS DE USO DEL SISTEMA.

2.8.1 Actores del sistema.

Los actores representan terceros fuera del sistema que colaboran con él.

Cada trabajador del negocio que tiene actividades a automatizar es un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

Actor	Descripción
1. Invitado	El invitado tiene la posibilidad de ver información general de Guyana y Cuba tales como historia, hechos importantes, noticias e información sobre el programa de estudios en Cuba. También puede visualizar las noticias y buscar un estudiante. (Este usuario no hace falta autentificación) Requerimientos asociados: R39, R40, R42, R43, R44, R45, R46, R47, R48
2. Usuario	Este actor es abstracto porque representa la intersección de las funcionalidades del administrador, el estudiante y un oficial además tiene los derechos como el invitado.
3. Administrador	El Administrador es el que controla todo del sistema. Tiene el derecho de hacer modificaciones a los datos de los estudiantes, las instituciones, las carreras y publicar las noticias e insertar nuevas cuentas para las oficiales que va a acceder la

	información de la base de datos. Requerimientos asociados: Todos los requerimientos del sistema.(R1 – R50)
4. Estudiante	El estudiante puede usar el sistema de mensajería, ver sus datos propios. Tiene que registrarse en el sistema. Requerimientos asociados: R1, R2, R3, R4, R5, R33, R34, R35, R38, R39, R40,R41, R42, R43, R44, R45, R46, R47, R48
5. Oficial	La oficial puede acceder alguna información de la base de datos, pero no puede cambiar los datos. También se puede crear reportes. Requerimientos asociados: R1, R2, R3, R4, R5, R33, R34, R35, R38, R39, R40,R41, R42, R43, R44, R45, R46, R47, R48, R49, R50

Tabla 10. Descripción de los actores del sistema.

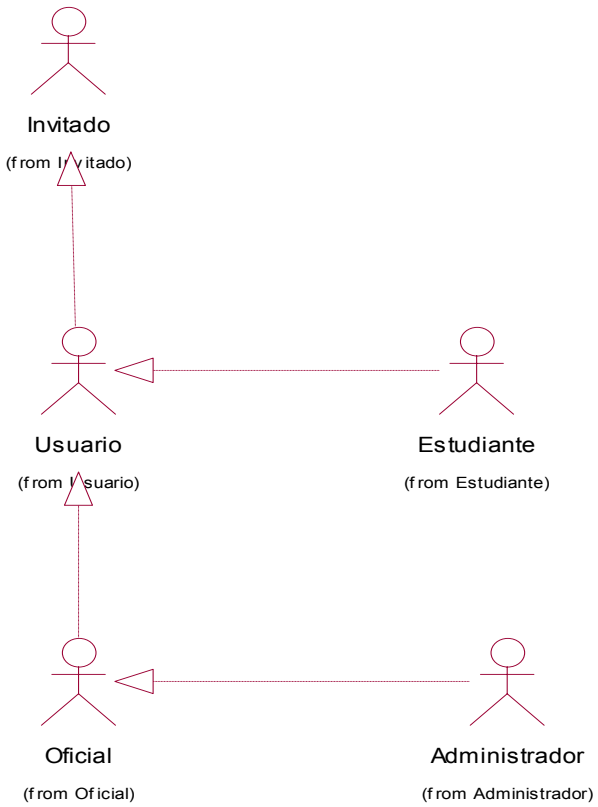


Figura 4: Vista global de los actores del sistema.

2.8.2 Casos de Uso del sistema.

Cada forma en que los actores usan el sistema se representa con un Caso de Uso.

Los Casos de Uso son “fragmentos” de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores. Un Caso de Uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de la secuencia.

El sistema propuesto cuenta con los siguientes casos de uso:

1. Autenticar.
2. Cambiar contraseña.
3. Administrar usuarios.
4. Gestionar estudiante.

5. Registrarse
6. Gestionar nota.
7. Gestionar permiso.
8. Gestionar campo de estudio.
9. Gestionar provincia.
10. Gestionar institución.
11. Gestionar carrera
12. Gestionar asignatura
13. Gestionar noticia
14. Usar mensajería
15. Gestionar imágenes
16. Buscar estudiante
17. Visualizar información general de estudiante
18. Visualizar información particular de estudiante
19. Visualizar notas de estudiante
20. Visualizar permisos de estudiante
21. Visualizar información propia del estudiante
22. Visualizar permisos propias de estudiante
23. Visualizar notas propias de estudiante
24. Visualizar datos propios del estudiante
25. Visualizar campos de estudios
26. Visualizar institución
27. Visualizar carrera
28. Visualizar asignatura
29. Visualizar noticias
30. Visualizar imágenes
31. Visualizar información general del curso de estudios
32. Generar reportes
33. Imprimir reportes

2.8.3 Paquetes y sus relaciones.

Los paquetes son un mecanismo de organización de elementos que subdividen el modelo en otros más pequeños que colaboran entre sí. Esta división debe hacerse sobre la base de los requerimientos funcionales y el dominio del problema; y debe ser reconocible por las personas con conocimiento del dominio.

Para ello se propone asignar la mayor parte de un cierto número de casos de uso a un paquete concreto.

Teniendo en cuenta los criterios antes expuestos se subdividieron los casos de uso propuestos para este sistema en los siguientes paquetes:

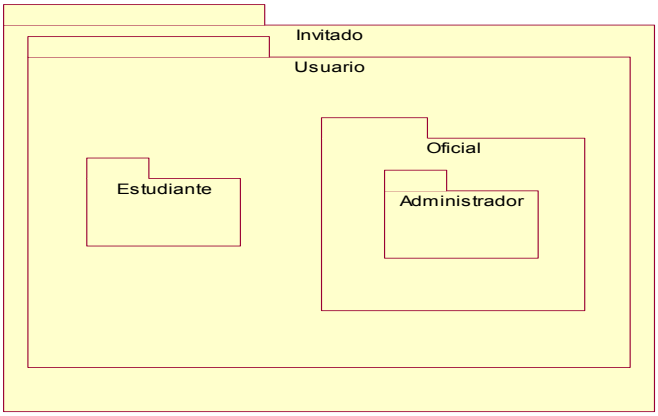


Figura 5: Diagrama de los paquetes por Casos de Uso.

2.8.4 Diagramas de casos de uso del sistema.

Diagrama de casos de uso del paquete invitado Ver Anexo 2.1

Diagrama de casos de uso del paquete usuario Ver Anexo 2.2

Diagrama de casos de uso del paquete oficial Ver Anexo 2.3

Diagrama de casos de uso del paquete estudiante Ver Anexo 2.4

Diagrama de casos de uso del paquete administracion Ver Anexo 2.5

2.8.5 Descripción de los casos de uso del sistema.

Caso de uso	Autenticar
Actores	Usuario (Inicia).
Propósito	Validar el acceso de los usuarios a las diferentes aplicaciones por las que está compuesto el sistema.
Resumen El caso de uso se inicia cuando un usuario intenta acceder al sistema. Este tiene que introducir su login y su contraseña de acceso, el sistema verifica que los datos introducidos por el usuario sean validos, si así fuera se le permite al mismo que acceda a las opciones del sistema que le corresponden y concluye el caso de uso, en caso contrario se le emite un mensaje de error y el caso de uso concluye.	
Referencias	R1
Precondiciones	Los usuarios deben ser registrados dentro del sistema
Post-condiciones	El usuario y la contraseña son válidos y el sistema muestra la pantalla principal del usuario.
Requisitos Especiales	
Prototipo	Ver Anexo3.1

Tabla 11. Descripción del caso de uso de sistema Autenticar.

Caso de uso	Cambiar contraseña
Actores	Administrador, Estudiante, Oficial
Propósito	Que el usuario puede cambiar su contraseña.
Resumen El caso de uso se inicia cuando un usuario decide que va a cambiar su contraseña. El	

usuario selecciona la opción para cambiar contraseña, y el sistema muestra la interfaz de cambiar su contraseña. El usuario hace uso de la interfaz y llena los datos para cambiar su contraseña y el caso de uso concluye cuando el usuario cambia su contraseña. En el caso que no puede cambiar su contraseña, el sistema muestra un mensaje que no puede, y el caso de u uso finaliza.	
Referencias	R 2
Precondiciones	El usuario es autenticado, y su cuenta existe en el sistema.
Post-condiciones	El usuario cambia su contraseña.
Requisitos Especiales	
Prototipo	Ver Anexo 3.2

Tabla 12: Descripción del caso de uso de sistema Cambiar contraseña.

Caso de uso	Administrar Usuarios
Actores	Administrador
Propósito	Permitir que el administrador administrar los datos de un usuario en el sistema.
Resumen Una vez autenticado, el administrador puede elegir cualquier función de actualización, puede agregar algún usuario. En caso que se requiera, se puede eliminar alguno de los usuarios existente o modificar los datos de cualquiera de ellos.	
Referencias	R4, R5, R6
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de los usuarios.
Requisitos Especiales	

Prototipo	Ver Anexo 3.3

Tabla 13: Descripción del caso de uso de sistema Administrar Usuario.

Caso de uso	Registrarse
Actores	Estudiante
Propósito	Que el usuario puede registrarse en el sistema.
Resumen: El caso de uso se inicia cuando un estudiante decide que quiere registrarse en el sistema para usar las funcionalidades del sistema para un estudiante. El estudiante escoge la opción para registrarse, y el sistema muestra la interfaz de registrar un nuevo usuario y verifica si el estudiante puede usar el sistema. El usuario hace uso de la interfaz y llena los datos para registrarse y el caso de uso concluye cuando el usuario se registra en el sistema. En el caso que no puede registrarse, el sistema muestra un mensaje que no puede, y el caso de u uso finaliza.	
Referencias	R3
Precondiciones	El usuario es autenticado, y su cuenta existe en el sistema.
Post-condiciones	El usuario cambia su contraseña.
Requisitos Especiales	
Prototipo	Ver Anexo 3.4

Tabla 14: Descripción del caso de uso de Registrarse.

Caso de uso	Gestionar Estudiante
Actores	Administrador
Propósito	Permitir que el administrador actualizar los datos de un estudiante en el sistema.
Resumen: Una vez autenticado, el administrador puede elegir cualquier función para	

actualizar la información referida a los estudiantes guyaneses en Cuba, puede agregar algún estudiante al sistema. En caso que se requiera, se puede eliminar alguno de los estudiantes existentes o modificar los datos de cualquiera de ellos.	
Referencias	R6, R7, R8
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos del estudiante.
Requisitos Especiales	
Prototipo	Ver Anexo 3.5

Tabla 14: Descripción del caso de uso de sistema Gestionar Estudiante.

Caso de uso	Gestionar nota
Actores	Administrador
Propósito	Permitir que el administrador actualizar las notas de un estudiante en el sistema.
Resumen: El administrador hace una búsqueda para el estudiante, una vez encontrada, puede elegir cualquier función para actualizar la información referida a las notas del estudiante. En caso requerido, se puede eliminar del sistema alguna nota del estudiante o modificar las notas de cualquiera de las ya existentes.	
Referencias	R 9, R10, R11
Precondiciones	El administrador debe estar autenticado, y el estudiante debe estar en el sistema.
Post-condiciones	Quedan actualizadas las notas de un estudiante.
Requisitos Especiales	
Prototipo	Ver Anexo 3.6

Tabla 15: Descripción del caso de uso de sistema Gestionar nota.

Caso de uso	Gestionar permiso
Actores	Administrador
Propósito	Permitir que el administrador actualizar los permisos de un estudiante en el sistema.
Resumen: El administrador hace una búsqueda para el estudiante, una vez encontrada, puede elegir cualquier función para actualizar la información referida a los permisos del estudiante. En caso requerido, se puede eliminar del sistema algún permiso del estudiante o modificar los permisos de cualquiera de las ya existentes.	
Referencias	R12, R13, R14
Precondiciones	El administrador debe estar autenticado, y el estudiante debe estar en el sistema.
Post-condiciones	Quedan actualizados los permisos de un estudiante.
Requisitos Especiales	
Prototipo	Ver Anexo 3.7

Tabla 16: Descripción del caso de uso de sistema Gestionar nota.

Caso de uso	Gestionar campo del estudio
Actores	Administrador
Propósito	Permitir que el administrador gestionar los datos de un campo del estudio en el sistema.

Resumen: Una vez autenticado, el administrador puede elegir cualquier función para actualizar la información referida a los campos del estudio que existen en el país, a las cuales pueden tener acceso. En caso requerido, se puede eliminar del sistema algún campo del estudio o modificar los datos de cualquiera de las ya existentes.	
Referencias	R15, R16, R17
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de un campo del estudio.
Requisitos Especiales	
Prototipo	Ver Anexo 3.8

Tabla 17: Descripción del caso de uso de sistema Gestionar campo del estudio.

Caso de uso	Gestionar provincia
Actores	Administrador
Propósito	Permitir que el administrador gestionar los datos de una provincia en el sistema.
Resumen Una vez autenticado, el administrador puede elegir cualquier función para actualizar la información referida a las provincias que existen en el país, a las cuales pueden tener acceso. En caso requerido, se puede eliminar del sistema alguna provincia o modificar los datos de cualquiera de las ya existentes.	
Referencias	R18, R19, R20
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de la institución.

Requisitos Especiales	
Prototipo	Similar a Anexo 3.9

Tabla 18: Descripción del caso de uso de sistema Gestionar provincia.

Caso de uso	Gestionar institución
Actores	Administrador
Propósito	Permitir que el administrador actualizar los datos de una institución en el sistema.
Resumen Una vez autenticado, el administrador puede elegir cualquier función para actualizar la información referida a las instituciones que existen en el país, a las cuales pueden tener acceso. En caso requerido, se puede eliminar del sistema alguna institución o modificar los datos de cualquiera de las ya existentes.	
Referencias	R21, R22, R23
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de la institución.
Requisitos Especiales	
Prototipo	Ver Anexo 3.9

Tabla 19: Descripción del caso de uso de sistema Gestionar institución.

Caso de uso	Gestionar carrera
Actores	Administrador
Propósito	Permitir que el administrador actualizar los datos de una carrera en el sistema.

Resumen Una vez autenticado, el administrador puede elegir cualquier función para actualizar la información referida a las carreras que existen en el país, a las cuales pueden tener acceso. En caso requerido, se puede eliminar del sistema alguna carrera o modificar los datos de cualquiera de las ya existentes.	
Referencias	R24, R25, R26
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de la carrera.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.9

Tabla 19: Descripción del caso de uso de sistema Gestionar carrera.

Caso de uso	Gestionar asignatura
Actores	Administrador
Propósito	Permitir que el administrador actualizar los datos de una asignatura en el sistema.
Resumen Una vez autenticado, el administrador puede elegir cualquier función para actualizar la información referida a las asignaturas que existen en el país, a las cuales pueden tener acceso. En caso requerido, se puede eliminar del sistema alguna asignatura o modificar los datos de cualquiera de las ya existentes.	
Referencias	R27, R28, R29
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Quedan actualizados los datos de una asignatura.

Requisitos Especiales	
Prototipo	Similar a Anexo 3.9

Tabla 20: Descripción del caso de uso de sistema Gestionar asignatura.

Caso de uso	Gestionar Noticia
Actores	Administrador
Propósito	Permitir que el administrador actualizar una noticia en el sistema.
Resumen Una vez autenticado, el administrador puede elegir cualquier función de actualización, puede agregar alguna noticia nacional, internacional o referida a la programa de estudios. En caso necesario, se puede proceder a eliminar alguna de las noticias existente o modificar los datos de cualquiera de las ya publicadas.	
Referencias	R30, R31, R32
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Queda actualizada las noticias.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.9

Tabla 21: Descripción del caso de uso de sistema Gestionar Noticia.

Caso de uso	Gestionar imagen
Actores	Administrador
Propósito	Permitir que el administrador actualizar una imagen en el sistema.

Resumen Una vez autenticado, el administrador puede elegir cualquier función para actualizar las imágenes presentadas en la galería, puede agregar alguna imagen relacionada con el programa, Guyana o Cuba. En caso que se requiera, se puede eliminar alguna imagen ya existente.	
Referencias	R33, R34
Precondiciones	El administrador debe estar autenticado.
Post-condiciones	Queda actualizada las imágenes
Requisitos Especiales	
Prototipo	Similar a Anexo 3.9

Tabla 22: Descripción del caso de uso de sistema Gestionar imagen.

Caso de uso	Usar mensajería
Actores	Administrador, Oficial, Estudiante
Propósito	Permitir que los usuarios usen la mensajería del sistema.
Resumen Una vez autenticado, el usuario puede elegir cualquier función de la mensajería, puede enviar un mensaje, puede leer sus mensajes, y en caso que se requiera, se puede eliminar algún mensaje ya existente.	
Referencias	R35, R36, R37
Precondiciones	El usuario debe estar autenticado y existe en el sistema.
Post-condiciones	Usar la mensajería del sistema.
Requisitos Especiales	

Prototipo	Ver Anexo 3.10

Tabla 23: Descripción del caso de uso de sistema Usar mensajería.

Caso de uso	Buscar Estudiante
Actores	Administrador, Oficial
Propósito	Buscar un estudiante dado nombre, apellido, o pasaporte
Resumen El caso de uso se inicia cuando un usuario quiere verificar la existencia de un estudiante o en las siguientes situaciones: 1. Al verificar si un estudiante existe en el caso gestionar las notas de un estudiante 2. Al verificar si un estudiante existe gestionar las permisos de un estudiante 3. Obtener los datos de un estudiante en el caso de uso Visualizar Estudiante El caso de uso concluye cuando el sistema le muestra al usuario el resultado de la búsqueda.	
Referencias	R 38
Precondiciones	El estudiante existe en la base de datos.
Post-condiciones	El sistema encontró el estudiante según el criterio de búsqueda.
Requisitos Especiales	
Prototipo	Ver Anexo 3.11

Tabla 24: Descripción del caso de uso de sistema Buscar Estudiante.

Caso de uso	Visualizar información general de estudiante
Actores	Administrador, Oficial, Estudiante, Invitado
Propósito	Permitir el usuario ver información generales de una estudiante, como su nombre, su carrera, la provincia

	donde esta, y su ano.
Resumen El CU se inicia cuando el usuario quiere ver los datos generales de un estudiante. El usuario le busca. El caso de uso termina cuando el sistema muestra la información del estudiante. En el caso que no le encuentra, el sistema muestra un mensaje que no encontró, y el caso de u uso finaliza.	
Referencias	R 39
Precondiciones	El estudiante existe en el depósito de datos.
Post-condiciones	EL usuario ha obtenido la información sobre el estudiante o vea un mensaje que no ha encontrado.
Requisitos Especiales	
Prototipo	Ver Anexo 3.12

Tabla 25: Descripción del caso de uso de sistema Visualizar información general de estudiante.

Caso de uso	Visualizar información particular del estudiante
Actores	Administrador, Oficial
Propósito	Permitir el usuario ver información particulares de una estudiante, como los datos privados como su carnet de identidad, pasaporte, detallas de su contrato etc.
Resumen El CU se inicia cuando el usuario quiere ver los datos particulares de un estudiante. El usuario le busca. El caso de uso termina cuando el sistema muestra la información del estudiante. En el caso que no le encuentra, el sistema muestra un mensaje que no encontró, y el caso de u uso finaliza.	

Referencias	R 40
Precondiciones	El estudiante existe en el depósito de datos, y el usuario debe estar autenticado.
Post-condiciones	EL usuario ha obtenido la información sobre el estudiante o vea un mensaje que no ha encontrado.
Requisitos Especiales	
Prototipo	Ver Anexo 3.13

Tabla 26: Descripción del caso de uso de sistema Visualizar información particular de estudiante.

Caso de uso	Visualizar notas del estudiante
Actores	Administrador, Oficial
Propósito	Permitir el usuario visualizar las notas de un estudiante.
Resumen El caso de uso inicializa cuando el usuario quiere ver las notas de un estudiante existente en el sistema. El usuario le busca el estudiante, y selecciona la opción a ver las notas del estudiante. El caso de uso concluye cuando el sistema muestra la información.	
Referencias	R 41
Precondiciones	
Post-condiciones	El usuario pudo ver las notas de un estudiante.
Requisitos Especiales	
Prototipo	Ver Anexo 3.14

Tabla 27: Descripción del caso de uso de sistema Visualizar notas del estudiante.

Caso de uso	Visualizar permisos del estudiante
Actores	Administrador, Oficial
Propósito	Permitir el usuario visualizar los permisos de un estudiante.
Resumen El caso de uso inicializa cuando el usuario quiere ver los permisos de un estudiante. El usuario le busca el estudiante, y selecciona la opción a ver los permisos del estudiante. El caso de uso concluye cuando el sistema muestra la información.	
Referencias	R 42
Precondiciones	
Post-condiciones	El usuario pudo ver los permisos de un estudiante.
Requisitos Especiales	
Prototipo	Ver Anexo 3.15

Tabla 28: Descripción del caso de uso de sistema Visualizar permisos del estudiante.

Caso de uso	Visualizar campos del estudio
Actores	Invitado, Estudiante, Oficial, Administrador
Propósito	Permitir el usuario visualizar campos del estudio.
Resumen El CU se inicia cuando el usuario elige la opción de listar los campos del estudio existentes en el país a los cuales se puede tener acceso, mostrando datos de interés sobre las mismas. El caso de uso finaliza cuando el sistema muestra la información.	
Referencias	R 43

Precondiciones	
Post-condiciones	El usuario pudo ver los campos del estudio.
Requisitos Especiales	
Prototipo	Ver Anexo 3.16

Tabla 29: Descripción del caso de uso de sistema Visualizar campos del estudio.

Caso de uso	Visualizar provincia
Actores	Invitado, Estudiante, Oficial, Administrador
Propósito	Permitir el usuario visualizar una provincia.
Resumen El CU se inicia cuando el usuario elige la opción de listar las provincias existentes en el país a los cuales se puede tener acceso, mostrando datos de interés sobre las mismas. El caso de uso finaliza cuando el sistema muestra la información.	
Referencias	R 43
Precondiciones	
Post-condiciones	El usuario pudo ver las provincias.
Requisitos Especiales	
Prototipo	Ver Anexo 3.17

Tabla 30: Descripción del caso de uso de sistema Visualizar provincia.

Caso de uso	Visualizar instituciones
Actores	Invitado, Estudiante, Oficial, Administrador
Propósito	Permitir el usuario ver información de las instituciones.

Resumen El CU se inicia cuando el usuario elige la opción de listar las universidades existentes en el país a las cuales se puede tener acceso, mostrando datos de interés sobre las mismas. El caso de uso finaliza cuando el sistema muestra la información.	
Referencias	R 44
Precondiciones	El sistema cuenta con toda la información acerca de las instituciones.
Post-condiciones	EL usuario ha obtenido toda la información acerca de las instituciones.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.17

Tabla 31: Descripción del caso de uso de sistema Visualizar institución.

Caso de uso	Mostrar información de carreras
Actores	Invitado, Estudiante, Oficial, Administrador
Propósito	Permitir el usuario ver información de las carreras.
Resumen El CU se inicia cuando el usuario elige la opción de listar las carreras existentes en el país a las cuales se puede tener acceso, mostrando datos de interés sobre las mismas. El caso de uso finaliza cuando el sistema muestra la información.	
Referencias	R 45
Precondiciones	El sistema cuenta con toda la información acerca de las carreras.
Post-condiciones	EL usuario ha obtenido toda la información acerca de las

	carreras.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.17

Tabla 32: Descripción del caso de uso de sistema Mostrar información de carreras.

Caso de uso	Visualizar Noticia
Actores	Invitado, Estudiante, Oficial, Administrador
Propósito	Permitir el usuario leer una noticia.
Resumen El caso de uso inicializa cuando el usuario intenta ver una de las noticias existentes en el sistema. El sistema busca todas las noticias y muestra los títulos de las noticias existentes. El caso de uso concluye cuando el usuario selecciona una noticia y el sistema la muestra.	
Referencias	R 46
Precondiciones	
Post-condiciones	El usuario pudo ver una noticia.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.17

Tabla 33: Descripción del caso de uso de sistema Visualizar Noticia.

Caso de uso	Visualizar imágenes
Actores	Invitado, Todos los Usuarios (Inicia).
Propósito	Permitir el usuario ver las imágenes.

Resumen El CU se inicia cuando el usuario desea acceder a las imágenes archivadas. El caso de uso termina cuando el usuario tiene la posibilidad de verlos.	
Referencias	R 47
Precondiciones	El sistema cuenta con una serie de imágenes disponibles para mostrar.
Post-condiciones	EL usuario ha accedido a las imágenes escogidas por él.
Requisitos Especiales	
Prototipo	Similar a Anexo 3.17

Tabla 34: Descripción del caso de uso de sistema Visualizar imágenes.

Caso de uso	Visualizar información general del sitio.
Actores	Invitado, Todos los Usuarios (Inicia).
Propósito	Permitir el usuario ver y leer los hechos de Guyana y Cuba.
Resumen El caso de uso inicializa cuando el usuario intenta ver información general del sitio, como hechos de Guyana, Cuba, La asociación de estudiantes, etc. El sistema muestra los hechos. El caso de uso concluye cuando el usuario selecciona una titulo y el sistema muestra el contenido.	
Referencias	R 48
Precondiciones	
Post-condiciones	El usuario pudo ver información general del sitio..

Requisitos Especiales	
Prototipo	Ver Anexo 3.18

Tabla 35: Descripción del caso de uso de sistema Visualizar información general del sitio.

Caso de uso	Generar reporte de estudiantes agrupados por diferentes criterios
Actores	Administrador, Oficial
Propósito	Permitir el usuario ver un listado de estudiantes agrupados por diferentes criterios.
Resumen El caso de uso se inicia cuando un Administrador decide ver listado de los estudiantes. El sistema muestra algunos criterios para la creación del listado. Cuando el Administrador seleccione los criterios el sistema genera el reporte y el caso de uso concluye.	
Referencias	R 50
Precondiciones	Hay estudiantes en el depósito de datos y el usuario es autenticado.
Post-condiciones	El administrador seleccionó los criterios que necesita para la creación del listado y el sistema generó el listado.
Requisitos Especiales	
Prototipo	Ver Anexo 3.19

Tabla 36: Descripción del caso de uso de sistema Mostrar listado de estudiantes agrupados por diferentes criterios.

Caso de uso	Generar reporte estadístico
Actores	Administrador, Oficial
Propósito	Permitir el usuario generar reporte.
Resumen: El caso de uso se inicia cuando un Administrador decide crear un reporte estadístico. El sistema muestra algunos criterios para la creación del reporte. Cuando el Administrador seleccione los criterios el sistema genera el reporte y el caso de uso concluye.	
Referencias	R 51
Precondiciones	Hay estudiantes en la base de datos, y el usuario es autenticado.
Post-condiciones	El usuario seleccionó los criterios que necesita para la creación del reporte y el sistema generó el reporte.
Requisitos Especiales	
Prototipo	Ver Anexo 3.20

Tabla 37: Descripción del caso de uso de sistema Generar reporte

Caso de uso	Imprimir reporte
Actores	Administrador, Oficial
Propósito	Permitir el usuario generar reporte.
Resumen: El caso de uso se inicia cuando un usuario decide crear un reporte y quiere imprimirlo. El usuario escoge la opción para crear PDF reporte. El sistema muestra algunos criterios para la creación del reporte. Cuando el usuario seleccione los criterios el sistema genera el reporte PDF y el usuario puede imprimirlo.	
Referencias	R 52

Precondiciones	Hay estudiantes en la base de datos, y el usuario es autenticado.
Post-condiciones	El usuario seleccionó los criterios que necesita para la creación del reporte y el sistema generó el reporte.
Requisitos Especiales	
Prototipo	Ver Anexo 3.21

Tabla 38: Descripción del caso de uso de sistema Generar reporte

2.9 CONCLUSIONES

- En la etapa de modelación del negocio se especificaron y quedaron reflejadas las reglas a seguir para la posterior implementación del sistema, así como los casos de uso del negocio, siendo descritos mediante su diagrama de actividad. Se identificaron los actores y trabajadores del negocio, entre otros aspectos. Esta etapa de modelación contribuye a derivar los requerimientos del sistema a implementar y a entender la dinámica de la asociación para así poder desarrollar el sitio.

- Se identificaron los objetivos, requerimientos, actores que interactúan con el sistema y la totalidad de los casos de usos. Los casos de uso son descritos y estos describen el comportamiento del sistema y cubren además los requerimientos funcionales.

- Durante el estudio realizado en esta etapa se ha obtenido buenos resultados en cuanto a organización, comprensión del sistema a desarrollar y profundidad con que se debe realizar para que cumpla con -los requisitos planteados, ya que el análisis del sistema permitió llegar a un alto nivel de detalle.

CAPÍTULO 3 – CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

3.1 – **INTRODUCCIÓN**

El presente capítulo se encarga de realizar un análisis de las funcionalidades deseadas y necesarias del sistema propuesto con el objetivo de plantear la concepción general del diseño y cómo se implementa éste. De esta forma se presentan los diagramas de clases web que detallan la interacción de las distintas páginas; se estructura la información que se desea persista a través del diseño de la base de datos; se describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Son también descritos los estándares de diseño y programación seguidos.

3.2 – DIAGRAMA DE CLASES DEL DISEÑO

Un diagrama de clases es una colección de elementos declaratorios del modelo, como clases, tipos y sus relaciones; conectados unos a otros y a sus contenidos en forma de grafo. Se usa como medio para definir las páginas y sus hipervínculos.

A partir de los casos de uso del sistema y haciendo uso de las extensiones de UML para web se definieron los distintos diagramas de clases web que se presentan en Tabla 39.

Núm.	Caso de Uso	Diagrama de clases Web
1	Autenticar.	Anexo 4.1
2	Cambiar contraseña	Anexo 4.2
3	Administrar usuarios	Anexo 4.3
4	Registrarse	Anexo 4.4
5	Gestionar estudiante	Anexo 4.5
6	Gestionar nota	Anexo 4.6
7	Gestionar permiso	Similar a Anexo 4.6
8	Gestionar campo de estudio	Anexo 4.7
9	Gestionar provincia	Similar a Anexo 4.7
10	Gestionar institución	Anexo 4.8
11	Gestionar carrera	Similar a Anexo 4.8
12	Gestionar asignatura	Similar a Anexo 4.8
13	Gestionar noticia	Similar a Anexo 4.7
14	Usar mensajería	Anexo 4.9
15	Gestionar imágenes	Similar a Anexo 4.7
16	Buscar estudiante	Anexo 4.10
17	Visualizar información general de estudiante	Similar Anexo 4.10

18	Visualizar información particular de estudiante	Anexo 4.10
19	Visualizar notas de estudiante	Anexo 4.11
20	Visualizar permisos de estudiante	Similar anexo 4.11
21	Visualizar información propia del estudiante	Anexo 4.12
22	Visualizar campos de estudios	Anexo 4.13
23	Visualizar institución	Similar anexo 4.13
24	Visualizar carrera	Similar anexo 4.13
25	Visualizar asignatura	Similar anexo 4.13
26	Visualizar noticias	Similar anexo 4.13
27	Visualizar imágenes	Similar anexo 4.13
28	Visualizar información general del curso de estudios	Similar anexo 4.13
29	Generar reportes	Anexo 4.14
30	Imprimir reportes	Anexo 4.14

Tabla 39: Clases del Diseño y sus referencias.

3.3 – DISEÑO DE LOS DATOS

3.3.1 – Modelo lógico de datos

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.

A partir de este planteamiento se definieron las clases que participan en el modelo lógico de datos tal y como se muestra en el anexo. (Ver Anexo 5)

3.3.2 – Modelo físico de datos

Cuando se define correctamente el modelo lógico, se hace mucho menos engorroso llegar al modelo de datos o modelo físico como también se le denomina en la metodología RUP de la siguiente forma: *“el modelo de datos representa la estructura o*

descripción física de las tablas de la base de datos y es obtenido a partir del diagrama de clases persistentes".

En el anexo aparece el diagrama clases del modelo físico para el sistema propuesto.

(Ver Anexo 6).

3.4 – MODELO DE IMPLEMENTACIÓN

El modelo de implementación denota la implementación del sistema en términos de componentes y subsistemas de implementación. Describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración, y modularización disponibles en el entorno de la implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros.

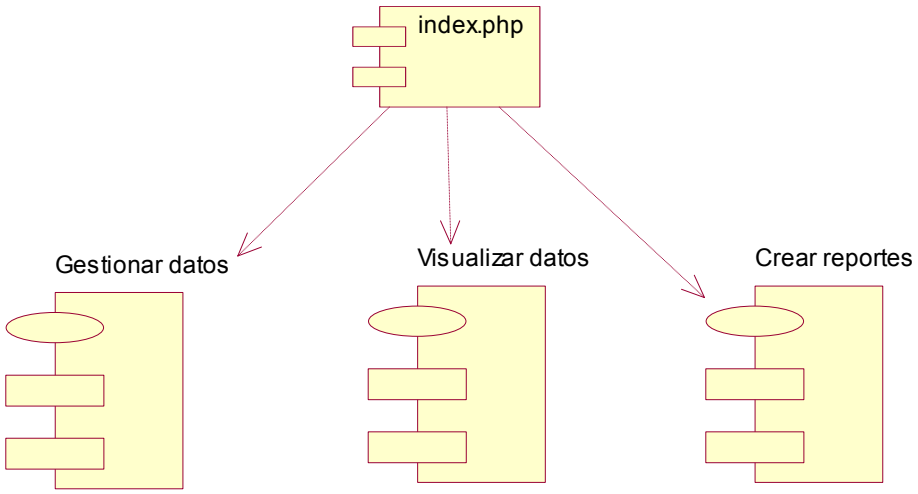


Figura 6: Modelo de implementación del componente SERVIDOR WEB

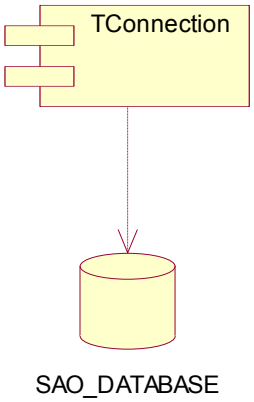


Figura 7: Modelo de implementación del componente SERVIDOR DB

3.5 – DIAGRAMA DE DESPLIEGUE

El modelo de despliegue las demuestra la configuración de los nodos de procesos y los dispositivos en tiempo de correr, la comunicación entre ellos, y las instancias de los componentes y los objetos que residen en ellos.

El modelo denota define las configuraciones de la red física típica, incluyendo aquéllos usados por los usuarios, así como las configuraciones especiales usaron para el desarrollo y prueba.

Los nodos son:

- Terminal Usuario: cualquier SO, 64MB de RAM, HTTP, cualquier navegador del internet que puede interpretar Java script.
- Servidor de Base de Datos: cualquier SO, 128Mb de RAM, 10Gb de Disco duro
- Servidor Web: cualquier SO, 128Mb de RAM

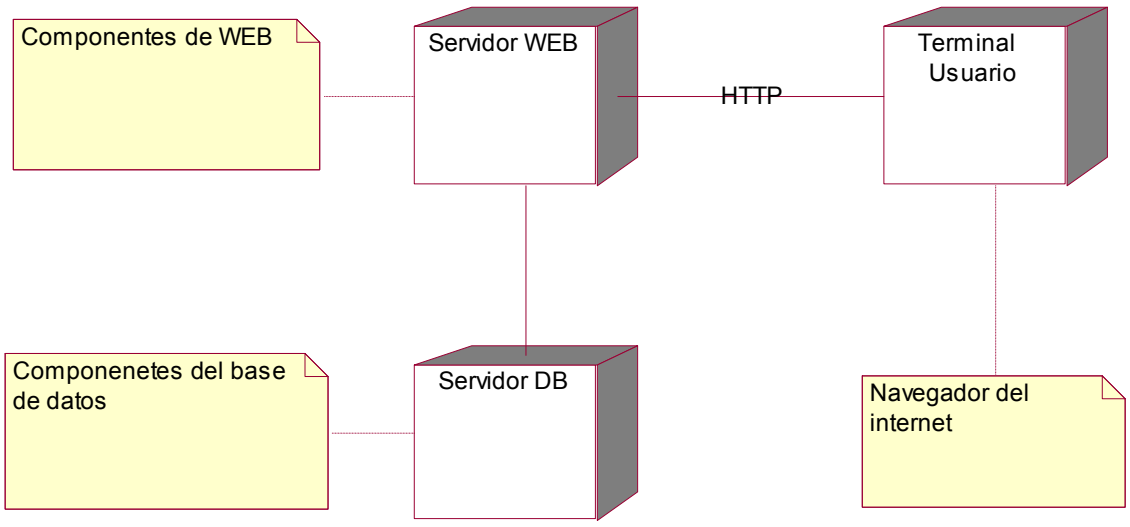


Figura 8: Diagrama del despliegue

3.6 – PRINCIPIOS DE DISEÑO

El diseño de la interfaz de una aplicación, el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones tiene gran influencia en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

3.6.1 – Estándares en la interfaz de la aplicación

Cuando los usuarios exploran una aplicación, sobre todo Web, miran y sienten. La apariencia y sentimiento del sistema constituyen en conjunto el modo en que este se muestra y la personalidad que le transmite al usuario, lo cual conducirá, sobre todo en una aplicación como la propuesta, al éxito o al fracaso. Es por ello que, para lograr la apariencia adecuada y que el usuario se sienta confortable, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sitio, etc., que a continuación se exponen.

Todas las páginas del sitio tienen un estilo común esto garantizó la uniformidad de la apariencia dentro de todo el sitio, además de mantener la homogeneidad definiendo de

forma única la manera de presentación de los contenidos, lo que facilita las tareas de mantenimiento y realización de cambios.

Para transmitir la sensación de confianza, seriedad y profesionalidad se utilizaron colores sencillos. Se minimizaron los contrastes fuertes y elementos gráficos que puedan resultar excesivamente llamativos o resaltantes dentro de las páginas del sitio. Se hizo uso de los colores azul, blanco, negro y gris fundamentalmente y sus diferentes tonalidades de forma balanceada.

La tipografía es clara, legible y sin adornos. El juego de tipos de letra escogido fue Tahoma, Arial, Sans-Serif para los titulares y textos normales.

3.6.2 – Tratamiento de errores

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en este se registra y muestra.

En el sistema las posibilidades de introducir información errónea por parte del usuario son mínimas, pues, aunque en muchas ocasiones el usuario teclea datos y en otras selecciona elementos de la pantalla (información de poca variabilidad), se mantiene un nivel de validación de la información y en caso de errores se le comunica el error cometido a través de mensajes de error. Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios.

3.7 – CONCLUSIONES

En el presente capítulo se mostraron los resultados de la etapa de diseño e implementación del sistema. Se desarrollaron los diagramas de clases de aplicaciones web, el diseño de la base de datos, el modelo de implementación y se definieron además una serie de políticas que tienen en cuenta el diseño de la interfaz, el tratamiento de los posibles errores, la concepción de la ayuda, así como la seguridad y protección de la aplicación.

Todo ello con el objetivo de lograr una herramienta automatizada de trabajo que cumpla con las expectativas del cliente.

CAPITULO IV – ESTUDIO DE FACTIBILIDAD

4.1 – INTRODUCCIÓN

Uno de los objetivos fundamentales de los estudios de factibilidad es disminuir el nivel de riesgo que implica todo proceso investigativo y evitar las pérdidas de recursos que afectan la economía de la entidad donde se lleva a cabo el proyecto en caso de no obtenerse los beneficios que se esperan del resultado de la investigación.

Las Técnicas de Estimación surgen tras la necesidad de planificar y controlar el esfuerzo, costo y tiempo de un proyecto de software (SW).

Las variables a estimar son:

Tiempo y costo: recursos necesarios para obtener el software.

Esfuerzo: total de tiempo que gasta una persona trabajando en el desarrollo del proyecto de software (horas persona / mes persona).

Entre las variantes para la estimación están:

1. Análisis de Puntos de Casos de Uso.

2. Análisis de Puntos de Función y COCOMO II.

Para la estimación de dichas variables se utiliza la primera variante: Análisis de Puntos de Casos de Uso, que es un método de estimación del esfuerzo de desarrollo de un proyecto directamente a partir de los modelos de Casos de Uso y factores de complejidad técnica y ambiente de desarrollo. El modelo de Casos de Uso, la complejidad del sistema y las características del grupo de desarrollo determinan el tamaño del proyecto.

En el capítulo se aborda a través de la técnica de puntos de casos de uso el tema relativo al estudio de la factibilidad del producto, se ofrece una descripción de la planificación del proyecto, así como sus costos asociados, los beneficios tangibles e intangibles que reportaría su elaboración y finalmente el análisis entre los costos y los beneficios para concluir si es o no factibles el desarrollo del sistema.

4.2 – PLANIFICACIÓN POR CASOS DE USOS.

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.

Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, como el Proceso Unificado de SW (Rational Unified Process) en las cuales se propone especificar la funcionalidad de los sistemas mediante la utilización de casos de uso.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

Cálculo de Puntos de Casos de Uso sin ajustar.

UUCP = UAW + UUCW

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Paso 1. Factor de Peso de los Actores sin ajustar (UAW).

El valor de UAW se calcula mediante un análisis de la cantidad de actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Actores	Tipo de actor	Factor de peso
Invitado	Promedio	1
Estudiante	Promedio	1
Oficial	Complejo	2
Administrador	Complejo	2
Total		6

Tabla 40: Factor de Peso de los Actores sin ajustar

En el sistema presentamos cuatro actores, el administrador, un oficial, un estudiante, y un invitado, ya que son personas utilizando el sistema mediante una interfaz gráfica y resultando el valor de peso de los actores sin ajustar:

UAW = 6

Paso 2. Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Para calcular este valor se realiza el análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos, a continuación mostramos la tabla de valores de complejidad de los casos de uso.

Tipo CU	Descripción	Factor	Núm. CU	Resultado
Simple	1-3 transacciones	5	24	120
Promedio	1-4 transacciones	10	6	60
Complejo	Más de 8 transacciones	15	0	0
			Total	180

Tabla 41: Factor de Peso de los Casos de Uso sin ajustar

Paso 3. Determinar los puntos de caso de uso sin ajustar (UUCP).

UUCW = 180

Quedando:

UUCP = UAW + UUCW

UUCP = 180 + 6 = 186

Cálculo de Puntos de Casos de Uso ajustados.

UCP = UUCP x TCF x EF

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

Paso 4. Determinar los factores de complejidad técnicos (TCF).

Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Factor	Descripción	Peso	Valor Asignado	Factor	Comentario
T1	Sistema Distribuido	2	2	4	Es un sistema distribuido
T2	Objetivos de performance o tiempo de respuesta	1	4	4	Tiempo de respuesta muy rápido

T3	Eficiencia del usuario final	1	4	4	Alto eficiencia para Usuario final.
T4	Procesamiento interno complejo	1	1	1	Procesamiento es bajo complejo.
T5	El código debe ser reutilizable	1	3	3	Medio reusabilidad
T6	Facilidad de instalación	0,5	4	2	Muy fácil de instalar
T7	Facilidad de uso	0,5	4	2	Muy fácil de usar
T8	Portabilidad	2	2	4	Baja portabilidad
T9	Facilidad de cambio	1	2	2	No es fácil de cambiar
T10	Concurrencia	1	2	2	Baja concurrencia
T11	Incluye objetivos especiales de seguridad	1	3	3	Medio especializado de seguridad
T12	Provee acceso directo a terceras partes	1	3	3	Medio acceso directo a tercera partes
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	1	1	Normal
			Total	35	

Tabla 42: Factor de complejidad técnica.

TCF = 0.6 + 0.01 x Σ (Peso (i) x Valor asignado (i)).

TCF = 0.6 +0.01 x 35

TCF = 0, 95.

Factor de ambiente.

Paso 5. Determinar el factor de ambiente (EF).

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del factor de ambiente.

Factor	Descripción	Peso	Valor	Factor	Comentario
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	1,5	Experiencia media
E2	Experiencia en la aplicación	0,5	3	1,5	Experiencia media
E3	Experiencia en orientación a objetos	1	5	5	Experiencia alta
E4	Capacidad del analista líder	0,5	4	2	Experiencia alta
E5	Motivación	1	5	5	Alto Motivación
E6	Estabilidad de los requerimientos	2	3	6	Requisito estable
E7	Personal a tiempo compartido	-1	4	-4	Existen en ocasiones que no se puede dedicar el tiempo por complete.
E8	Dificultad del lenguaje de programación	-1	3	-3	medio
			Total	14	

Tabla 43: Factor de ambiente.

El Factor de ambiente se calcula mediante la siguiente ecuación:

EF = 1.4 - 0.03 x Σ (Peso (i) x Valor asignado (i))

EF= 1.4 - 0.03 x 14

EF= 0.98.

Paso 6. Determinar los puntos de caso de uso ajustados (UCP).

UCP = UUCP x TCF x EF donde,

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

(UCP = UUCP x TCF x EF)

UCP =186 x 0,95 x 0,98

UCP =173,17

Paso 7. Determinar el esfuerzo.
De los Puntos de Casos de Uso a la estimación del esfuerzo.

En nuestra investigación según lo establecido se utilizará el factor de conversión donde un punto de caso de uso toma 8 horas – hombre. El esfuerzo horas/hombre viene dado por:

E = UCP x CF

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión.

E = 173,17 x 8

E = 1385,33Horas-Hombre.

El esfuerzo que hemos calculado hasta el momento es solo para la programación por lo que se tiene que utilizar la siguiente tabla para el cálculo total.

Actividad	%	E (Horas/Hombre)
Análisis	10	346,34
Diseño	20	692,68
Implementación	40	1385,36
Pruebas	15	519,51
Sobrecargas	15	519,51
Total	100	3463,4

Tabla 44: Esfuerzo

Esfuerzo Total:

TDES (total) = E (total) / CH (total) TDES: Tiempo de Desarrollo

CH: Cantidad de Hombres

TDES (total) = 3463,4 Hombres/hora / 1 Hombres = 3463,4 horas

El Esfuerzo Total calculado para el desarrollo de la aplicación será de 3463,4 horas-hombre.

4.3 – DETERMINACIÓN DE LOS COSTOS.

Paso 8. Determinar el Costo Total a partir del esfuerzo.

Para poder analizar los costos del trabajo tendremos en cuenta el tiempo para el cual se ha estimado la realización del mismo, la cantidad de personas que trabajaran en él, así como el sueldo que reciben.

El esfuerzo realizado, calculado anteriormente, obtuvo un valor de 3463,4 horas/hombre, de aquí que se considere que un trabajador emplea en su trabajo 24 días de cada mes y que labora diariamente 8 horas.

CMT = E (Total)/ (CT x CDL x CHL)

Donde:

CMT: Cantidad de meses para la realización del trabajo.

CDL: Cantidad de días laborables por mes.

CHL: Cantidad de horas laborable por día.

CT: Cantidad de trabajadores.

E (Total): Esfuerzo total para la aplicación.

Donde:

CMT = 3463,4 /(1 trabajador x 24 días x 8 horas)

CMT = 3463,4 / 192

CMT = 18 meses.

Costo

Para este análisis tendremos en cuenta que el sueldo básico de un trabajador es de $225.00 CUP.

Costo = CMT x CT x $225.00

En nuestra investigación existe 1 persona por lo que:

Costo = 18 x 1persona x $225.00

Costo = $4050 CUP

Calculo de	Valor
Esfuerzo	3463,4
Tiempo estimado	18 meses
Cantidad de personas	1
Costos	$4050
Salario medio	$225

Tabla 45: Resultados del Estudio de Factibilidad.

4.4 – BENEFICIOS TANGIBLES E INTANGIBLES.

Los beneficios de una aplicación pueden ser económicos o de orden social, es decir tangibles o intangibles, siendo estos últimos de tanta importancia como los primeros.

En nuestro caso los beneficios son intangibles por lo que no será un medio de ingreso económico a la entidad pero por el contrario si le será de utilidad a la misma en la gestión y control de información de la oficina de asuntos estudiantiles de forma rápida y confiable.

Beneficios intangibles

1. El negocio ya no está limitado a los horarios de trabajo. A cualquier hora, un usuario con su máquina personal puede conectarse al sitio y realizar lo que desee.
2. Mejor calidad de servicio. Al estar basado en el Internet, los usuarios pueden hacer sus negocios normales solo con un clic en los botones de su PC, sea en casa, o en oficina lo cual es una gran ventaja.
3. Centralizar todos los datos e informaciones del negocio en un sitio para facilitar la búsqueda de información.
4. Facilitará la actualización de datos e información del la oficina de asuntos estudiantiles.
5. Facilitará la comunicación entre los estudiantes, entre si.
6. Facilitará la comunicación entre los estudiantes y las entidades que muestran interés en los estudiantes y graduados.
7. El oficial de asuntos estudiantes cuenta con mayor tiempo para cumplir el resto de sus funciones ya que anteriormente se mantenía ocupado con la creación de reportes.

4.5 – ANÁLISIS DE COSTOS Y BENEFICIOS

Para poder analizar los costos del trabajo tendremos en cuenta el tiempo para el cual se ha estimado la realización del mismo, la cantidad de personas que trabajaran en él, así como el sueldo que reciben.

La publicación del sitio en Internet traerá consigo considerables beneficios intangibles. Según el estudio de factibilidad realizado, el costo de la realización de este sistema es de aproximadamente $4050.

Teniendo en cuenta el análisis realizado de los costos y beneficios que reporta el sistema y que no fue necesario realizar inversiones en equipos técnicos para el desarrollo del producto, se concluye que ha sido factible llevar a cabo la realización del software.

4.6 – CONCLUSIONES

En esta etapa se especificaron el estudio de factibilidad del software en cuanto el tiempo de desarrollo, esfuerzo, el costo a desarrollarlo, el análisis de los costos y beneficios y el análisis de los beneficios tangibles e intangibles, se concluye que la realización del sitio ha sido factible.

El estudio de factibilidad constituye una de las bases para evaluar el trabajo de investigación y es una valiosa herramienta que permite establecer con seguridad el alcance, el enfoque y los diferentes aspectos que deben considerarse al efectuar el análisis y la evaluación económica de los proyectos de software.

La herramienta propuesta trae consigo una serie de beneficios sobre todo intangibles para la organización, pero no menos necesarios e importantes, porque va a contribuir a mejorar su funcionamiento, lo que indica que es factible implementar la herramienta propuesta. Una vez terminado el estudio de factibilidad del sistema, se estima un tiempo de 18 meses para su construcción por un hombre y su costo asciende a $4050.

CONCLUSIONES

Durante el desarrollo de este Trabajo de diploma se pudo realizar un análisis de los procesos de gestión de la información relacionados con los estudiantes guyaneses en Cuba.

También se realizó el análisis y diseño del sistema para la gestión de esta información.

Finalmente se implementó la aplicación teniendo en cuenta las características definidas en los procesos de análisis y diseño.

Con el cumplimiento del objetivo propuesto en el trabajo se pudo obtener una aplicación que satisface los requerimientos determinados inicialmente.

RECOMENDACIONES

1. Implantar el sistema para su definitivo uso en la oficina de asuntos estudiantiles.
2. Efectuar la visualización del sistema en otros idiomas para aquellos usuarios que no dominen el idioma inglés.
3. Implementar el sistema en un servidor basado en Cuba para los estudiantes que están en Cuba y no tiene acceso a internet.
4. Para futuras épocas se recomienda que este software se generalice y se ponga en práctica – o en su defecto herramientas similares – en el resto de las embajadas que tienen su sede en la República de Cuba, ya que es una excelente vía para optimizar el trabajo y la gestión de Recursos Humanos en estas oficinas, sobre todo si se trata de asuntos relacionados con los estudiantes.

REFERENCIAS BIBLIOGRAFÍAS

[1] Concepto de gestión. Enciclopedia Universal Ilustrada Europeo Americana.-- Barcelona: Hijos de J. Espasa, 1926 – TXXV –p.1508.

[2] Martínez Iglesias, Marisol Isabel. Estrategia didáctica cooperativa e intercultural para atender la diversidad cultural en español como lengua extranjera en la Facultad Preparatoria de Cienfuegos / --Tesis para optar por el grado de Doctor en Ciencias Pedagógicas, Universidad de Cienfuegos (Cf), 2006. -- h.15.

[3] Rodríguez Febles, Janet. Sistema Informático para la Gestión Integral de Comedores (SISCOMED) / -- Trabajo de Diploma, CUJAE. (UH), 2005. --h.19

[4] Arquitectura Cliente Servidor. Tomado De: www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml, 2008.

[5] Cuba. Universidad de las Ciencias Informáticas. Arquitectura: Manual del profesor -- La Habana: [sn], 2007. –p189

[6] Programación Web. Tomado De: http://www.arsys.es/ ayuda/directorio/programación/windows.htm, 2007.

[7] Soto, Tay-Lien. HTML y Cascading Style Sheets. Giga (España) (2): 41-45.

[8] Álvarez, Rubén. Páginas dinámicas vs HTML. Tomado De: http://www.desarrolloWeb.com/articulos/238.php?manual=7, 2008

[9] Álvarez, Rubén. Concepto de páginas dinámicas. Tomado De: http://www.desarrolloWeb.com/articulos/237.php?manual=7, 2008

[10] Puerta, Leslie. JavaScript. Giga (España) (2):21.

[11] Álvarez, Miguel Ángel. Introducción a Java Script. Tomado De: http://www.desarrolloWeb.com/articulos/490.php?manual=2, 2008

[12] Alvarez Marañón, Gonzalo. ActiveX, ¿amigo o enemigo? PC World, (Mexico) (180): 237, Octubre 2001.

[13] Chappel, David y Linthicum, David S. Active X Demystified. Tomado De: http://www.terra.com./informatica/que-es/activex.cfm, 2008.

[14] Mariñas, Luis. Applets de Java. Tomado De: http://www.pangea.org/pacoc/manuales/manual4.htm, 2008.

[15] Rodríguez Herrera, Daniel. Applets Java. Tomado De: http://www.programacion.net/tutorial/applets, 2008.

[16] López Pedraza, Oscar. Páginas Web Dinámicas. Giga (España) (2): 41.

[17] Software Linux: Programas y Aplicaciones de Linux. Tomado De: http://linux.bankhacker.com/software/Apache, 2008.

[18] Álvarez, Rubén. Introducción a la programación en ASP. Tomado De: http://www.desarrolloWeb.com/articulos/244.php?manual=8, 2008.

[19] Álvarez, Rubén. Trabajar con bases de datos en ASP. Tomado De: http://www.desarrolloWeb.com/articulos/256.php, 2008

[20] Álvarez, Rubén. Lenguajes del lado del servidor o cliente. Tomado De: http://www.desarrolloWeb.com/articulos/239.php?manual=7, 2008.

[21] Manual de PHP básico. Tomado De: http://www.Webestilo.com/php/php00.phtml, 2008.

[22] Tutorial de PHP. Tomado De: http://www.openresources.com/es/magazine/tutoriales/php/e1, 2008.

[23] Programación en php. Tomado De: http://www.desarrolloweb.com/manuales/12, 2008.

[24] Tomas, Alex. ¿Por elegir PHP? Tomado De: http://www.noticias.com

[25] Matos, Rosa María. Introducción al trabajo con Base de Datos: material para uso Docente. -- Ciudad de La Habana: [sn], 2004 --p.4.

[26] Hernández, Jorge Adriel. Procedimientos almacenados en SQL Server. Giga , (España)(6): 40, 41

[27] Servidores de Base de Datos. Tomado De: http://www.soportelinux.com.pe/menu/soluciones.php?solucion_id=2, 2008

[28] Plataforma donde corre el MYSQL. Tomado De: http://www.mysql.com/downloads/mysql-3.23.html, 2008.

[29] Jacobson, I. “El Proceso Unificado de Desarrollo de software”. Massachusetts, EUA: Addison-Wesley 2000. -- p. 25.

[30] Ibídem --p. 58.

[31] Ibídem --p. 116.

[32] Ibidem – p.30.

[33] Ibidem – p.34.

[34] Ibidem – p.35.

BIBLIOGRAFÍA

- Conallen, J. Modeling Web Application Architectures with UML/ Jim Conallen, Massachusetts, EUA: Addison-Wesley, 1999. --360p.
- Conallen, J. Building Web Applications with UML Second Edition/ Jim Conallen, EUA, Massachusetts: Addison-Wesley, 2002. --496p.
- Eckel, B. Professional JavaScript™ for Web Developers/ Bruce Eckel, EUA: Indiana Wiley Publishing Inc. 2004. – p. 159-198.
- Griffiths, P. The Best-Practice to XHTML & CSS/ Patrick Griffiths, EUA: Pearson Education.2007. – 426p.
- Melonie, J. PHP5 Fast & Easy Web Development/ Julie Melonie, EUA: Thomson Course Technology PTR. 2004. – 450p.
- Luke Welling, Thomson, Laura. PHP and MySQL Web Development / Laura Thomson, Luke Welling. –Third Edition, EUA: Sams, 2004. – 260p.
- Peralta, M. Estimación del Esfuerzo basados en casos de uso/ Mario Peralta, Argentina, Instituto Tecnológico de Buenos Aires, 2004. p1-16.
.
- Goodman, D. JavaScript Bible / D. Goodman. EUA: Hungry Minds. 2001. — p. 23- 36.

GLOSARIO DE TÉRMINOS

A

ActiveX: Es un objeto que se programa en un lenguaje como C++, Pascal y luego es compilado en una dll. Se registra en el Servidor Web y se utiliza para dotar a las páginas del servidor web de funciones adicionales como el acceso a Bases de Datos, ficheros del servidor.

API: Interfaz de programas de aplicación (Siglas en Inglés)

Applet: Un *applet* es un pequeño programa ejecutable escrito en lenguaje Java, que se coloca en un servidor Internet.

B

BDEE: Base de Datos de Estudiantes Extranjeros

BDEDUCACION: Base de Datos Educación

C

CGI: La página web no procesa por si sola los datos suministrados a ella, ni tampoco lo hace directamente el servidor, es un *script* quien proporciona las instrucciones al servidor web para que envíe los datos, esto lo hace a través de la CGI.

COM: Modelo de Objeto Componente (siglas en Inglés).

CSS: Página de Estilos de Código (siglas en Inglés)

D

DOM: Document Object Model

DLL: Bibliotecas de enlaces dinámicos. (Siglas en Inglés)

H

HTTP: Protocolo de la transferencia Hipertexto (siglas en inglés)

HTML: Lenguaje de Marcado de Hipertexto (siglas en inglés). Lenguaje estándar que define la apariencia de las páginas web.

I

IIS: Servicios de la información del internet. (Siglas en Inglés)

M

MD5: En criptografía, MD5 (acrónimo de Message-Digest Algorithm 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991.

MES: Ministerio del Educación Superior

N

Navegador o browser: Programa o aplicación diseñando para la búsqueda y presentación de páginas web.

O

ODBC: Conectividad de bases de datos abiertos (Siglas en Inglés)

ONABE: Oficina Nacional de Atención a Becarios Extranjeros

OP: Oficina del Presidente (siglas en inglés)

P

Página web: Archivos de texto que contienen un código interpretable por los navegadores. Dicho código ordena como mostrar los datos en pantalla.

Protocolo: Un estándar para la interpretación de señales transmitidas

PSM: Ministerio de Servicios Públicos (siglas en inglés)

S

SAO: Oficina de los asuntos estudiantiles (siglas en inglés)

SACBEXT: Sistema Automatizado de control de Becarios Extranjeros.

Servidor Web: Computadora con altas prestaciones que se utiliza para ofrecer el servicio WWW a los usuarios dentro de una red. En el Servidor se alojan principalmente documentos HTML, páginas dinámicas desarrolladas con alguna tecnología como ASP, que pudieran tener acceso a una Base de Datos que reside en el mismo o en otro Servidor.

Sitio Web: Conjunto de páginas web estáticas y dinámicas que interactúan unas con otras y con diversos recursos en un servidor Web, incluidas bases de datos.

SQL: Lenguaje de consultas estructuradas. (Siglas en Inglés).

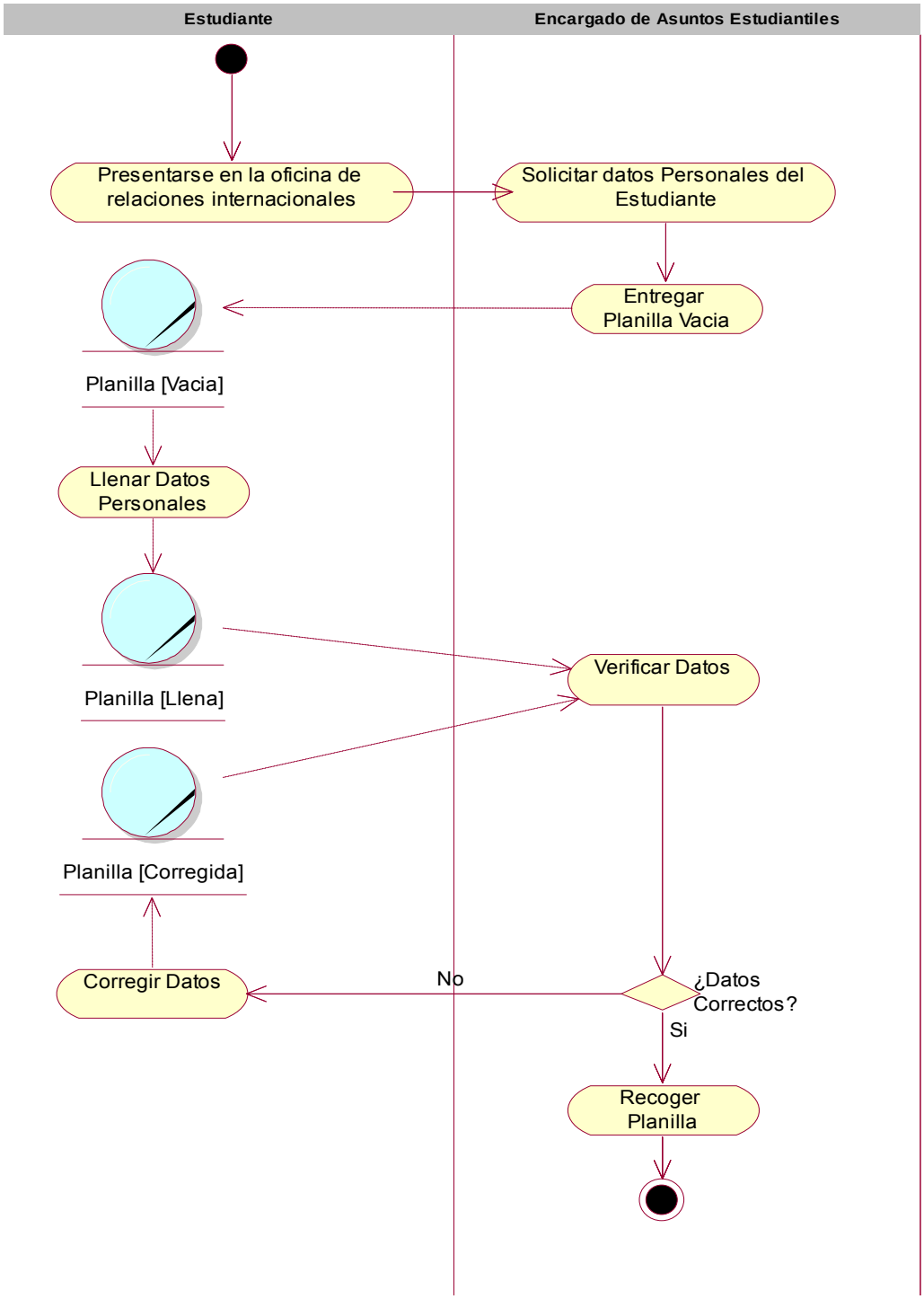
X

XML: Lenguaje de Marcado Ampliable o Extensible (siglas en Inglés)

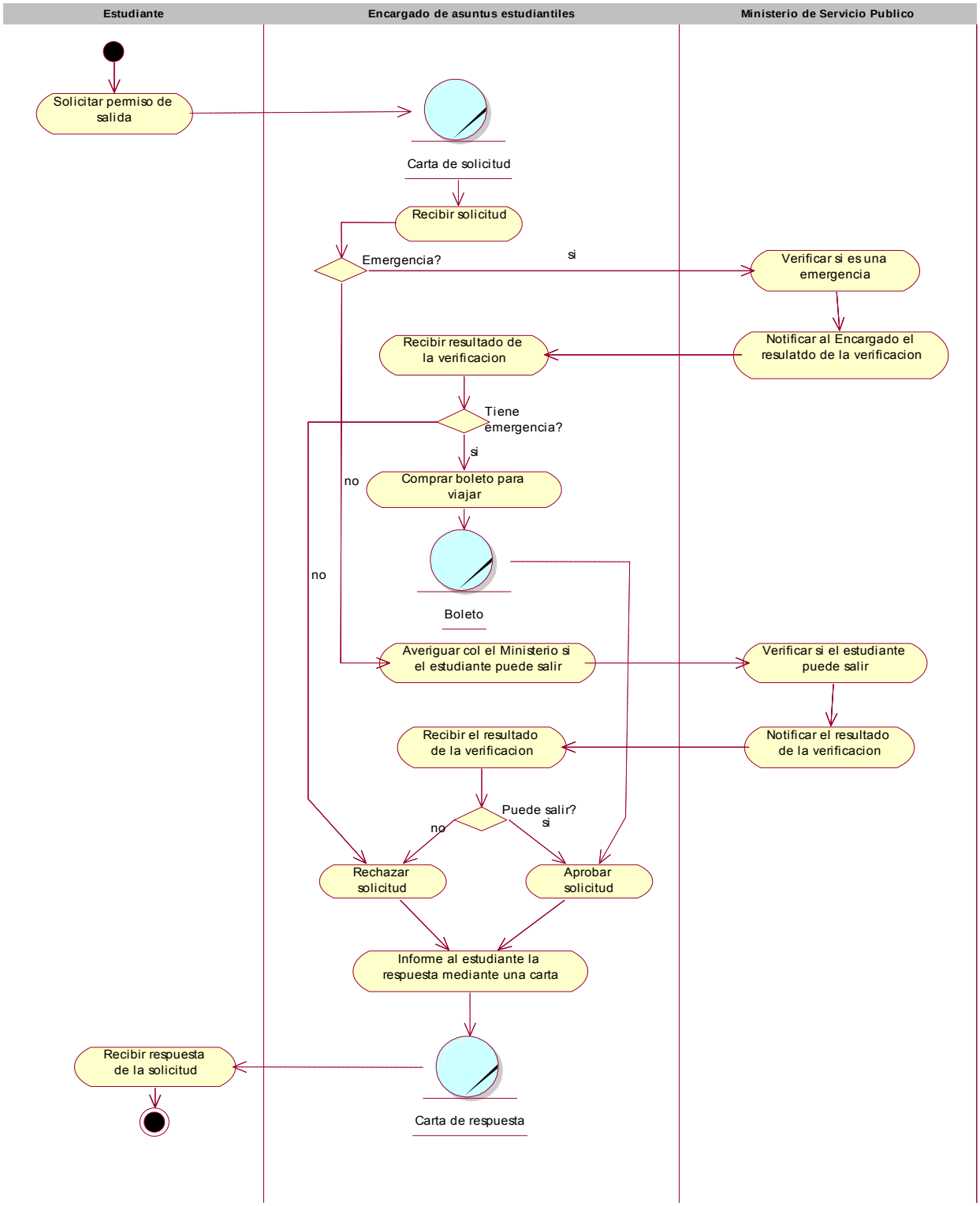
W

ANEXOS

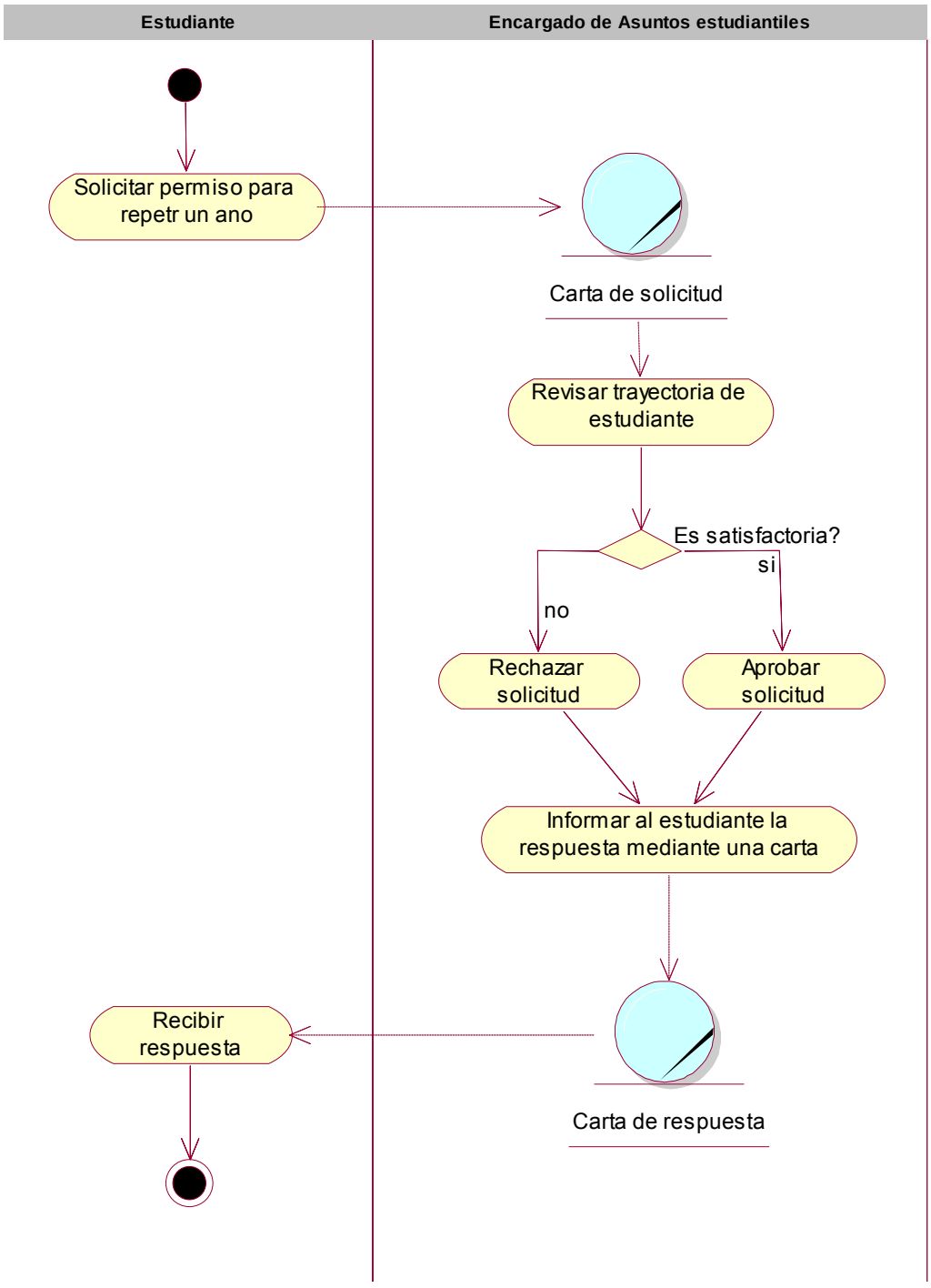
Diagramas de actividades



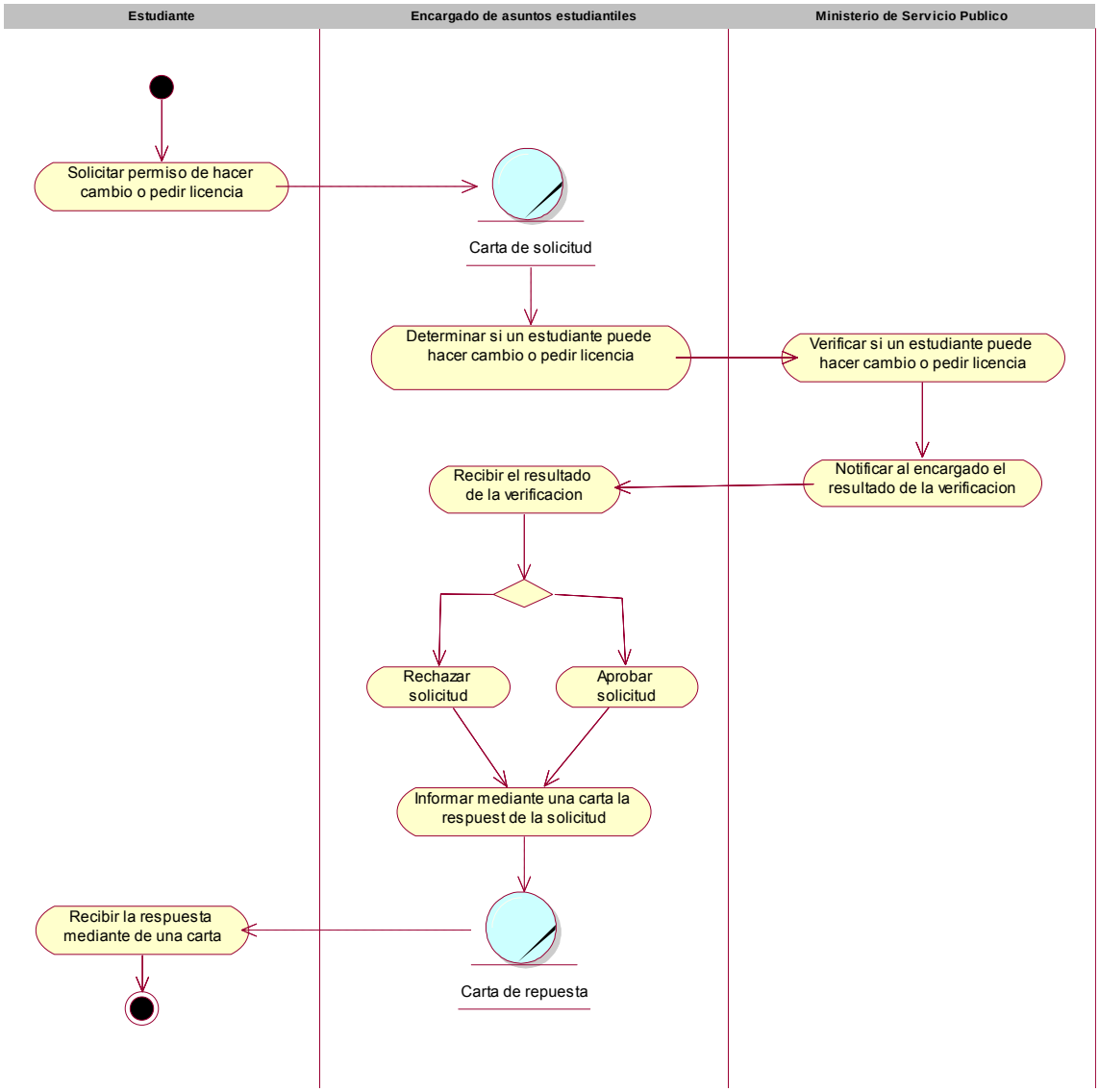
Anexo 1: Diagrama de Actividades de Registrar datos personales



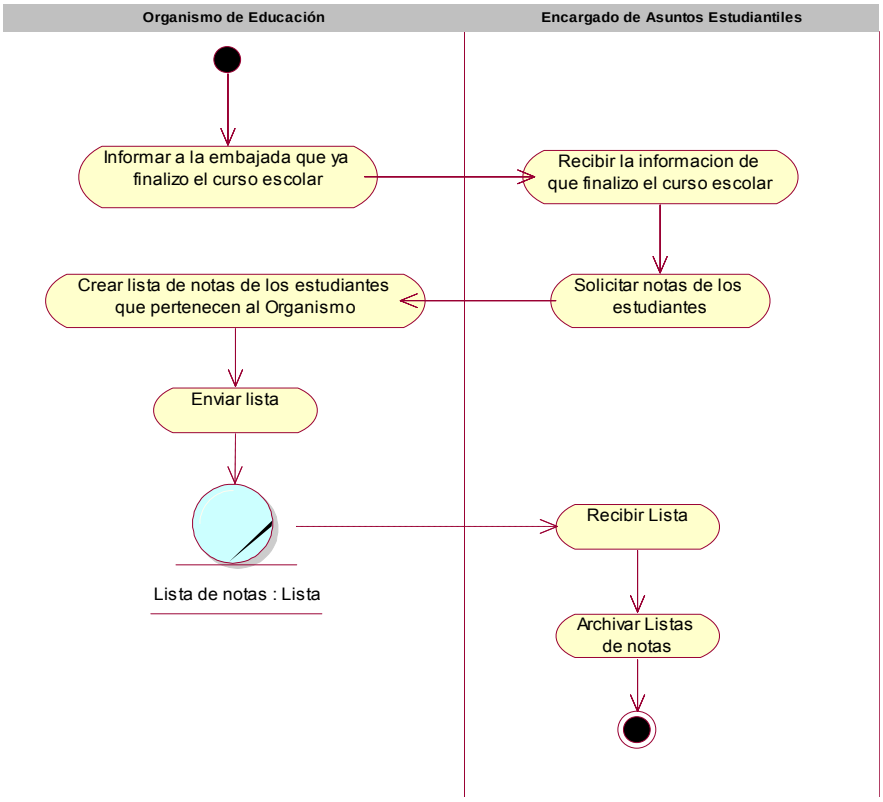
Anexo 1.2: Diagrama de actividades del Caso de Uso Solicitar Permiso de Salida



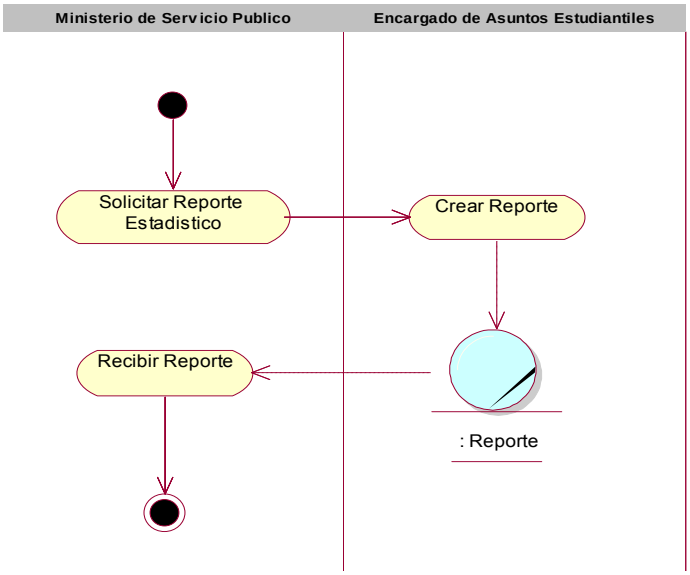
Anexo 1.3: Diagrama de actividades del Caso de Uso Solicitar Permiso de repetir un ano



Anexo 1.4 Diagrama de Actividades de Uso Solicitar Permiso para Hacer un Cambio o Pedir Licencia

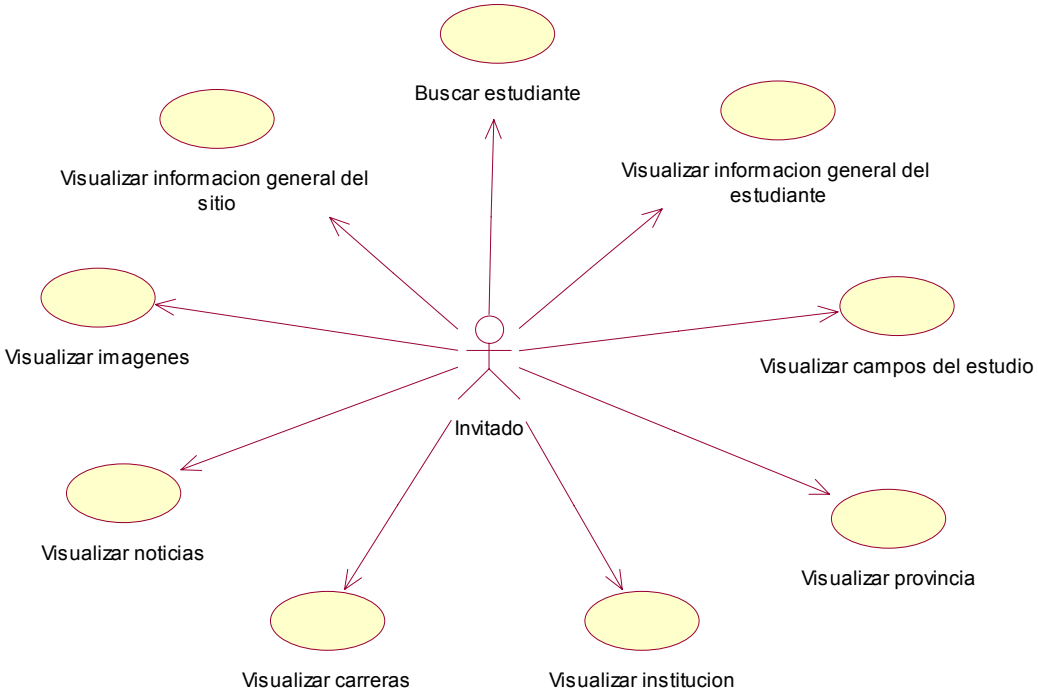


Anexo 1.5 Diagrama de Actividades de Archivar notas de los estudiantes

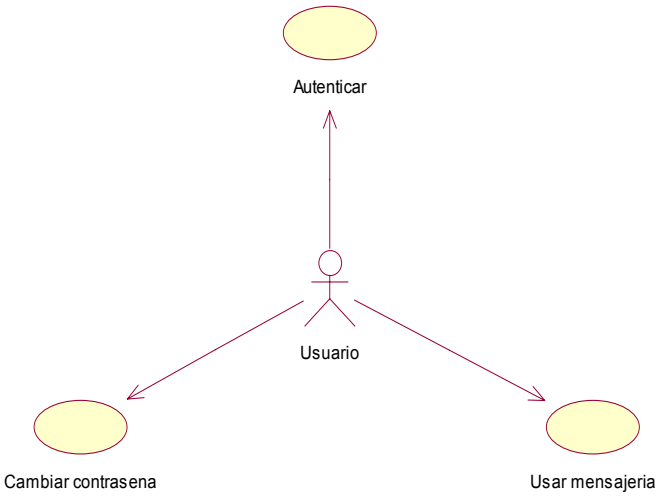


Anexo 1.6 Diagrama de Actividades de Uso Solicitar Reporte

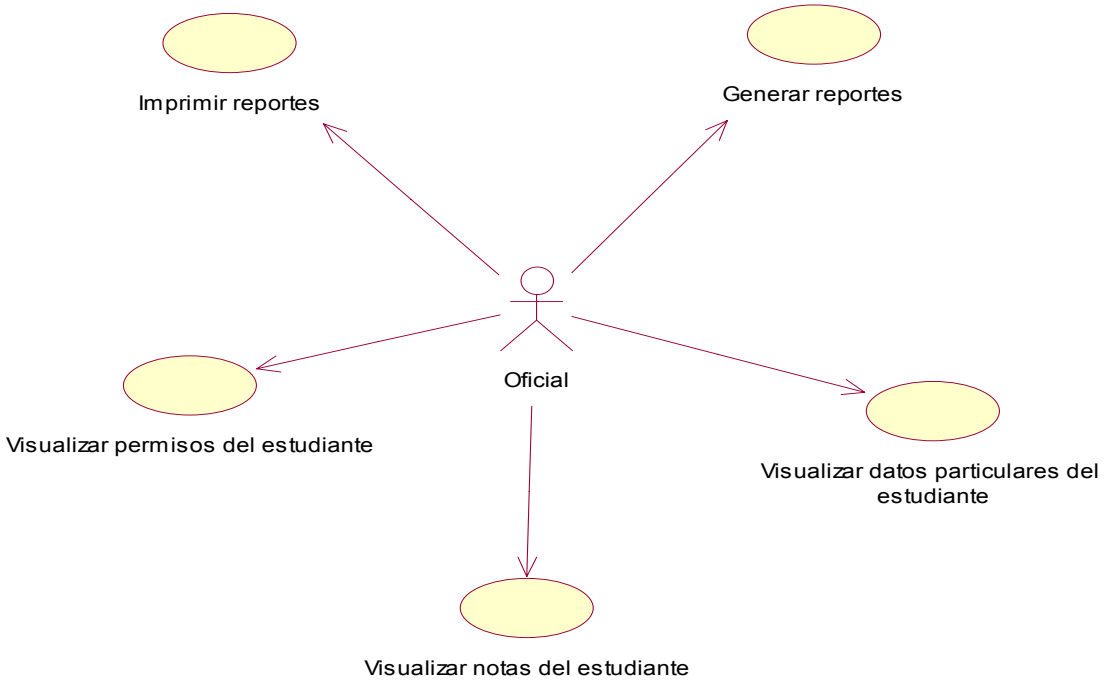
Diagramas de los casos del uso del sistema



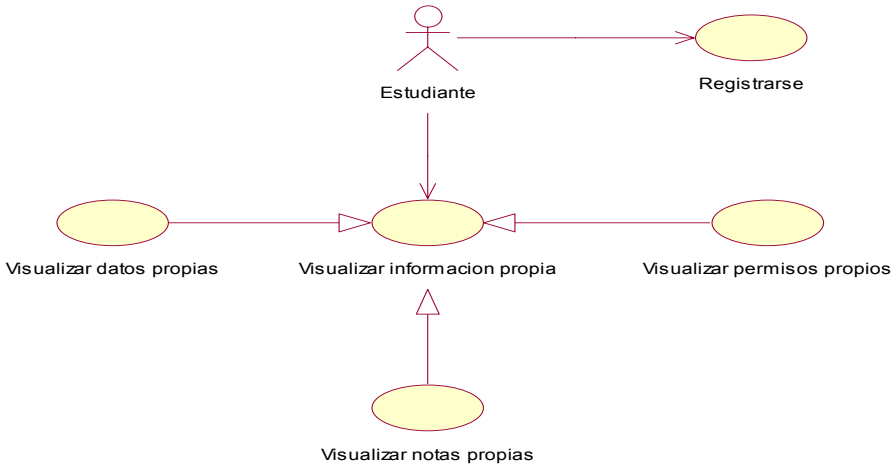
Anexo 2.1 Diagrama de casos de uso del paquete invitado



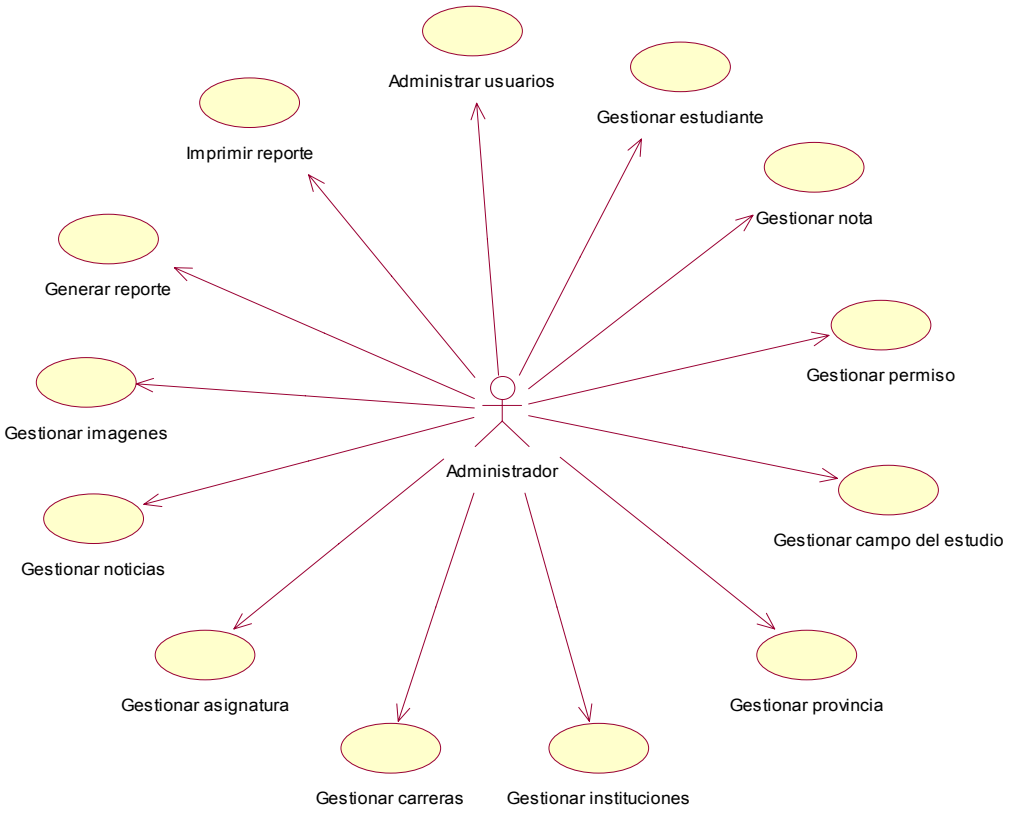
Anexo 2.2 Diagrama de casos de uso del paquete usuario



Anexo 2.3 Diagrama de casos de uso del paquete oficial

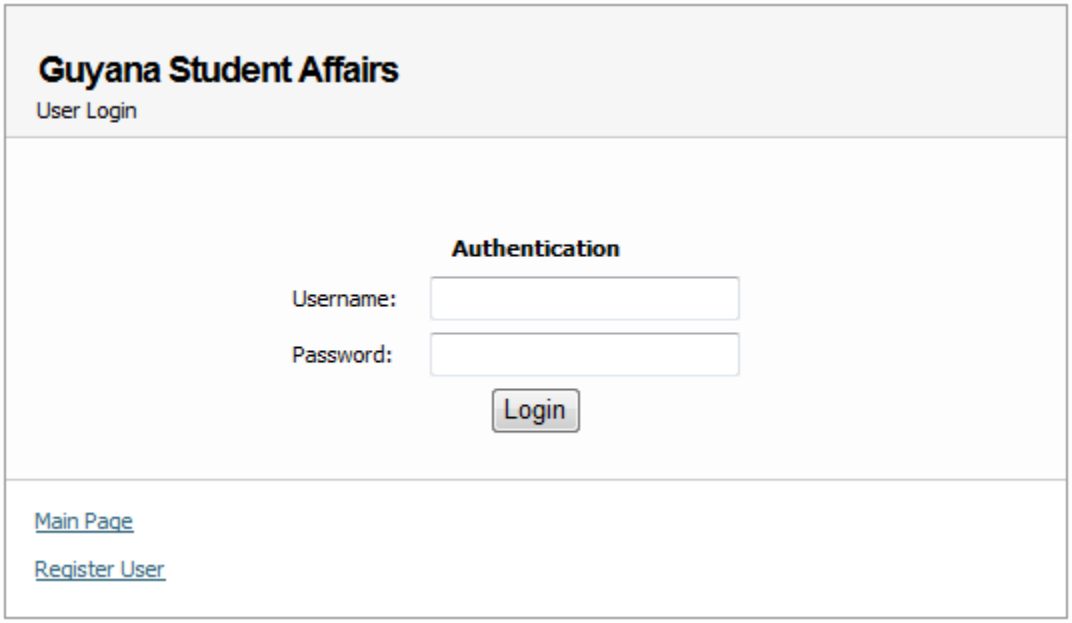


Anexo 2.4 Diagrama de casos de uso del paquete estudiante

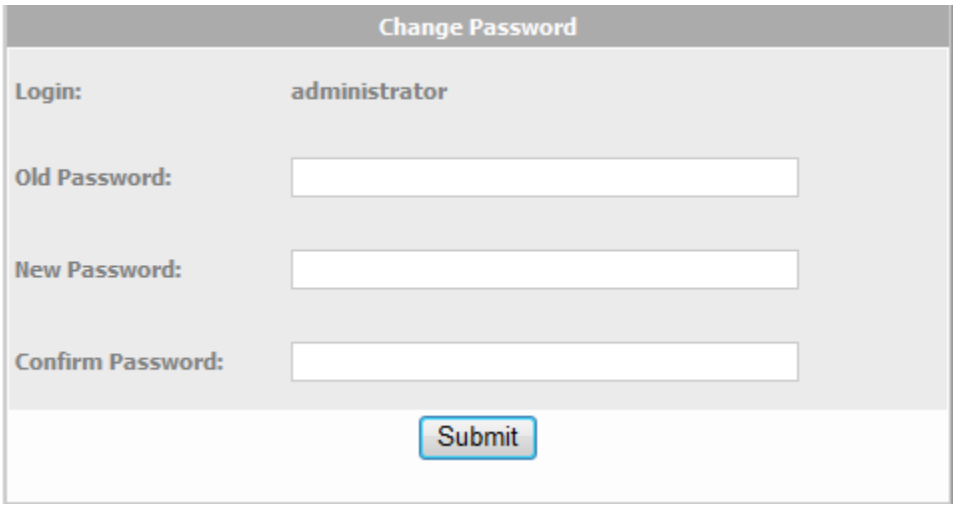


Anexo 2.5 Diagrama de casos de uso del paquete administracion

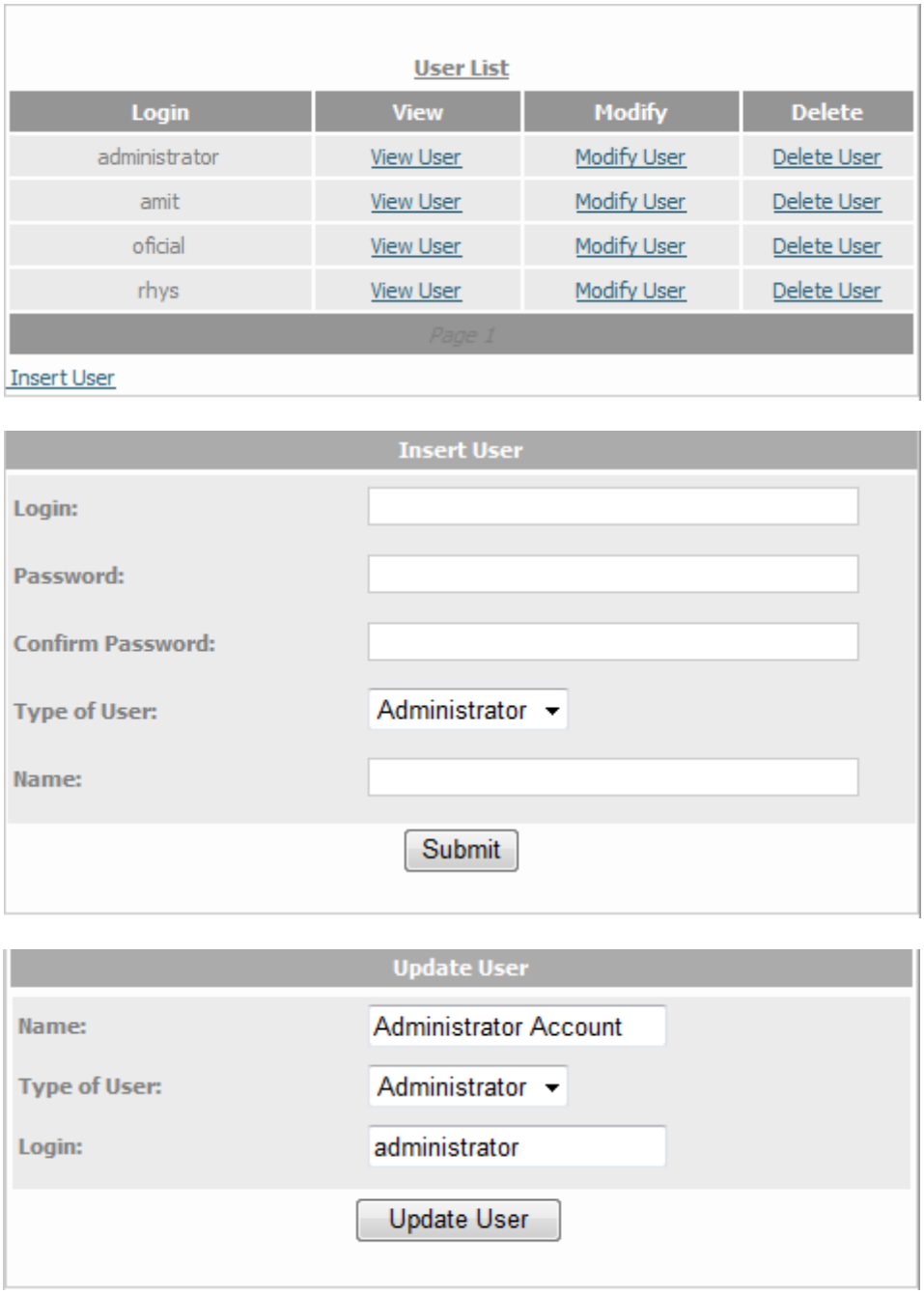
Prototipos del sistema



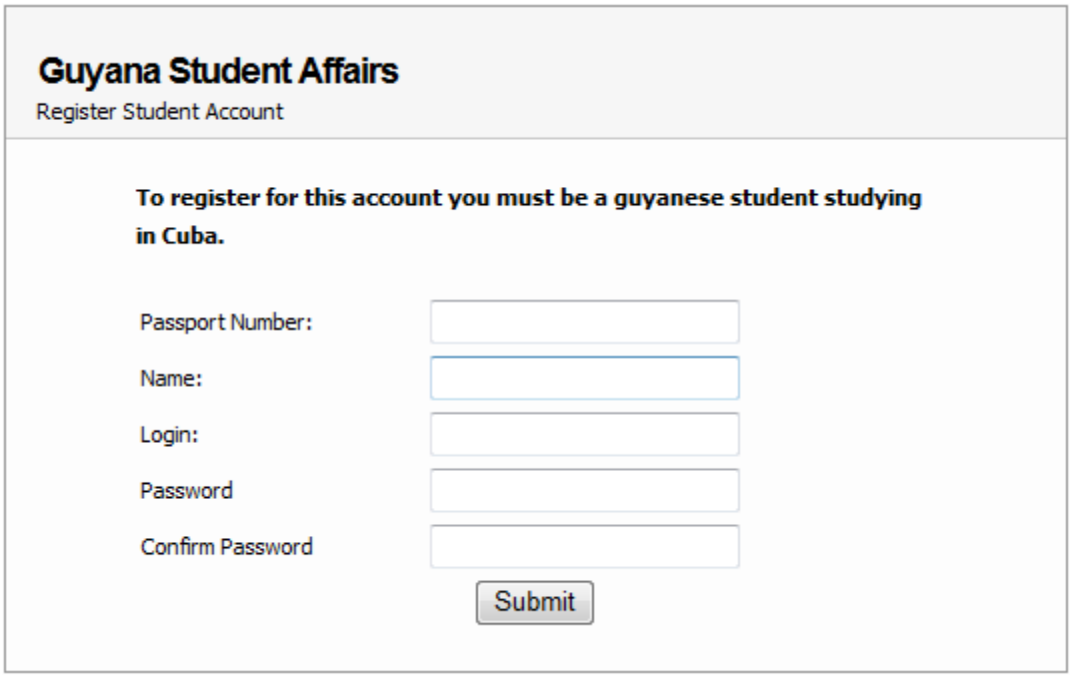
Anexo 3.1 Prototipo “Autenticar”



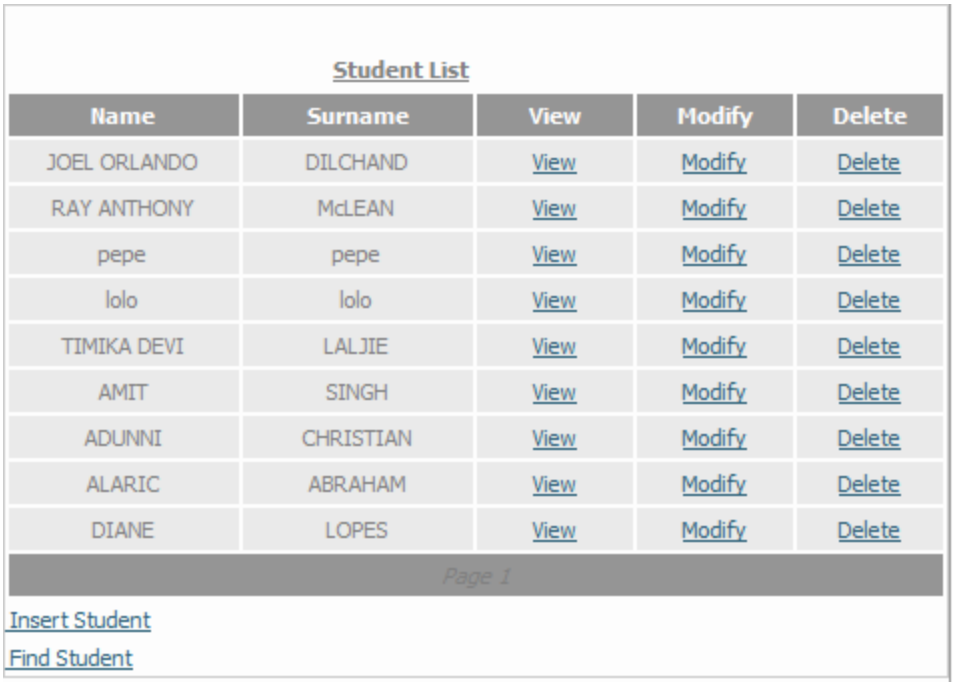
Anexo 3.2 Prototipo “Cambiar contraseña”



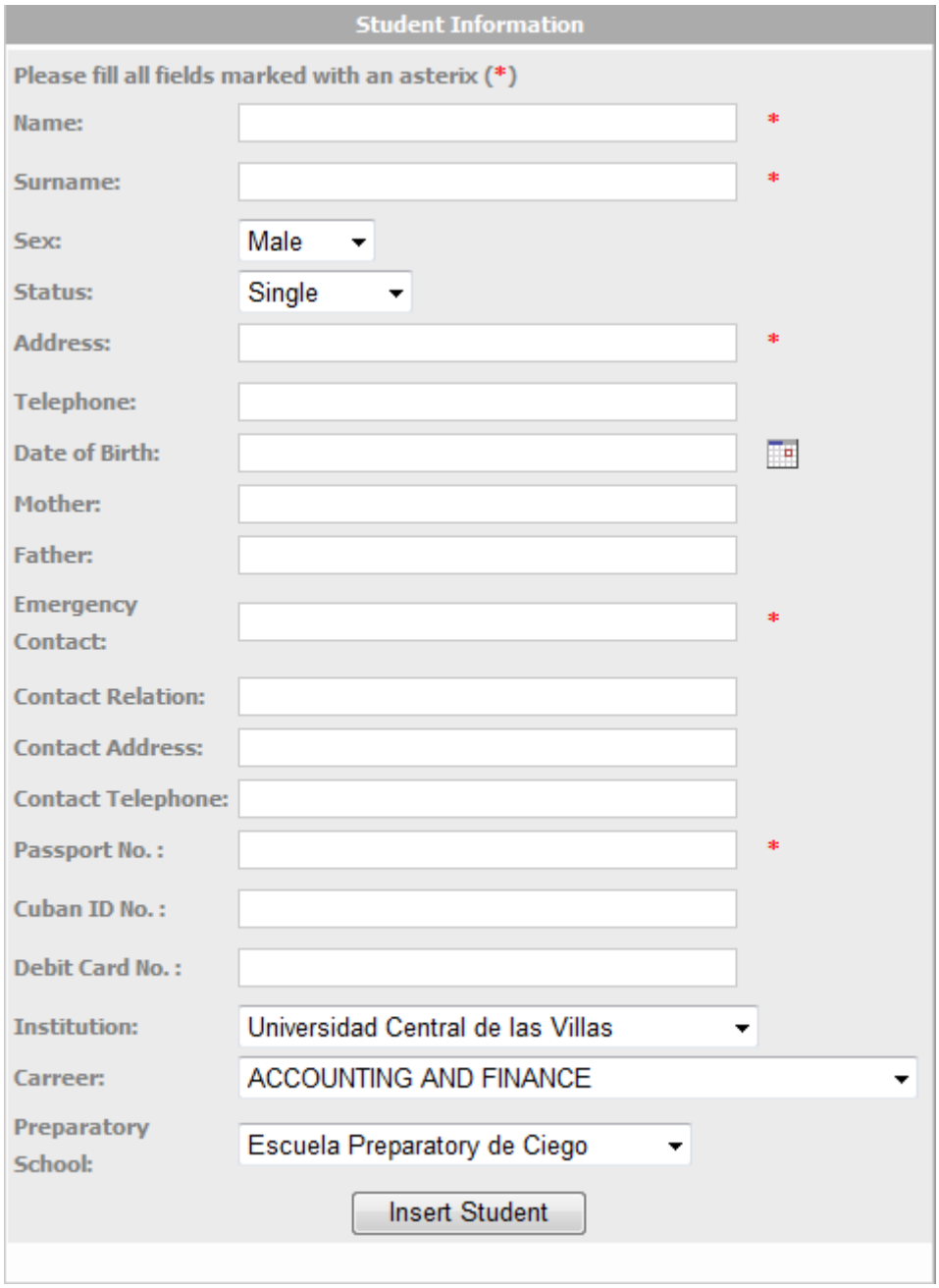
Anexo 3.3 Prototipo “Administrar Usuarios”



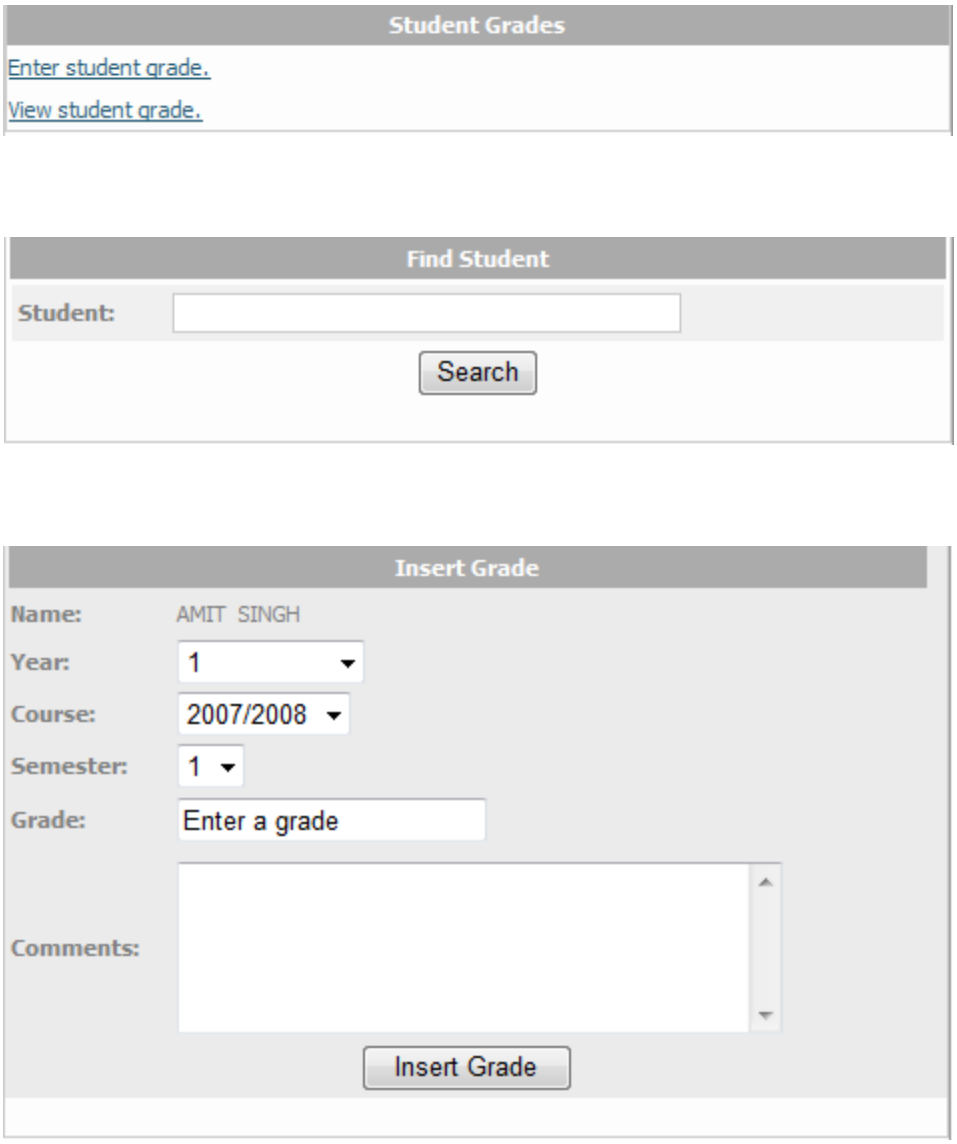
Anexo 3.4 Prototipo “Registrarse”



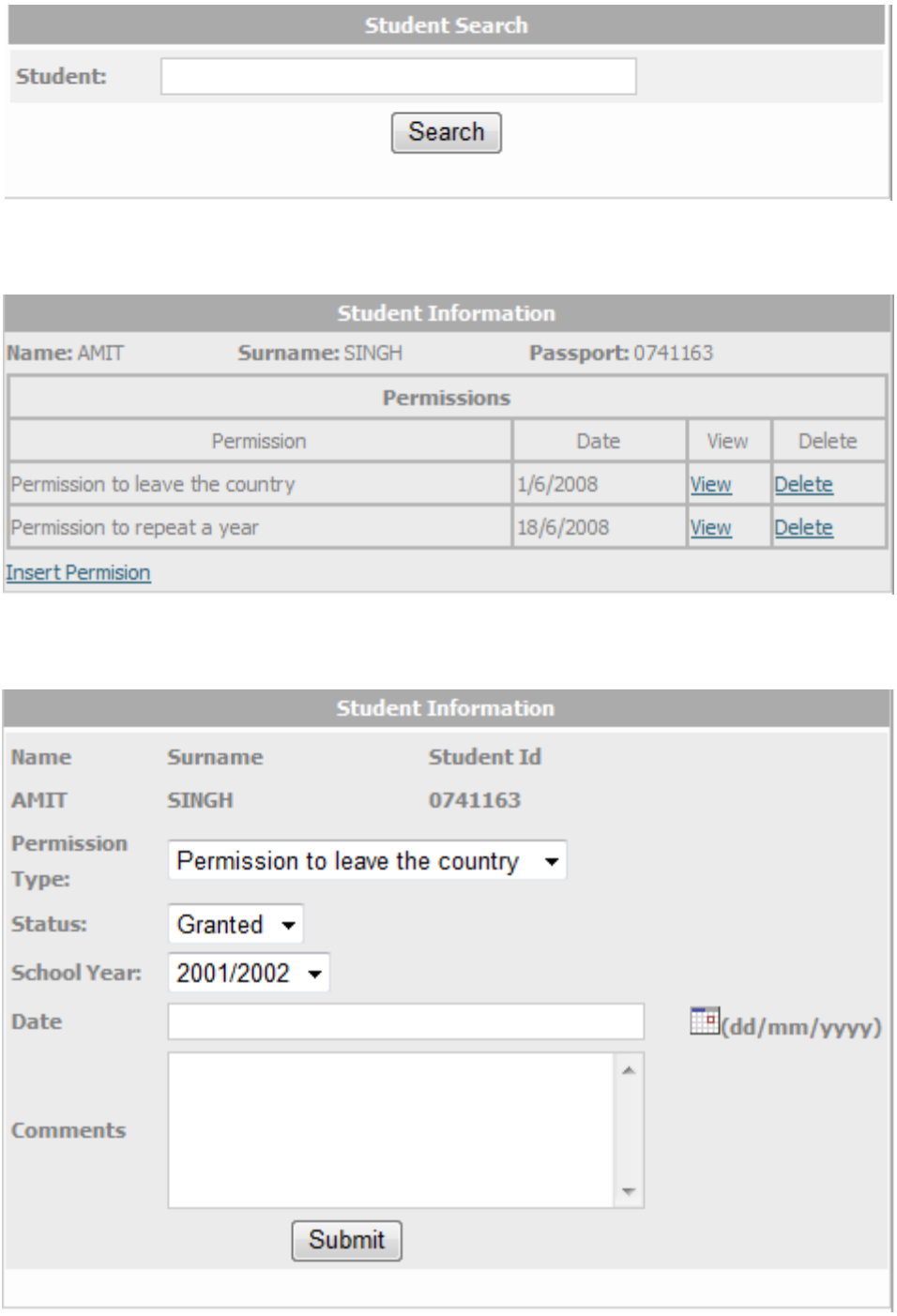
Anexo 3.5 Prototipo “Gestionar Estudiante”



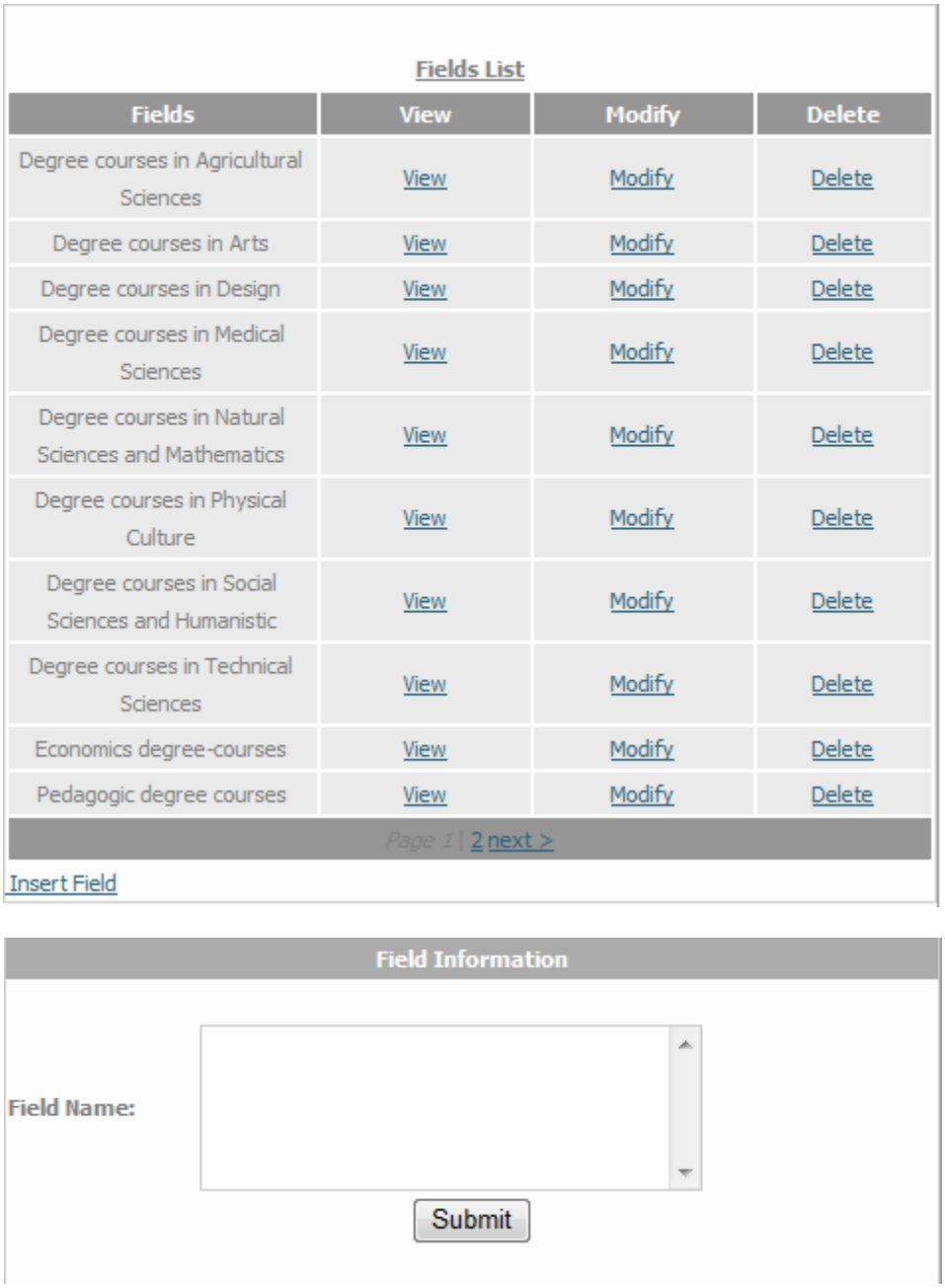
Anexo 3.5 Prototipo “Gestionar Estudiante”



Anexo 3.6 Gestionar Nota



Anexo 3.7 Prototipo “Gestionar Permiso”



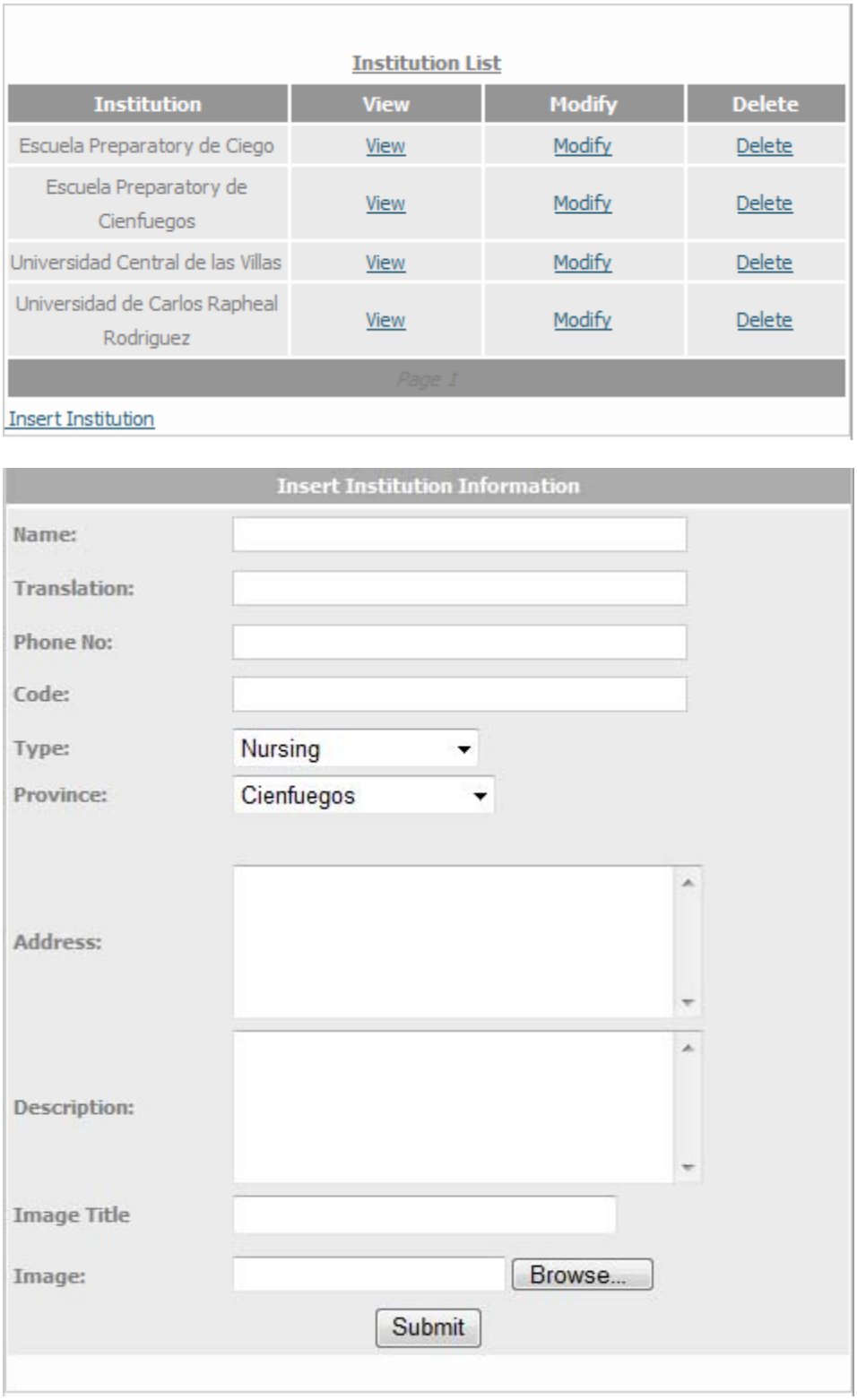
Fields List

Fields	View	Modify	Delete
Degree courses in Agricultural Sciences	View	Modify	Delete
Degree courses in Arts	View	Modify	Delete
Degree courses in Design	View	Modify	Delete
Degree courses in Medical Sciences	View	Modify	Delete
Degree courses in Natural Sciences and Mathematics	View	Modify	Delete
Degree courses in Physical Culture	View	Modify	Delete
Degree courses in Social Sciences and Humanistic	View	Modify	Delete
Degree courses in Technical Sciences	View	Modify	Delete
Economics degree-courses	View	Modify	Delete
Pedagogic degree courses	View	Modify	Delete

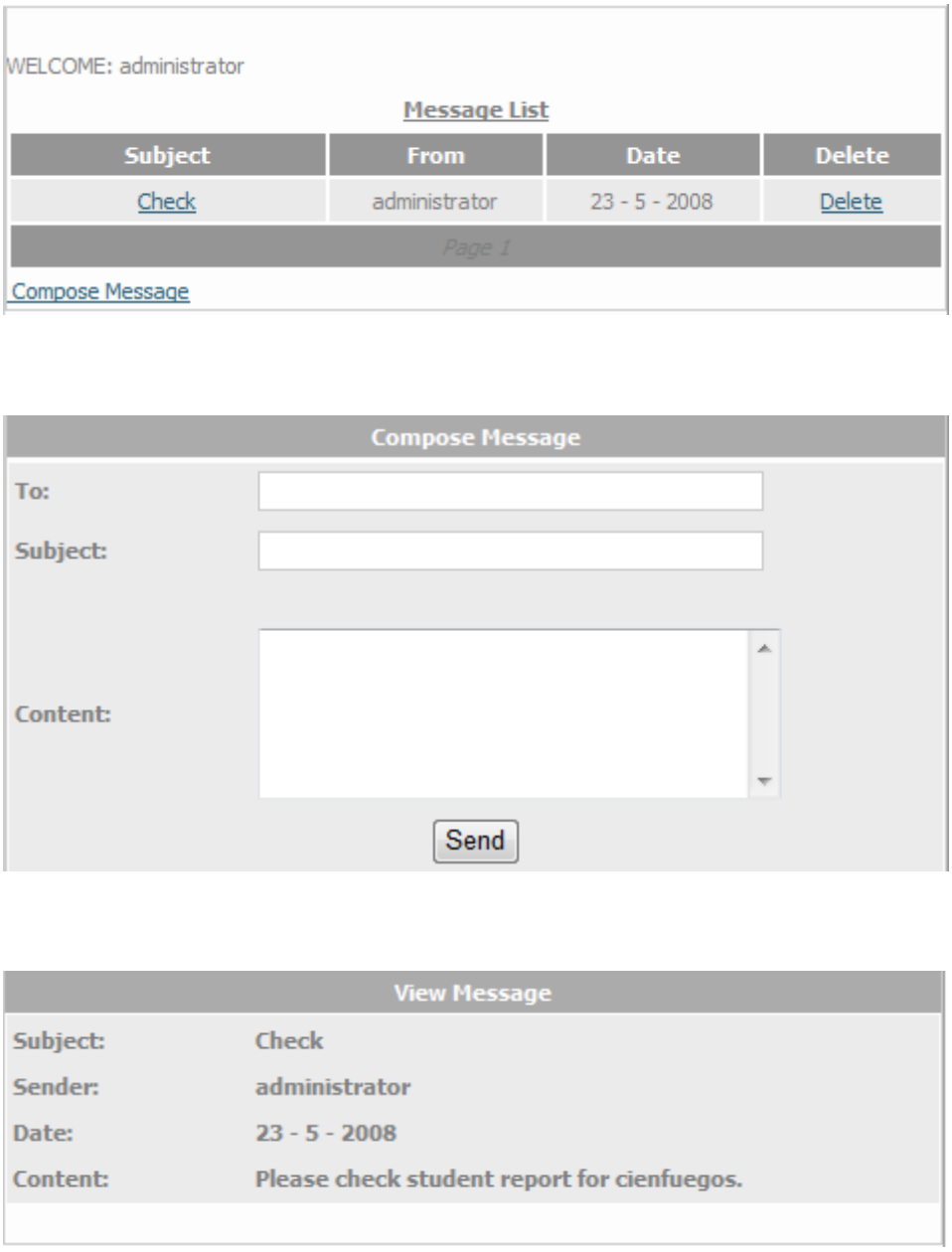
Page 1 | 2 next >

Insert Field

Anexo 3.8 Prototipo “Gestionar Campo de Estudio”

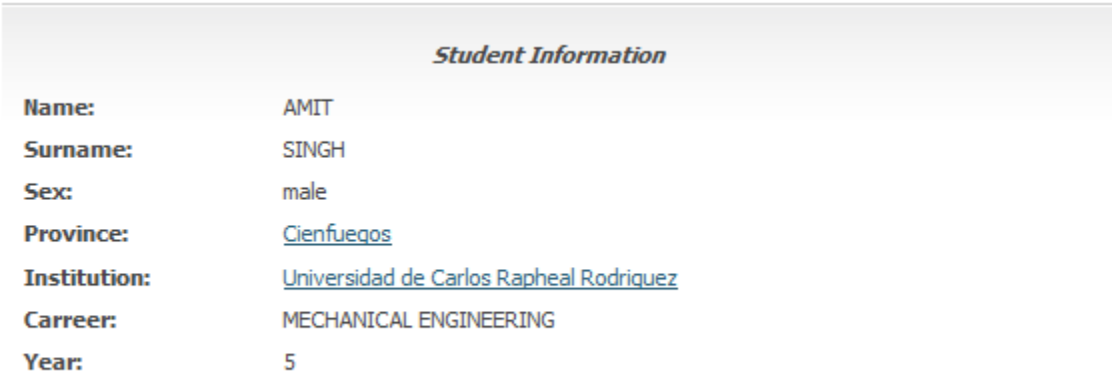


Anexo 3.9 Prototipo “Gestionar Institución”



Anexo 3.10 Prototipo “Usar Mensajería”

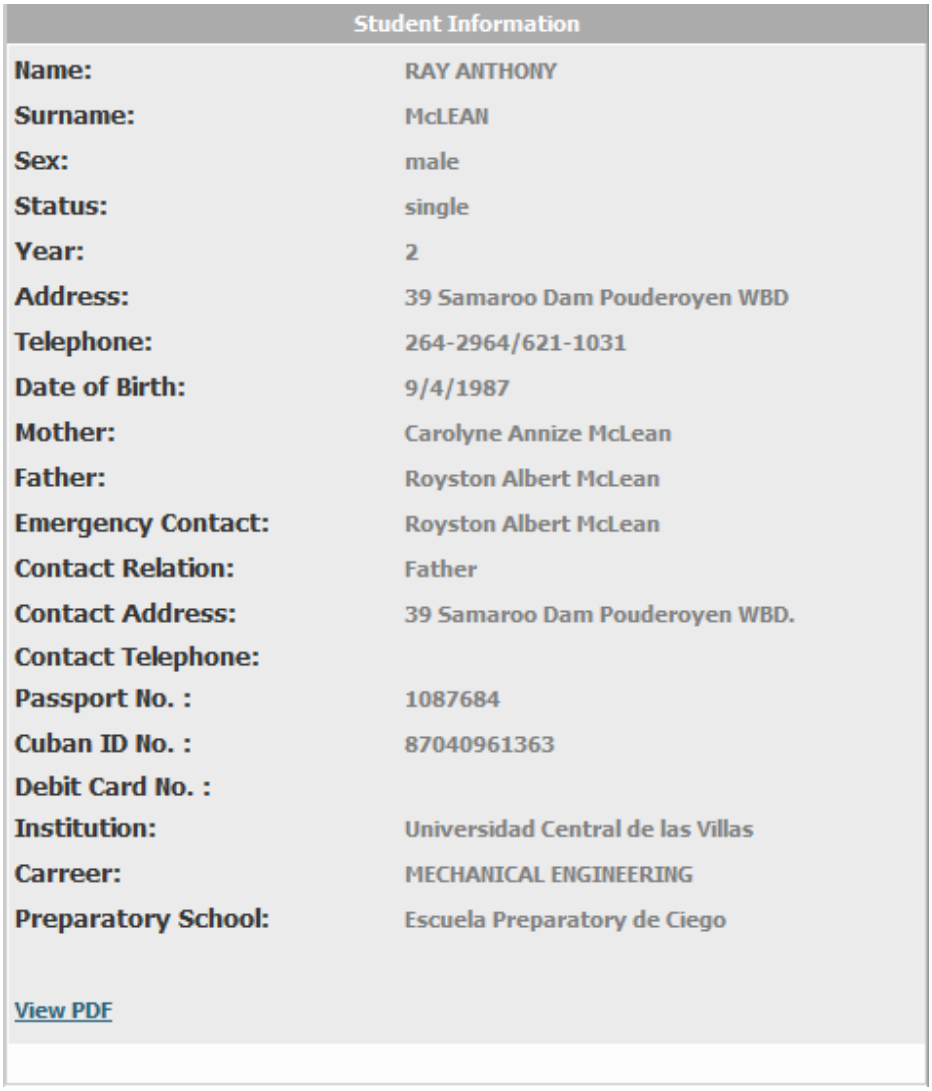
Anexo 3.11 Prototipo “Buscar Estudiante”



Student Information

Name:	AMIT
Surname:	SINGH
Sex:	male
Province:	Cienfuegos
Institution:	Universidad de Carlos Rapheal Rodriguez
Carreer:	MECHANICAL ENGINEERING
Year:	5

Anexo 3.12 Prototipo “Visualizar información general de estudiante”

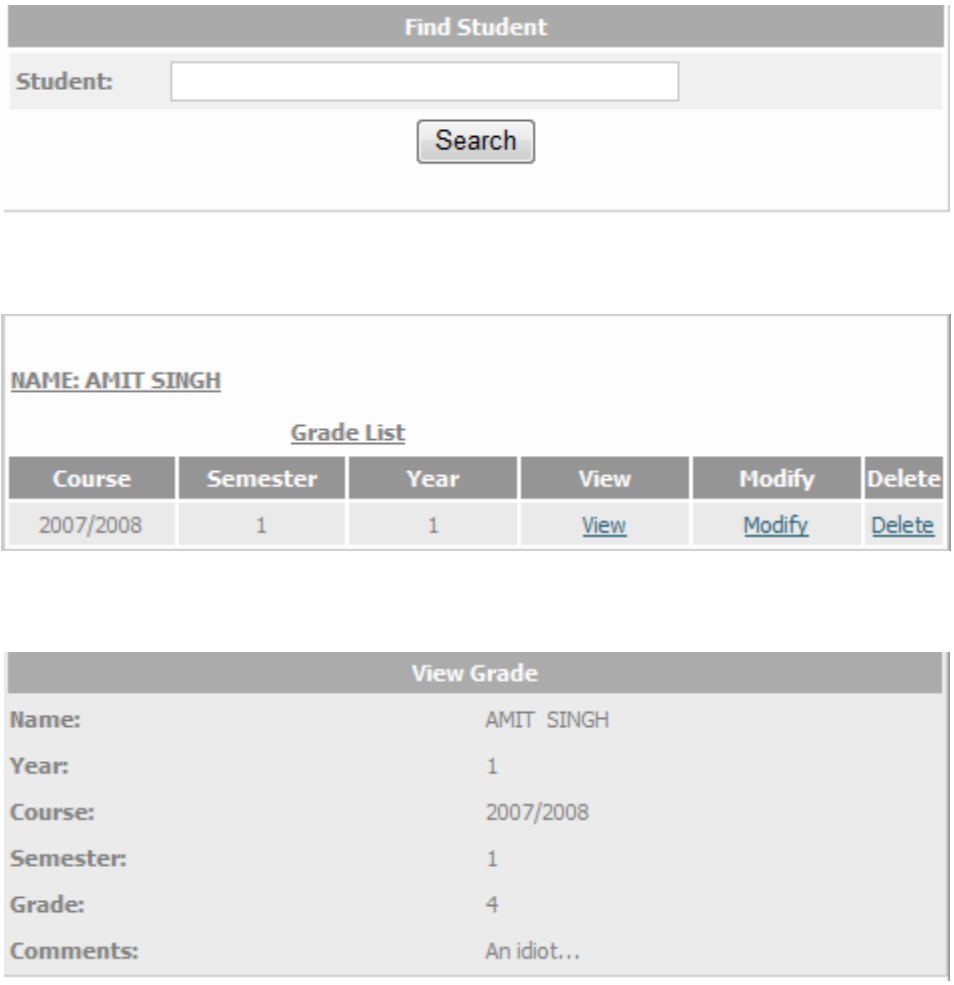


Student Information

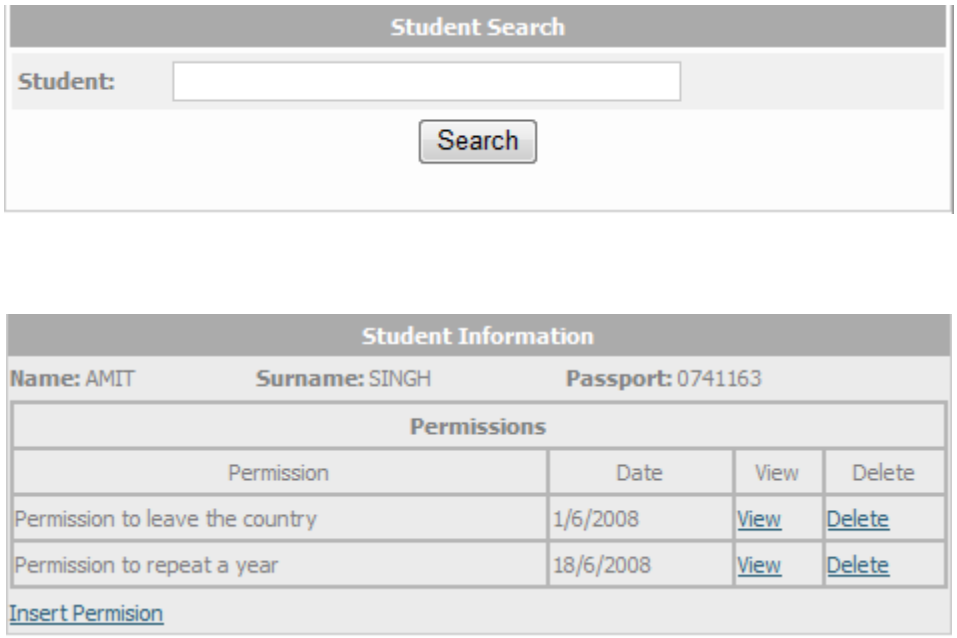
Name:	RAY ANTHONY
Surname:	McLEAN
Sex:	male
Status:	single
Year:	2
Address:	39 Samaroo Dam Pouderoyen WBD
Telephone:	264-2964/621-1031
Date of Birth:	9/4/1987
Mother:	Carolyne Annize McLean
Father:	Royston Albert McLean
Emergency Contact:	Royston Albert McLean
Contact Relation:	Father
Contact Address:	39 Samaroo Dam Pouderoyen WBD.
Contact Telephone:	
Passport No. :	1087684
Cuban ID No. :	87040961363
Debit Card No. :	
Institution:	Universidad Central de las Villas
Carreer:	MECHANICAL ENGINEERING
Preparatory School:	Escuela Preparatory de Ciego

View PDF

Anexo 3.13 Prototipo “Visualizar información particular del estudiante”



Anexo 3.14 Prototipo “Visualizar notas del estudiante”



Anexo 3.15 Prototipo “Visualizar permisos del estudiante”

Province List

Cienfuegos

Ciudad de la Habana

La Habana

Matanzas

Pinar del Rio

Sancti Spiritus

Villa Clara

Page 1

View Province Detail

Name: Matanzas

Code: MAT

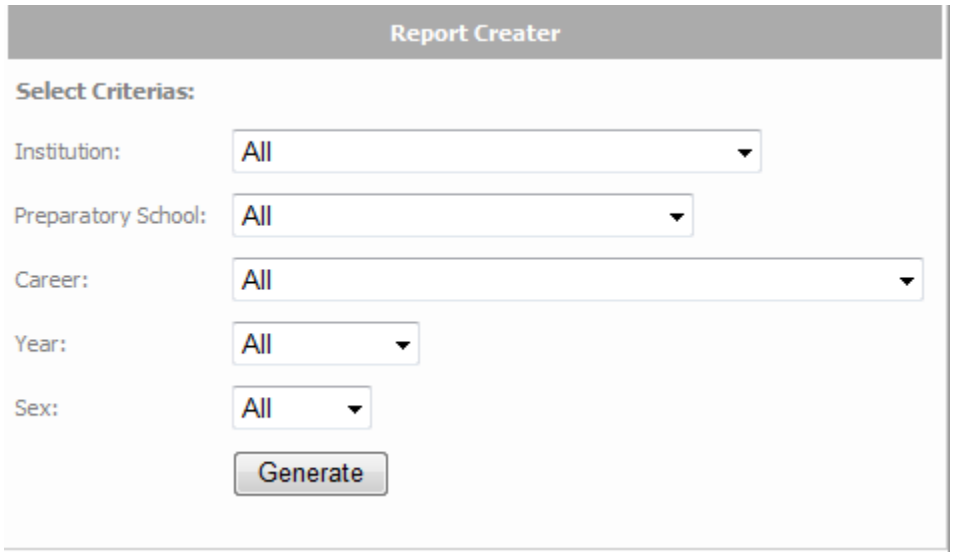
Institutions:

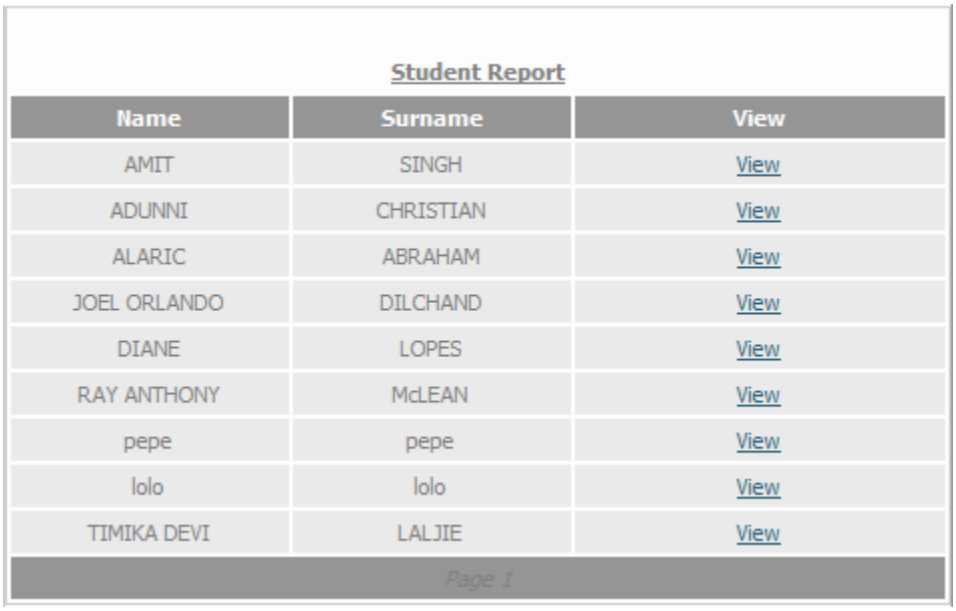
Description: Matanzas is one of the provinces of Cuba. Major towns in the province include Cárdenas, Jovellanos and the capital of the same name, Matanzas. The resort town of Varadero is also located in this province. Of Cuba's provinces, Matanzas is one of the most industrialised, with petroleum wells, refineries, supertanker facilities, and 21 sugar mills to process the harvests of the fields of sugarcane in the province.

Title: The Bridge of Bacunayagua marks the western boundary of the

Anexo 3.17 Prototipo “Visualizar provincia”

Anexo 3.18 Prototipo “Visualizar información general del sitio”



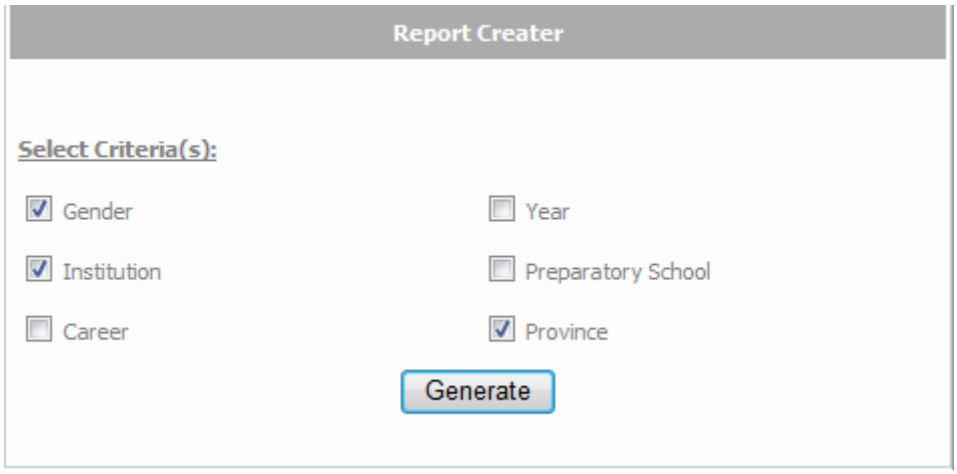


Student Report

Name	Surname	View
AMIT	SINGH	View
ADUNNI	CHRISTIAN	View
ALARIC	ABRAHAM	View
JOEL ORLANDO	DILCHAND	View
DIANE	LOPES	View
RAY ANTHONY	McLEAN	View
pepe	pepe	View
lolo	lolo	View
TIMIKA DEVI	LALJIE	View

Page 1

Anexo 3.19 Prototipo “Generar reporte de estudiantes agrupados por diferentes criterios”



Statistical Reports

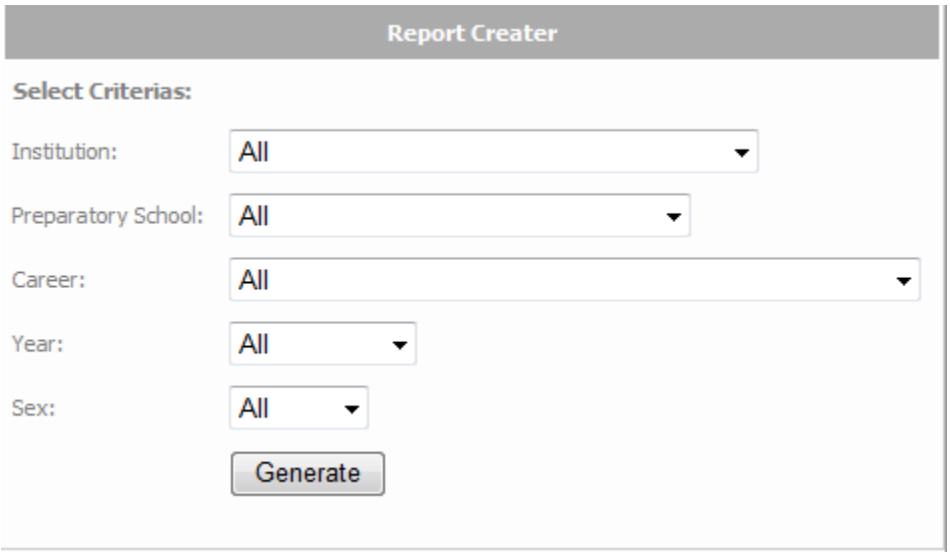
Total Students	9
Graduated Students:	0
Failed Students:	1
Expelled Students:	0
Preparatory Students:	0
Current Career Students:	8

Sex	Amount of Students
female	2
male	7

Provinces	Amount of Students
Cienfuegos	4
Villa Clara	5

Institutions	Amount of Students
Universidad Central de las Villas	5
Universidad de Carlos Rapheal Rodriguez	4

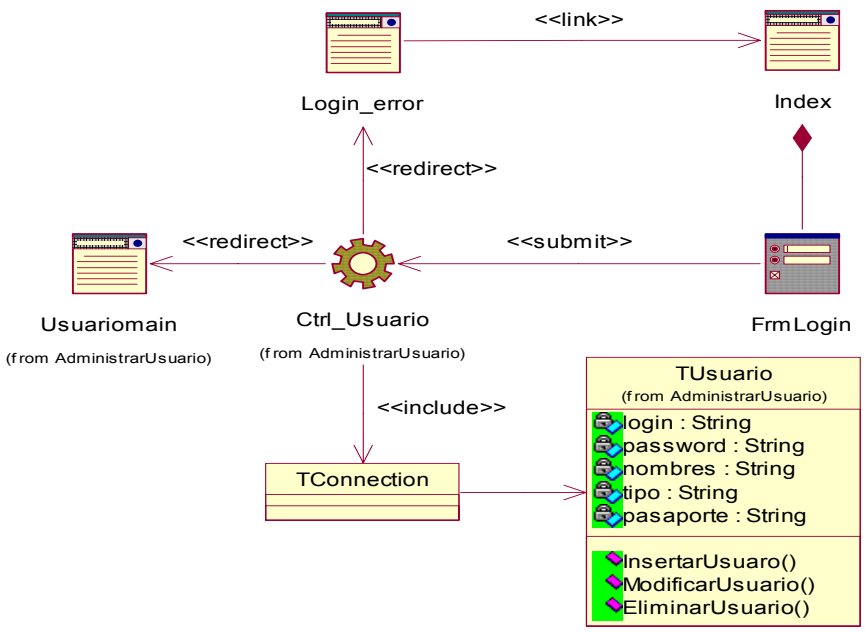
Anexo 3.20 Prototipo “Generar reporte estadístico”



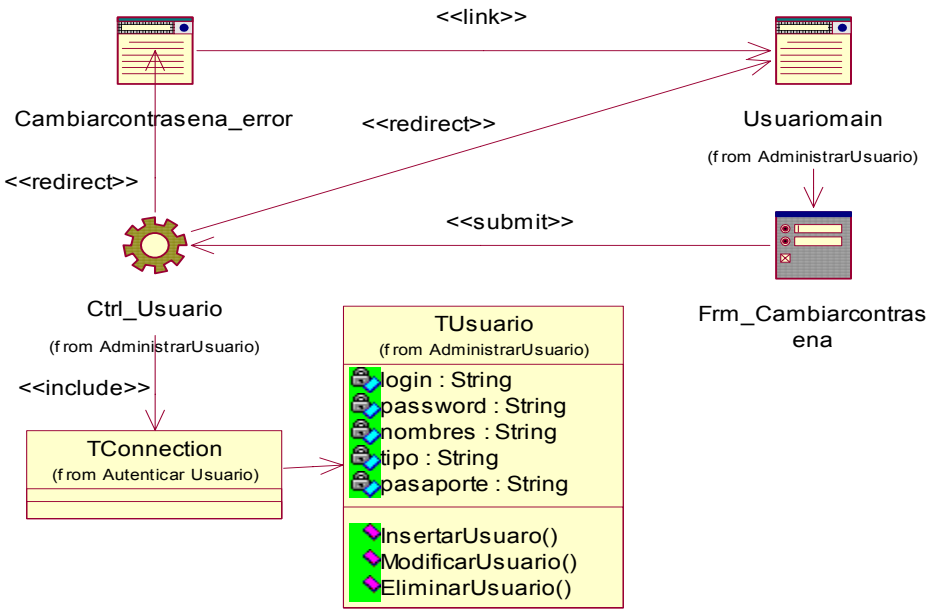
NAME	SURNAME
AMIT	SINGH
ADUNNI	CHRISTIAN
ALARIC	ABRAHAM
JOEL ORLANDO	DILCHAND
DIANE	LOPES
RAY ANTHONY	McLEAN
pepe	pepe
lolo	lolo
TIMIKA DEVI	LALJIE

Anexo 3.21 Prototipo “Imprimir reporte”

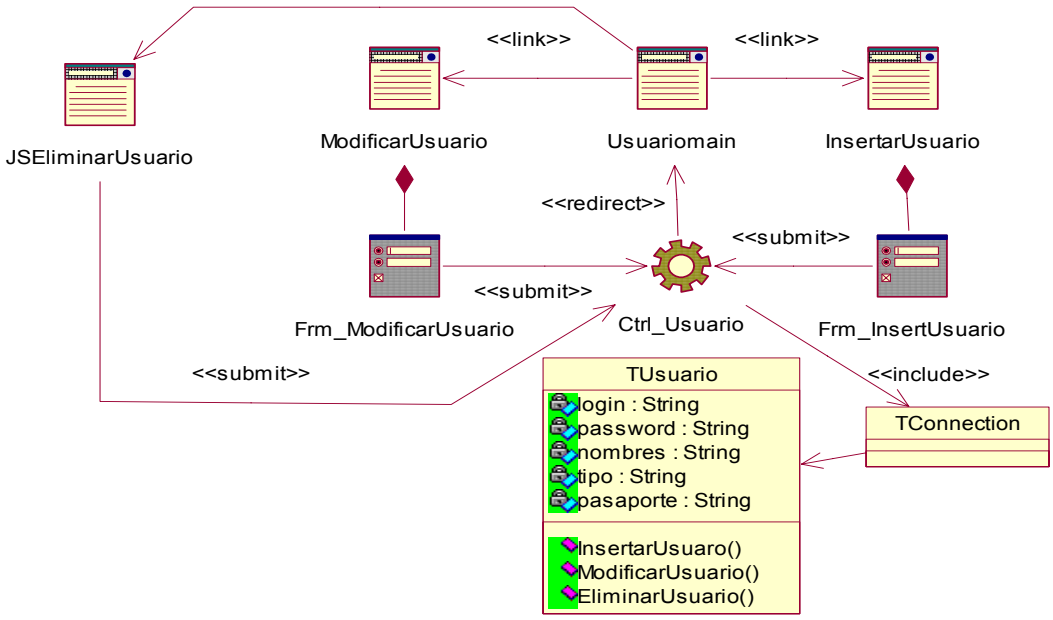
Diagramas de clases web



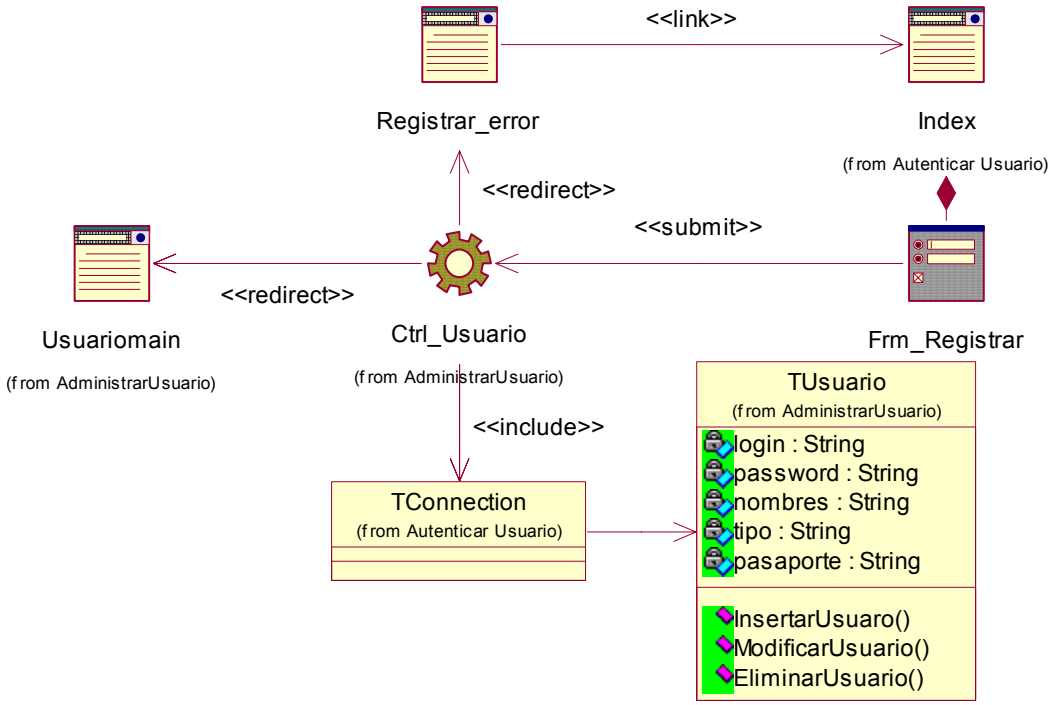
Anexo 4.1: Diagrama de clases Web "Autenticar Usuario"



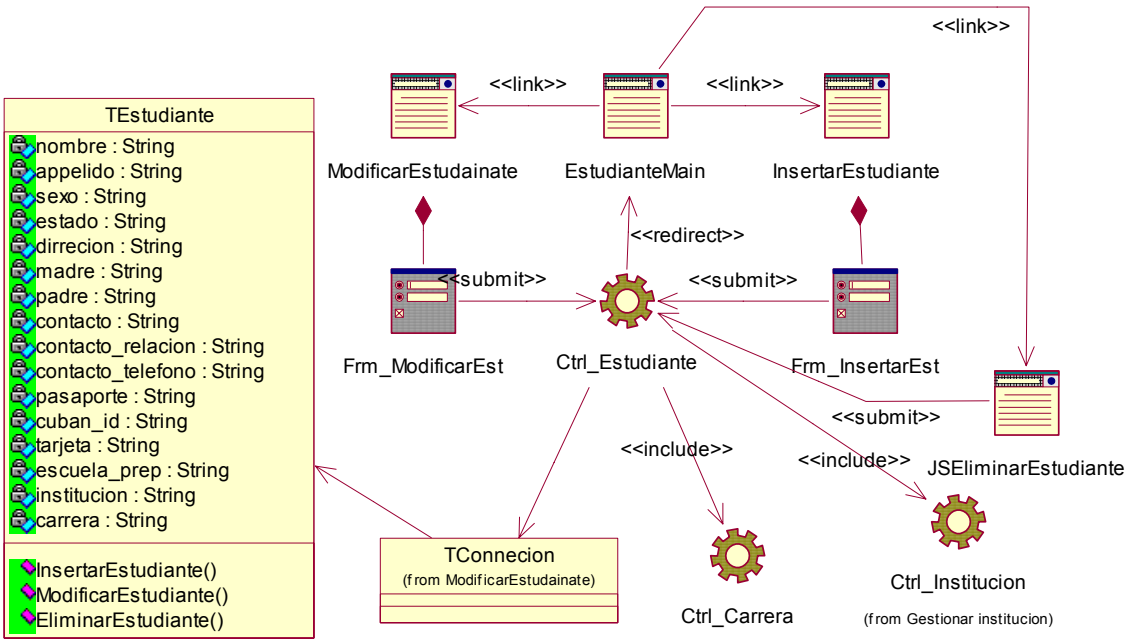
Anexo 4.2: Diagrama de clases Web "Cambiar Contraseña"



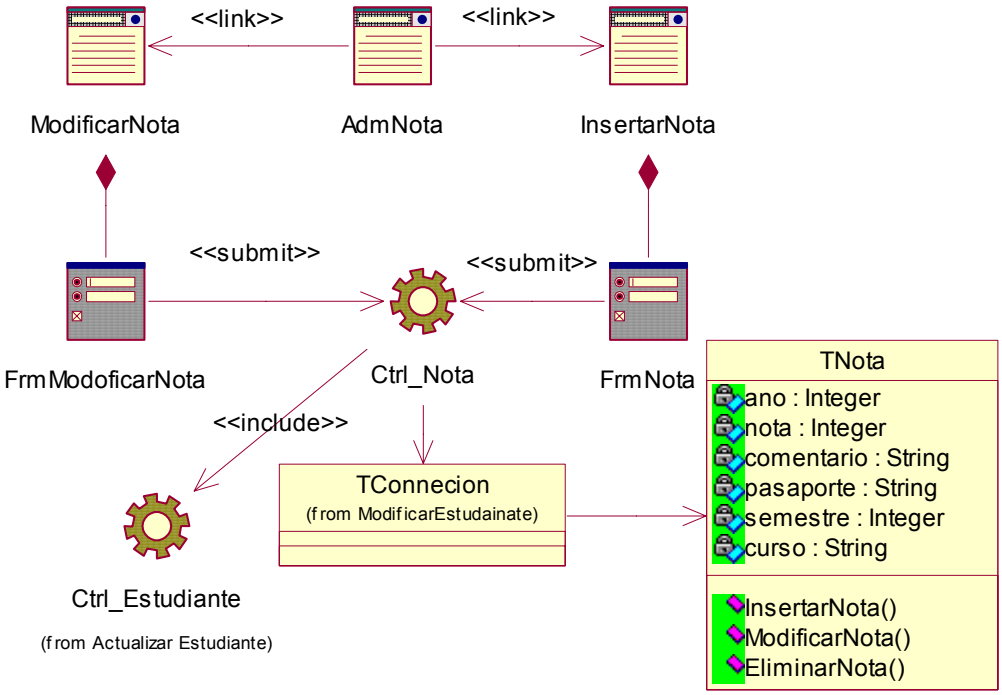
Anexo 4.3: Diagrama de clases Web "Administrar Usuario"



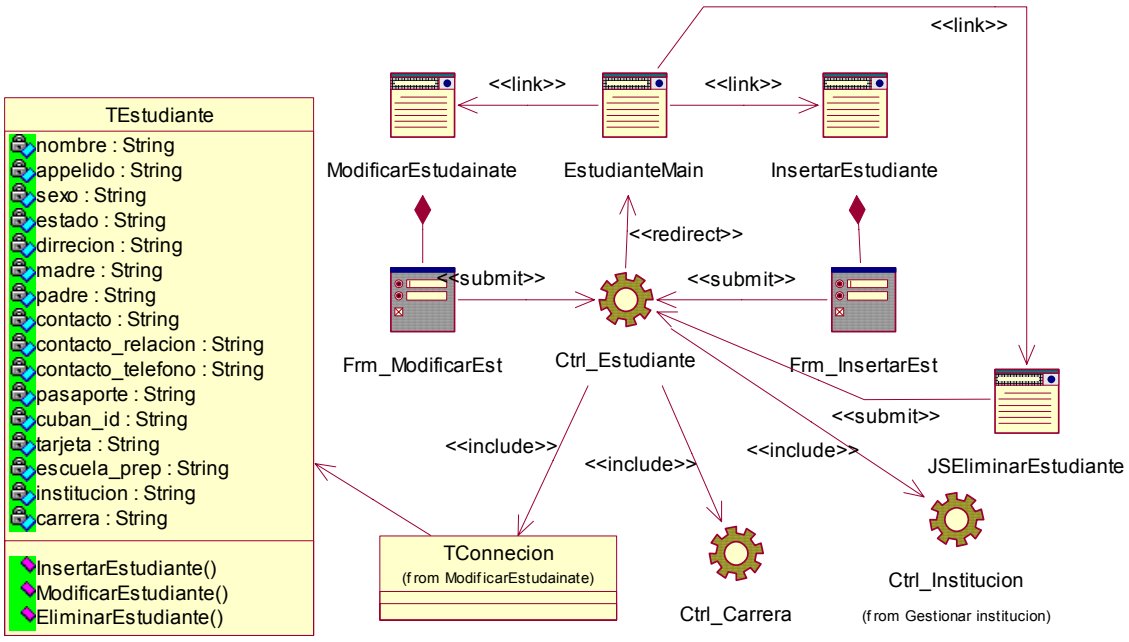
Anexo 4.4: Diagrama de clases Web "Registrar Usuario"



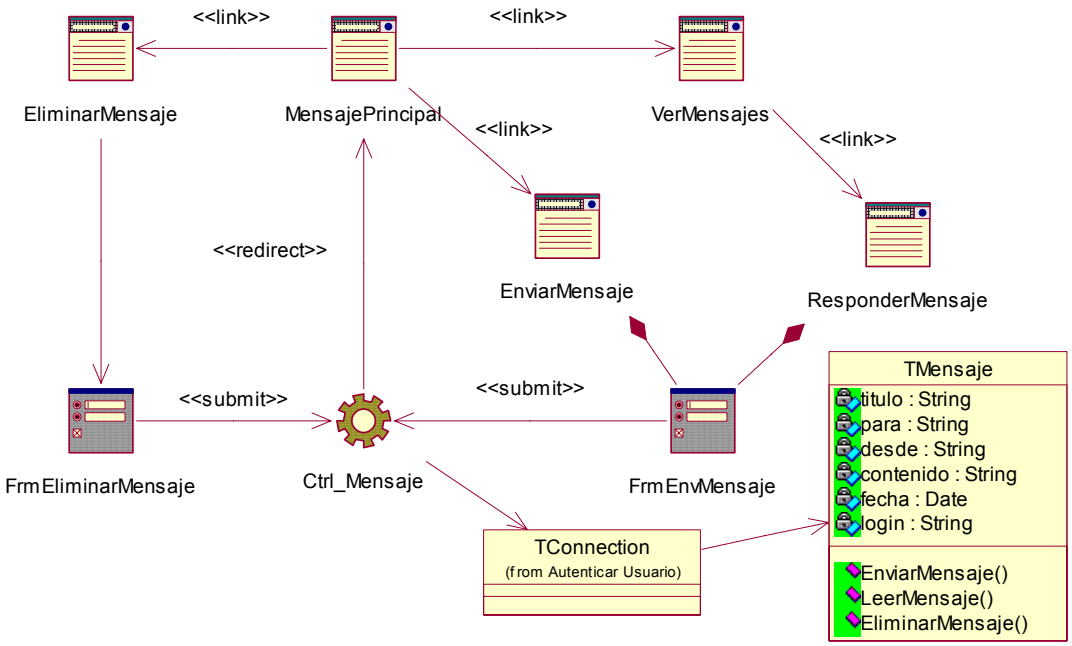
Anexo 4.5: Diagrama de clases Web "Gestionar Estudiante"



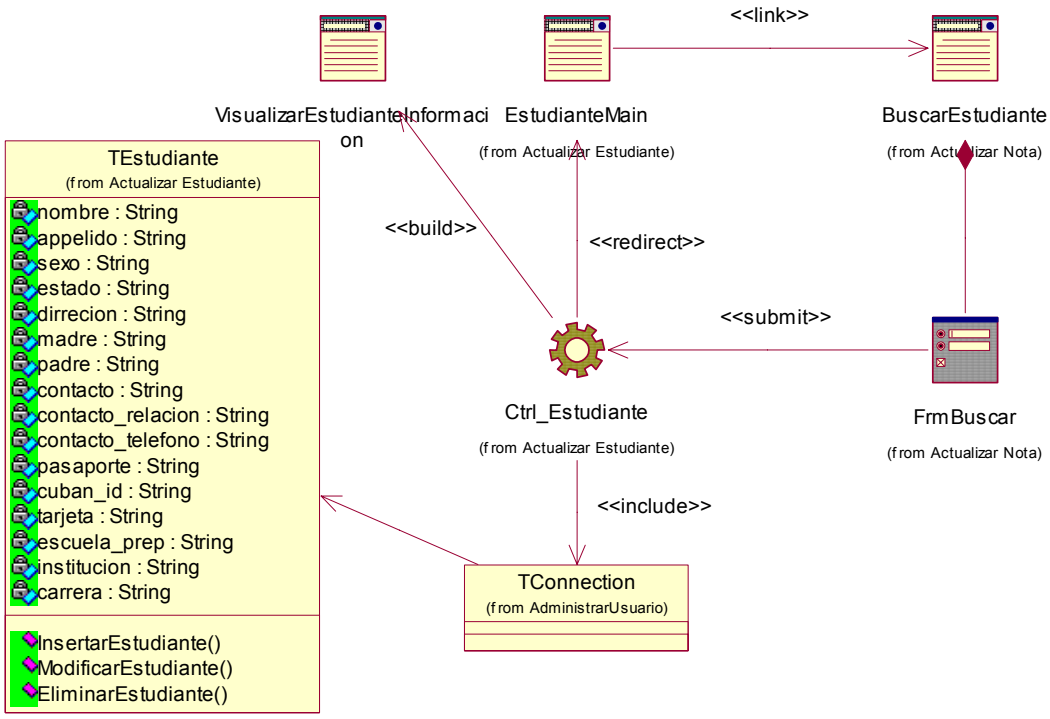
Anexo 4.6: Diagrama de clases Web "Gestionar Nota"



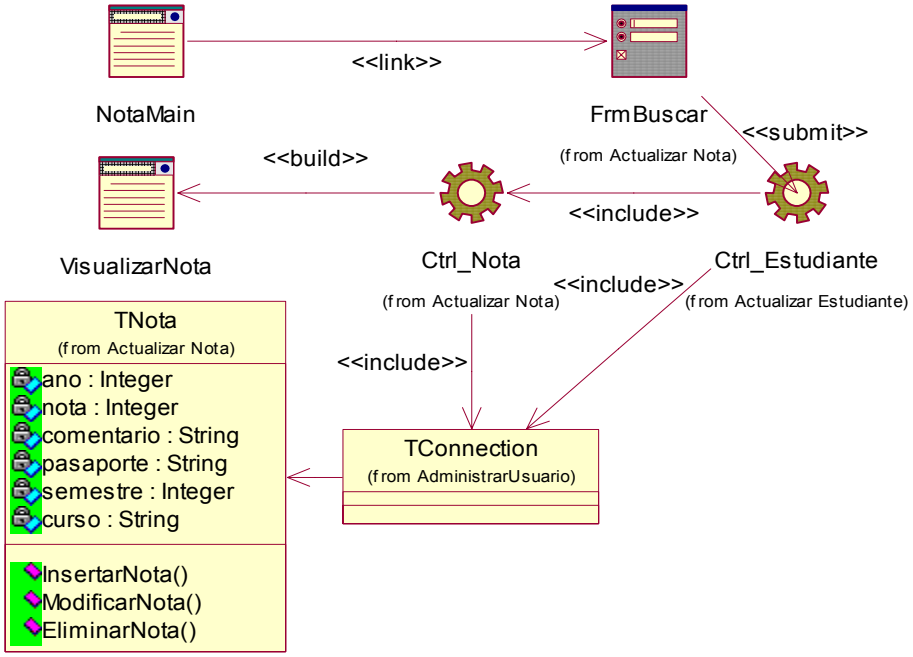
Anexo 4.8: Diagrama de clases Web “Gestionar Institución”



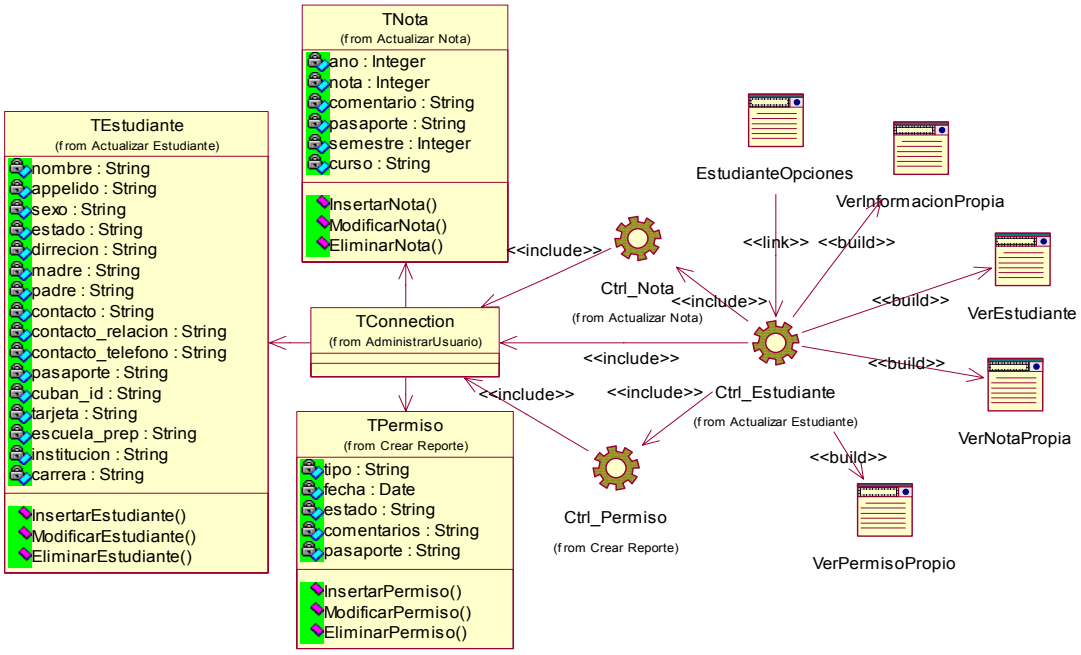
Anexo 4.9: Diagrama de clases Web "Usar Mensajería"



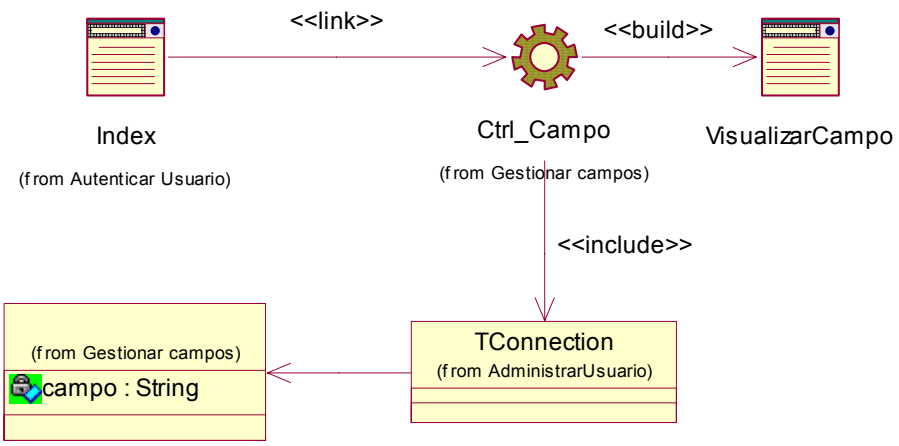
Anexo 4.10: Diagrama de clases Web "Buscar Estudiante"



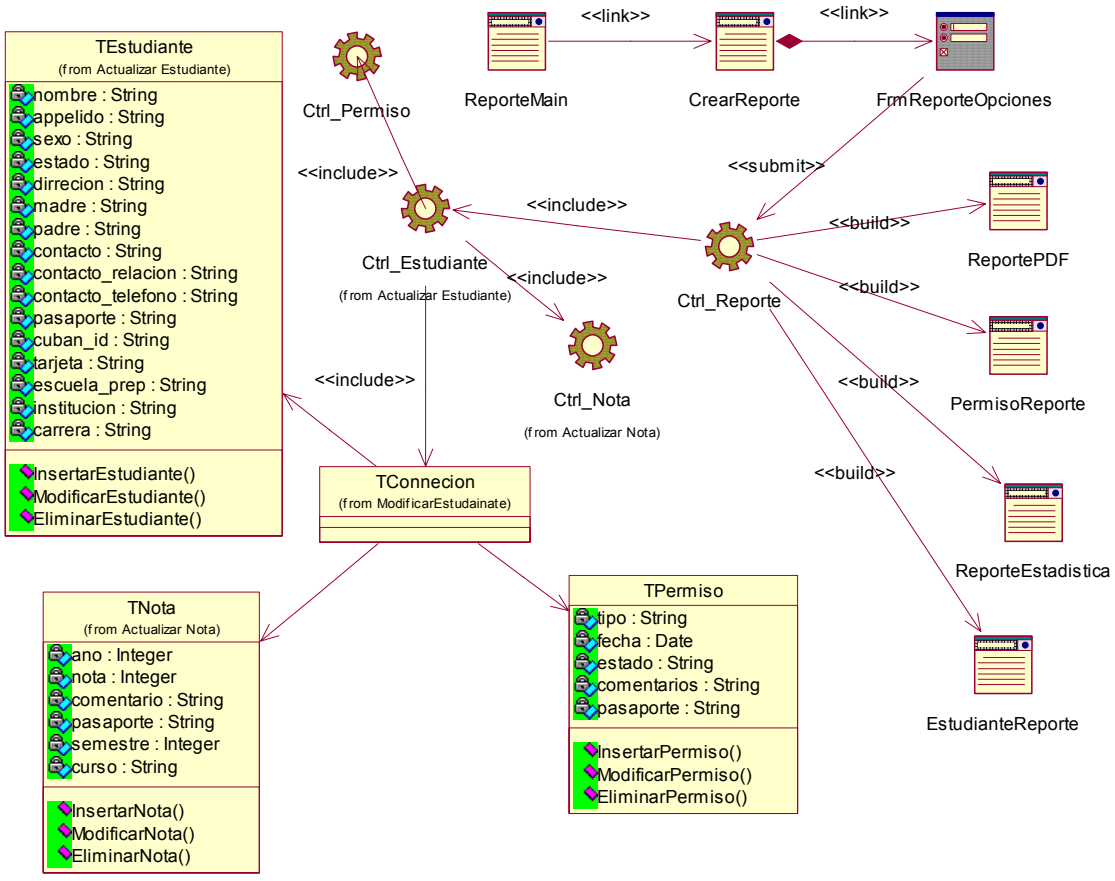
Anexo 4.11: Diagrama de clases Web "Visualizar Nota"



Anexo 4.12: Diagrama de clases Web "Visualizar Estudiante"

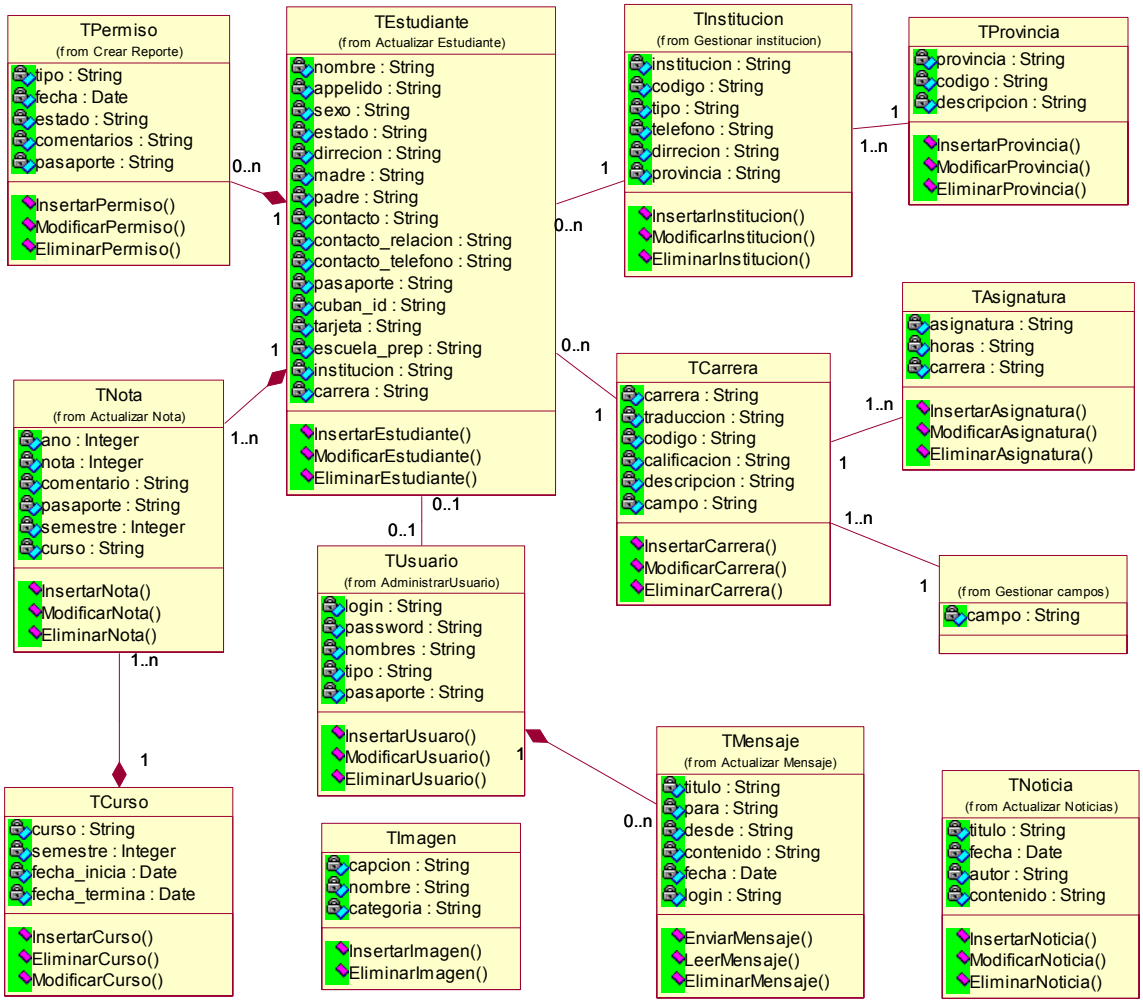


Anexo 4.13: Diagrama de clases Web "Visualizar Campo"



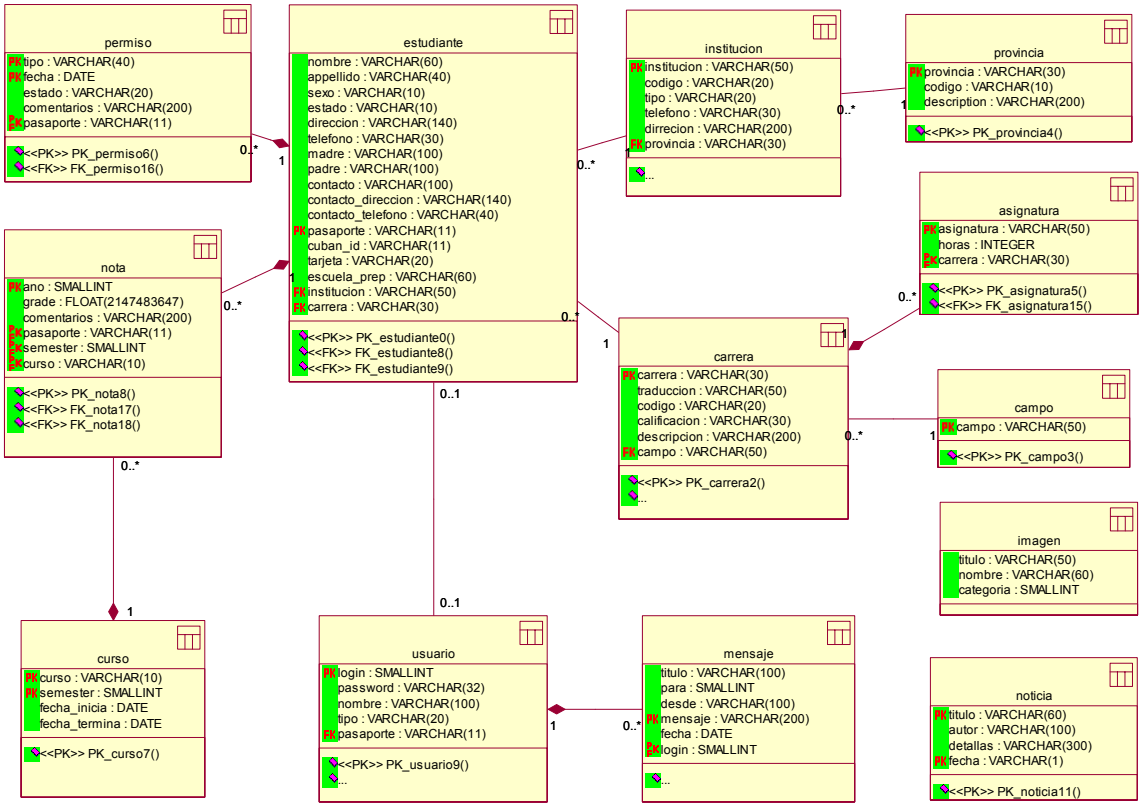
Anexo 4.14: Diagrama de clases Web "Generar Reporte"

Modelo lógico de datos



Anexo 5: Modelo lógico de datos

Modelo físico de datos



Anexo 6: Modelo físico de datos