

UNIVERSIDAD DE CIENCIAS INFORMÁTICAS

Facultad 2



*Sistema automatizado para la gestión de información de la secretaría docente.*

**Trabajo de diploma para optar por el título de Ingeniero Informático.**

**Autor:** Osleydis Velázquez Pérez

**Tutor:** Ing. Rodney del Valle Torres

**Ciudad de la Habana, Junio 2007**

*Agradecimientos*

*A la mujer más hermosa de este mundo, mi mamá, la principal causante que este momento se halla hecho realidad, por guiar cada uno de mis pasos.*

*A mi hermana por el cariño que siempre me ha tenido y que ha sido mi luz en cada sendero.*

*A mi novio por la paciencia con que ha sabido tratarme en cada momento que todo parecía desmoronarse, por el cariño que brota de sus palabras, por la dedicación de cada día.*

*A mi hermano Yunisel que en cada momento bueno o malo estuvo presente, por la forma en que siempre supo hacerme saber que donde quiera que este va a seguir siendo el mejor de los hermanos.*

*A mi familia que ha sabido estar en cada momento de debilidad para apoyarme.*

*A mis compañeros que durante todos los cursos han sido siempre buenos amigos.*

*A las personas más incondicionales que me han extendido su brazo y no por ser los últimos son menos reconocidos, al Ing. Ronny Zamora Aguilar, Ing. Alberto Tamayo Ramos, para los cuales no tengo palabras para agradecerles pero siempre les voy a reconocer todo lo que han hecho por hacer realidad este momento.*

*A todos muchas gracias.*

*Dedicatoria*

*A quien se merece todo mi respeto, admiración y sobre todo un cariño infinito que brota de mi corazón, a mi madre, que sus brazos siempre se abren cuando necesito un abrazo. Su corazón sabe comprender cuándo necesito una amiga. Sus ojos sensibles se endurecen cuando necesito una lección. Su fuerza y su amor me han dirigido por la vida y me han dado las alas que necesitaba para volar.*

**Declaración de autoría**

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Osleydis Velazquez Perez

Rodney del Valle Torres

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

## **Resumen**

Dada la importancia que posee el manejo de la información en la toma de decisiones en cualquier proceso de la vida, se han realizado varios intentos de automatizar el procesamiento de la información en las instituciones universitarias del país. Los objetivos de este trabajo son la formalización y creación de un sistema informativo automatizado que permita actualizar y mantener la información sobre estudiantes, desarrollar una herramienta de fácil manejo para el personal de secretaría docente de la facultad 2 de la Universidad de las Ciencias Informáticas y desarrollar un ambiente WEB que permita obtener resultados de consultas desde cualquier lugar de la red universitaria en cualquier momento de tiempo, con acceso para todas las personas que se determine. Actualmente en la secretaría docente de la facultad el control de la información relacionada con el proceso de petición de licencias especiales, licencias de matrículas y pases eventuales se hace de forma manual ocasionando demoras, pérdidas de la información, falta de eficiencia y gasto de recursos materiales. Se desea tener toda la información disponible, actualizada y organizada para agilizar el trabajo del personal de la secretaría para lo que es necesario implementar un software de gestión. Los software de gestión son todos aquellos programas creados para la gestión de grandes cantidades de documentos. Estos documentos no tienen una organización de sus contenidos, al contrario de lo que suele suceder con la información almacenada en una base de datos. Los documentos almacenados en una base de datos permite el acceso rápido mediante diversos métodos a la información.

Como conclusión se puede arribar que sin información y sin datos, es imposible garantizar la calidad o convertirse en una organización de excelencia.

## Índice

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1 Fundamentación Teórica .....</b>	<b>4</b>
<b>1.1 Introducción .....</b>	<b>4</b>
<b>1.2 Estado del arte .....</b>	<b>4</b>
<b>1.3 Tendencias y tecnologías actuales .....</b>	<b>6</b>
1.3.1 Programación Web .....	7
1.3.2 Modelo Cliente Servidor .....	9
1.3.3 Lenguajes de Programación .....	11
1.3.4 Servidores Web .....	11
1.3.5 Ajax .....	12
1.3.6 Patrones de Diseño .....	14
Patrones de Comportamiento .....	16
1.3.7 Modelo Vista Controlador .....	17
1.3.8 Sistema de Gestión de Base de Datos .....	18
1.3.9 Proceso de Desarrollo .....	22
1.3.10 Herramientas utilizadas .....	23
<b>1.4 Conclusiones .....</b>	<b>26</b>
<b>Capítulo 2 Características del Sistema .....</b>	<b>29</b>
<b>2.1 Introducción .....</b>	<b>29</b>
<b>2.2 Objeto de Estudio .....</b>	<b>30</b>
2.2.1 Problema y situación problemática .....	30
2.2.2 Objeto de automatización .....	30
2.2.3 Información que se maneja .....	31
2.2.4 Propuesta del sistema .....	31
<b>2.3 Modelo del negocio actual .....</b>	<b>32</b>
2.3.1 Reglas del negocio a considerar .....	32
2.3.2 Determinación y justificación de los actores y trabajadores del negocio .....	33
2.3.3 Diagrama de Casos de Uso del Negocio .....	35
2.3.4 Descripción Textual del los Casos de Uso del Negocio .....	35
2.3.5 Diagramas de Actividades .....	35
2.3.6 Modelo de objetos .....	36
<b>2.4 Relación de los Requerimientos .....</b>	<b>36</b>
2.4.1 Definición de los requisitos funcionales .....	36
<b>R 1- Gestionar pases eventuales .....</b>	<b>36</b>
<b>R 1.1- Insertar estudiantes .....</b>	<b>37</b>
<b>R 1.2- Eliminar estudiantes .....</b>	<b>37</b>

<b>R 1.3- Modificar estudiantes</b> .....	<b>37</b>
2.4.2 Definición de los requisitos no funcionales.....	37
<b>2.5 Modelo de Casos de Uso del Sistema</b> .....	<b>39</b>
2.5.1 Definición de los actores del sistema a automatizar.....	40
2.5.2 Diagrama de casos de uso del sistema a automatizar. ....	40
2.5.3 Descripción de los casos de uso del sistema.....	41
<b>2.6 Conclusiones</b> .....	<b>42</b>
<b>Capítulo 3 Análisis y diseño del sistema</b> .....	<b>61</b>
<b>3.1 Introducción</b> .....	<b>61</b>
<b>3.2 Análisis</b> .....	<b>61</b>
3.2.1 Modelo de clases de análisis.....	61
3.2.2 Caso de uso Gestionar Pases Eventuales.....	61
3.2.3 Caso de uso Gestionar Licencias de Matrículas.....	62
3.2.4 Caso de uso Gestionar Licencias Especiales .....	63
3.2.5 Caso de uso Autenticarse .....	63
3.2.6 Caso de uso Administración.....	64
3.2.7 Clases Bases.....	64
Tabla 22. Diagrama de clases bases.....	64
3.2.7.1 Descripción de las clases.....	64
<b>3.3 Diseño</b> .....	<b>69</b>
3.3.1 Diagramas de interacción.....	69
3.3.2 Diagrama de clases del diseño.....	69
3.3.2.1 Caso de uso Gestionar Pases Eventuales.....	70
3.3.2.2 Caso de uso Gestionar Licencias de Matrículas.....	71
Tabla 43. Caso de uso “Gestionar Licencias de Matrículas” .....	72
3.3.2.3 Caso de uso Gestionar Licencias Especiales .....	73
Tabla 44. Caso de uso “Gestionar Licencias Especiales” .....	73
3.3.2.4 Caso de uso Autenticarse .....	74
3.3.2.5 Caso de uso Administración .....	75
<b>3.4 Diseño de la base de datos</b> .....	<b>75</b>
3.4.1 Modelo lógico de datos (diagrama de clases persistentes) .....	76
3.4.2 Modelo físico de datos (modelo de datos).....	76
<b>3.5 Conclusiones</b> .....	<b>78</b>
<b>Capítulo 4 Implementación</b> .....	<b>62</b>
<b>4.1 Introducción</b> .....	<b>62</b>
<b>4.2 Diagrama de despliegue</b> .....	<b>62</b>
<b>4.3 Diagrama de componentes</b> .....	<b>63</b>
<b>4.4 Diagrama de Componentes de BD</b> .....	<b>66</b>

<b>Capítulo 5 Estudio de Factibilidad.....</b>	<b>66</b>
5.1 Introducción .....	66
5.2 Planificación basada en casos de uso.....	66
5.3 Beneficios tangibles e intangibles.....	72
5.4 Análisis de costos y beneficios. ....	72
5.5 Conclusiones .....	73
<b>Conclusiones.....</b>	<b>74</b>
<b>Recomendaciones.....</b>	<b>75</b>
<b>Referencias Bibliográficas .....</b>	<b>76</b>
<b>Anexos .....</b>	<b>77</b>
Anexo 1.....	77
Anexo 2.....	81
Anexo3.....	83
Anexo 4.....	84
Anexo 5.....	95
<b>Glosario de Términos.....</b>	<b>107</b>



## **Introducción**

En la secretaría docente de la Facultad 2, se maneja diariamente mucho volumen de información acerca de los estudiantes. Se han creado aplicaciones que facilitan el trabajo de los trabajadores, por ejemplo Akademos donde se ha clasificado y organizado parte de la información que se procesa en la secretaría docente. También se ha aprovechado las posibilidades que brinda la tecnología para un mayor control de la información como Microsoft Word y Excel. Pero la mayoría del trabajo relacionado con los pases eventuales, licencias especiales y licencias de matrículas se hace manualmente provocando lentitud, difícil acceso e ineficiencias en el trabajo.

Por todo lo antes expuesto a la secretaría docente de la Facultad 2 le sería de gran utilidad un sistema donde esté la información clasificada y organizada para una mayor rapidez y eficiencia en el trabajo.

Pero, ¿Cómo lograr solucionar las dificultades que existen en la secretaría docente de la Facultad 2?

Para poder comprender todos los aspectos necesarios para darle solución a este problema, nos enfocamos (objeto de estudio) específicamente en automatizar la gestión de la información asociada a la petición de pases eventuales, licencias especiales y licencias de matrículas.

Luego de analizar el problema se llegó a que el objetivo del mismo es desarrollar un módulo para un sistema automatizado que permita al personal de la secretaría docente de la Facultad 2, mantener actualizada la información asociada a la petición de pases eventuales, licencias especiales y licencias de matrículas.

Para el cumplimiento del objetivo se trazaron tareas como:

1. Investigación del problema y su situación actual.
2. Proposición y acuerdo de las características de la arquitectura de la aplicación.
3. Desarrollo del modelo del problema real siguiendo la metodología de RUP.
4. Proposición y acuerdo de un diseño preliminar.
5. Implementación de un módulo que formará parte del sistema final.

Con todo esto se espera llegar a aumentar la eficiencia del trabajo en la secretaría docente de la Facultad 2, rapidez y facilidad de acceso a la información.

La estructura del documento quedó de la siguiente forma:

Capítulo 1: Fundamentación teórica: Estudio de los lenguajes de programación, gestores de bases de datos y herramientas utilizadas.

Capítulo 2: Características del sistema: Identificar las necesidades del cliente. Descripción del funcionamiento del sistema.

Capítulo 3: Análisis y diseño: Se describen las etapas de análisis y diseño. Se definen las funcionalidades y los requisitos no funcionales del sistema.

Capítulo 4: Implementación y prueba: Explicación de las funcionalidades definidas anteriormente.

Capítulo 5: Estudio de Factibilidad. Análisis de las variables tiempo, costo y esfuerzo. Enuncia los beneficios tangibles e intangibles del sistema.

## **Capítulo 1 Fundamentación Teórica**

### **1.1 Introducción**

La solicitud de licencias especiales, licencias de matrículas y pases eventuales es un proceso que se realiza actualmente en la secretaría docente de la Facultad 2. Estas solicitudes poseen un enorme flujo de información constante por lo que en este capítulo se abordarán los conceptos referentes al tema así como las tendencias que permiten crear un diseño Web exitoso.

### **1.2 Estado del arte**

Dada la importancia que posee el manejo de la información en la toma de decisiones en cualquier proceso de la vida, se han realizado varios intentos de automatizar el procesamiento de la información en las instituciones universitarias del país, ejemplo de ello lo es el Sistema Estudiante y otros sistemas que se han utilizado en los Centros de Educación Superior (CES) cubanos, los que han perdido actualidad con las nuevas tecnologías de la información y las comunicaciones.

En Cuba no existe un sistema dedicado al procesamiento de la información académica directamente, de forma tal que facilite a los diferentes tipos de usuarios obtener información y a los involucrados (dirigentes docentes, profesores, personal de apoyo y los propios estudiantes) actualizar las informaciones correspondientes según los niveles de protección que se determinen.

#### **Gestacad**

Sistema para la gestión académica (Universidad Matanzas “Camilo Cienfuegos”. Facultad De Informática, CUBA)

Brinda servicios como:

1. Búsqueda de un alumno.[Brinda la ubicación según el horario docente]
2. Listado de estudiantes por grupo

3. Reportes dinámicos de la información existente. [Se le da la posibilidad al usuario de seleccionar los campos de datos que desea obtener en el reporte así como el título de este y las condiciones que debe cumplir la información a mostrar].
4. Reporte de notas por asignatura y grupo[examen final, extraordinario, especial, premio]
5. Tabla con los resultados docentes de un grupo en un semestre.
6. Reporte de los resultados académicos de un estudiante en toda su carrera. [ Hoja de Rendimiento]
7. Actas de exámenes de las diferentes asignaturas.
8. Registro de características de un grupo de estudiantes.
9. Dar baja a un estudiante

El sistema consta de:

1. Un Módulo de Administración para la gestión de las tablas del sistema vía Web así como agregar nuevas consultas al sitio oficial y establecer los distintos niveles de acceso a estas.
2. Un Módulo Web para las Secretarías Docentes para la Gestión de Estudiantes que permite hasta el momento la realización de acciones generales comunes en una Secretaría Docente así como la obtención de reportes oficiales
3. Un Módulo Web para los Jefes de Departamentos docentes donde se incluyen acciones relativas como la asignación de la carga docente y el control sobre los profesores del Dpto.
4. Un Módulo Web para la Gestión de la Matrícula.
5. Un Módulo Web para los Profesores donde estos pueden llevar el control docente de sus estudiantes, el control de las evaluaciones así como reportes relativos a su carga docente.
6. Un sitio Web con reportes en línea con la utilidad del registro docente para los profesores además de la búsqueda de estudiantes la cual devuelve, además de algunos datos personales del estudiante, su ubicación según el horario docente detallando aula, asignatura y tipo de clases que está recibiendo además de su estado si se ha pasado asistencia en el turno de clase.

## **Akademios**

En la actualidad este sistema es capaz de realizar diversas acciones y brindar numerosos reportes como son:

Matrícula: Gestión de estudiantes y estructuras.

Estudiante: Herramienta que mantiene informado a los estudiantes sobre su desempeño académico.

Registro: Control del desarrollo del proceso docente, notas y asistencia.

Profesores: Gestión del claustro del centro.

Plan de Estudio: Definición de los planes de estudio. Asignaturas, disciplinas, perfiles.

Reportes: Diseño y generación de reportes personalizados.

Como se ha explicado hasta aquí el sistema *Akademios* es una acercamiento a la solución de un problema que atañe directamente a la educación superior cubana y que necesita una necesaria y rápida solución debido a la calidad y necesaria competitividad, prestigio y eficiencia de nuestra educación.

Después de analizar estos dos importantes sitios web se ha pensado en reunir las mejores características de ambos para implementar una interfaz sencilla que cumpla con estas condiciones para el desarrollo del sistema automatizado para la gestión de información de la secretaría docente de la facultad 2 de la Universidad de las Ciencias Informáticas (SGISD).

### **1.3 Tendencias y tecnologías actuales**

En las últimas décadas, y particularmente en la actualidad, los avances tecnológicos alcanzados en los campos de la informática y las comunicaciones, han despertado un entusiasmo inusitado y un afán por obtener provecho de aquellos, en todos los sectores del que hacer humano.

Poner en duda la utilidad de la tecnología informática en su estado actual, sería un absurdo, como lo sería también desconocer su capacidad de transformación de la realidad social, económica y cultural.

### **1.3.1 Programación Web**

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial.

Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL (Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una pagina Web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una pagina Web por toda la red de Internet, como es el HTTP (Hypertext Transfer Protocol).

Y por ultimo el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una pagina Web, este lenguaje es el HTML (Hypertext Markup Language).

En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una pagina Web.

Con el comienzo de Internet y la programación Web, se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación Web se orientaba a un diseño muy cargado y interactuando con el usuario, mientras que al empezar a competir con millones de Web se ha optado más por el diseño sencillo y de fácil comprensión.

En programación Web se creó la necesidad de conocer a fondo diferentes lenguajes de programación como HTML, JavaScript y DHTML. Con esto se creó un nuevo profesional de la informática, el diseñador Web, que viene siendo algo así como un experto en programación Web, a caballo entre el diseñador gráfico tradicional y el programador de aplicaciones llevadas a Internet.

### Alternativas de diseño Web

1. En la creación del diseño Web se debe hacer una especificación del trabajo a realizar teniendo en cuenta los servicios que brindará el sitio Web y el nivel de interactividad con sus usuarios:
2. Contenidos: debe definirse la cantidad de páginas/secciones y los nombres de las mismas, también la cantidad de imágenes a incluir en el sitio, datos tabulados (especificaciones técnicas, etc)
3. Formularios: pueden utilizarse formularios de consultas simples, de reservas, de solicitud de productos o servicios determinados, formularios de encuestas, etc
4. Programación a medida: esto queda determinado por necesidades muy específicas de programas que deban cumplir determinadas tareas en la gestión de información, cuando no existen aplicaciones prearmadas que puedan ser adaptadas a dichas necesidades particulares.
5. Aplicaciones dinámicas: ideales para otorgar interactividad con los usuarios. Ejemplos de estas son: encuestas y votaciones, sistemas de FAQs, foros de soporte, libros de visitas, newsletters, autorrespondedores inteligentes, etc.

En la etapa de Creación Web que está dedicada al armado de las páginas Web, creación del mapa de sitio, pruebas, y subida al servidor de hosting. Es importante tener en cuenta aspectos como:

Tamaño máximo del archivo que puedes subir.

1. El tiempo que tienen alojado tu archivo antes de borrarlo.
2. Indicar la velocidad de descarga.
3. Si existe alguna restricción al número de veces que puede descargarse un archivo.
4. Si la publicidad es muy poca, soportable o llega a complicar el proceso.
5. Si tienes que dar varios pasos para obtener el link de descarga, si además hay un tiempo de espera para descargar y en general la simplicidad del proceso subir/descargar.
6. Cuando estás subiendo algún archivo si tienes indicador del estado de la transferencia y si este incluye % de lo completado.
7. Indicar si la Web ha fallado o dado error en algún momento de las múltiples pruebas.
8. Indicar si puedes o debes registrarte en la página

### **1.3.2 Modelo Cliente Servidor**

La tecnología denominada Cliente -Servidor es utilizada por todas las aplicaciones de Internet/Intranet:

Un cliente funciona en su ordenador local, se comunica con el servidor remoto, y pide a éste información.

El servidor envía la información solicitada.

Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

1. Los sistemas Cliente-Servidor pueden ser de muchos tipos, dependiendo de las aplicaciones que el servidor pone a disposición de los clientes. Entre otros, existen:
2. Servidores de Impresión, mediante el cual los usuarios comparten impresoras.
3. Servidores de Archivos, con el cual los clientes comparten discos duros.
4. Servidores de Bases de Datos, donde existe una única base de datos.
5. Servidores de Lotus Notes, que permite el trabajo simultáneo de distintos clientes con los mismos datos, documentos o modelos.
6. Los Servidores Web también utilizan la tecnología Cliente- Servidor, aunque añaden aspectos nuevos y propios a la misma.

La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

1. Ventajas de la arquitectura cliente-servidor
2. Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
3. Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

Características de un servidor:



Voz pasiva (esclavo) • Espera para las peticiones • Sobre el recibo de peticiones, las procesa y entonces los servicios son contestados.

Características de un cliente:

Activo (amo) • Envía peticiones • Las esperas para y reciben contestaciones del servidor.

Ventajas:

Todos los datos se almacenan en los servidores, así que tienen capacidad mejor del control de la seguridad. El servidor puede controlar el acceso y el recurso al cerciorarse que dejó solamente esos accesos de usuarios permitidos y cambia datos. Es más flexible que el paradigma del P2P para poner al día los datos u otros recursos. Hay las tecnologías maduras diseñadas para el paradigma de C/S que asegura seguridad, el usuario-friendliness del interfaz, y la facilidad de empleo. Cualquier elemento de la red C/S puede ser aumentado fácilmente.

Desventajas:

La congestión del tráfico ha sido siempre un problema desde el primer día del nacimiento del paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones al mismo servidor al mismo tiempo, puede ser que cause muchos de los apuros para el servidor. Más clientes hay más apuros que tiene. Mientras que, el ancho de banda de la red del P2P se compone de cada nodo en la red, cuanto más nodos hay, mejor el ancho de banda que tiene. • El paradigma de C/S no tiene la buena robustez como red del P2P. Cuando el servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes del P2P, los recursos están situados generalmente en nodos por todas partes de la red. Aunque algún nodos salen o abandonan la descarga; otros nodos pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red. • El software y el hardware de un servidor es generalmente muy terminantes. Un hardware regular del ordenador personal puede no poder servir sobre cierta cantidad de clientes. Mientras tanto, una edición casera de Windows XP incluso no tiene IIS a trabajar como servidor. Necesita software y el hardware específico satisfacer el trabajo. Por supuesto, aumentará el coste.

### **1.3.3 Lenguajes de Programación**

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

**PHP** fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5. Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma, con una gran librería de funciones. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos comunes, como MySQL, mSQL, postgresQL, Oracle, Informix y ODBC.

### **1.3.4 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 (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

1. Sobre el servicio Web *clásico* podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:
  2. Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse más lenguajes mediante el uso de plugins.

3. Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

### **Servidor Web Apache**

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 <sup>[1]</sup> y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a *patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

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.

### **1.3.5 Ajax**

AJAX, acrónimo de **A**synchronous **J**avaScript **A**nd **X**ML (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. Es una combinación de tecnologías ya existentes:

1. XHTML(o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
2. Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.

3. El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.

4. XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Está basado en una nueva tecnología web que aúna los beneficios de los abordajes basados en pantallas y los abordajes basados en páginas. Permitiéndole más funcionalidad sofisticada utilizando estándares web fáciles de implementar, provee una alternativa real para crear aplicaciones web poderosas.

Aprovecha las mismas herramientas que los abordajes basados en páginas: XHTML, CSS, y JavaScript.

Repentinamente es un tópico popular y la razón es la continua adopción de navegadores que cumplen con los estándares que soportan la tecnología Ajax, más notablemente Firefox, Safari, Opera e Internet Explorer 6.

Las características principales de esta tecnología son:

1. Libertad del Refresco de Página: Las aplicaciones Ajax no necesitan refrescar la página completa para actualizar la información. Dándole a los usuarios una respuesta instantánea a sus ingresos y consultas. Esto le permite a los usuarios ver continuamente con lo que están trabajando y reaccionar a cualquier cambio, error, o actualización que la interfaz les notifique.

2. Chequeo y guardado instantáneo del campo: Por razones de seguridad es necesario chequear todos los campos también del lado del servidor. Afortunadamente, Ajax permite también que esto suceda.

3. Interfaz De Pantalla Única: La interfaz de pantalla única le permite al usuario modificar y cambiar información en el orden que elija. Si quiere agregar su información de cuenta primero debe poder hacerlo. O, si quiere retroceder y cambiar algo que ya hizo

4. Relativamente fácil de implementar: Debido a que las aplicaciones Ajax están construidas utilizando nada más que estándares web actuales, ellas son relativamente sencillas de construir.

Ventajas representativas del uso de AJAX:

1. La experiencia de usuario en la navegación es mucho más rica. Ya no se refresca la página constantemente al interactuar con ella.
2. El tiempo de espera para una petición se reduce. El usuario al hacer un pedido (request) al servidor, no se envía toda la página.
3. Por la misma razón anterior el tráfico al servidor se reduce.

Desventajas que debemos tener en cuenta:

1. Falta de integración con el botón retroceder del navegador. Se debe tener en cuenta esto al intentar guardar funcionalidad con este botón.
2. Falta de soporte para todos los navegadores. Aunque esto se va reduciendo, el problema se presenta por la falta de soporte para JavaScript y XMLHttpRequest.
3. Problemas si el usuario ha deshabilitado el uso de JavaScript en su navegador. Hay que tener esto en cuenta cuando desarrollamos nuestro sitio web, para enfrentarnos a esta situación.
4. No poder recomendar links específicos. Si hace que toda la web sea interactiva, no podremos recomendar algún link, ya que el contenido fue generando dinámicamente. Se debería encontrar un equilibrio dependiendo de las necesidades de su escenario.
5. Demasiado código Ajax hace lento el navegador. A más Ajax, más uso de código JavaScript del lado del browser, por consiguiente mayor trabajo del browser. El rendimiento del cpu puede verse afectado. Hace años recuerdo un código JavaScript que consumía casi toda la capacidad del CPU, por eso no hay que abusar del AJAX

### **1.3.6 Patrones de Diseño**

Un **patrón de diseño** es:

1. Una solución estándar para un problema común de programación
2. Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios
3. Un proyecto o estructura de implementación que logra una finalidad determinada
4. Un lenguaje de programación de alto nivel
5. Una manera más práctica de describir ciertos aspectos de la organización de un programa

6. Conexiones entre componentes de programas
7. La forma de un diagrama de objeto o de un modelo de objeto.

Los patrones de diseño pretenden:

1. Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
2. Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
3. Formalizar un vocabulario común entre diseñadores.
4. Estandarizar el modo en que se realiza el diseño.
5. Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

1. Imponer ciertas alternativas de diseño frente a otras.
2. Eliminar la creatividad inherente al proceso de diseño.

No es obligatorio utilizar los patrones siempre, solo en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable.

### **Patrones Creacionales**

Fábrica abstracta: Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí y haciendo transparente el tipo de familia concreta que se esté usando.

1. Constructor virtual: Abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto.
2. Método de fabricación: Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear.
3. Prototipo: Crea nuevos objetos clonándolos de una instancia ya existente.
4. Instancia única: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

### **Patrones Estructurales**

Adaptador: Adapta una interfaz para que pueda ser utilizada por una clase que de otro modo no podría utilizarla.

1. Puente: Desacopla una abstracción de su implementación.
2. Objeto compuesto: Permite tratar objetos compuestos como si de uno simple se tratase.
3. Envoltorio: Añade funcionalidad a una clase dinámicamente.
4. Fachada: Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.
5. Peso ligero: Reduce la redundancia cuando gran cantidad de objetos poseen idéntica información.
6. Proxy: Mantiene un representante de un objeto.

### **Patrones de Comportamiento**

1. Cadena de responsabilidad: Permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.
2. Orden: Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma.
3. Intérprete: Dado un lenguaje, define una gramática para dicho lenguaje, así como las herramientas necesarias para interpretarlo.
4. Iterador: Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos.
5. Mediado): Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.
6. Recuerdo: Permite volver a estados anteriores del sistema.
7. Observador: Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.
8. Estado: Permite que un objeto modifique su comportamiento cada vez que cambie su estado interno.
9. Estrategia: Permite disponer de varios métodos para resolver un problema y elegir cuál utilizar en tiempo de ejecución.

10. Método plantilla: Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

11. Visitante: Permite definir nuevas operaciones sobre una jerarquía de clases sin modificar las clases sobre las que opera.

### **1.3.7 Modelo Vista Controlador**

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos.

1. Modelo: Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.

2. Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

3. Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)

2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.



3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

### **1.3.8 Sistema de Gestión de Base de Datos**

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y *DataBase Management System*, su expresión inglesa.

Existen distintos objetivos que deben cumplir los SGBD:

1. Abstracción de la información. Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios *niveles de abstracción*.

2. Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
3. Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
4. Consistencia. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
5. Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
6. Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
7. Respaldo y recuperación. Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
8. Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
9. Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Ventajas:

1. Facilidad de manejo de grandes volúmenes de información.
2. Gran velocidad en muy poco tiempo.
3. Independencia del tratamiento de información.
4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
6. Integridad referencial al terminar los registros.

Inconvenientes:

1. El costo de actualización del hardware y software son muy elevados.
2. Costo (salario) del administrador de la base de datos es costoso.
3. El mal diseño de esta puede originar problemas a futuro.
4. Un mal adiestramiento a los usuarios puede originar problemas a futuro.
5. Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
6. Generan campos vacíos en exceso.
7. El mal diseño de seguridad genera problemas en esta.

### **Oracle**

Oracle es una potente herramienta cliente/servidor para la gestión de Bases de Datos. Es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.

### **Microsoft SQL Server**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera

simultánea. Así de tener unas ventajas que a continuación se pueden describir. Entre sus características figuran:

1. Soporte de transacciones.
2. Escalabilidad, estabilidad y seguridad.
3. Soporta procedimientos almacenados.
4. Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
5. Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
6. Permite administrar información de otros servidores de datos

### **Mysql Server**

MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

### **PostgreSQL**

PostgreSQL es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD.

Algunas de sus principales características son:

1. Alta concurrencia
2. Amplia variedad de tipos nativos
3. Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*).
4. Disparadores (*triggers*).
5. Vistas.

6. Integridad transaccional.
7. Herencia de tablas.
8. Tipos de datos y operaciones geométricas

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

PostgreSQL es un ORDBMS con licencia BSD (esta licencia básicamente consiste en que ves el código, puedes redistribuirlo y puedes modificarlo. La FSF lo considera, junto con la licencia GPL, Software libre) y ampliamente utilizado.

Este tipo de software es fundamental en cualquier esquema computacional moderno (centrado en las bases de datos).

PostgreSQL tiene la mayoría de las características presentes en grandes DBMSs comerciales, tales como transacciones, subconsultas, gatillos (triggers), vistas, integridad referencial con llaves externas, y bloqueo sofisticado. También tenemos algunas características que no tienen las otras, como tipos definidos por el usuario, herencia, reglas y control de concurrencia multi-versión para reducir el bloqueo de controversias.

### **1.3.9 Proceso de Desarrollo**

#### **Rational Unified Process**

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:

1. Inicio, El Objetivo en esta etapa es determinar la visión del proyecto.
2. Elaboración, En esta etapa el objetivo es determinar la arquitectura óptima.
3. Construcción, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

4. Transmisión, El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

#### Disciplina de Desarrollo

1. Ingeniería de Negocios: Entendiendo las necesidades del negocio.
2. Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
3. Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
4. Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
5. Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

#### Disciplina de Soporte

1. Configuración y administración del cambio: Guardando todas las versiones del proyecto.
2. Administrando el proyecto: Administrando horarios y recursos.
3. Ambiente: Administrando el ambiente de desarrollo.
4. Distribución: Hacer todo lo necesario para la salida del proyecto

### **1.3.10 Herramientas utilizadas**

#### **Dreamweaver 8**

Dreamweaver 8 es un software fácil de usar que permite crear páginas Web profesionales. Las funciones de edición visual de Dreamweaver 8 permiten agregar rápidamente diseño y funcionalidad

a las páginas, sin la necesidad de programar manualmente el código HTML. Se puede crear tablas, editar marcos, trabajar con capas, insertar comportamientos JavaScript, etc., de una forma muy sencilla y visual.

Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios Web, actualizando el sitio Web en el servidor sin salir del programa.

### **Zend Studio**

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos.

### **Photoshop**

Adobe Photoshop es una aplicación informática de edición y retoque de imágenes bitmap, jpeg, gif, etc, elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

Photoshop en sus primeras versiones trabajaba en un espacio bitmap formado por una sola capa, donde se podían aplicar toda una serie de efectos, textos, marcas y tratamientos. En cierto modo tenía mucho parecido con las tradicionales ampliadoras. En la actualidad lo hace con múltiples capas.

A medida que ha ido evolucionando el software ha incluido diversas mejoras fundamentales, como la incorporación de un espacio de trabajo multicapa, inclusión de elementos vectoriales, gestión avanzada de color (ICM / ICC), tratamiento extensivo de tipografías, control y retoque de color, efectos creativos, posibilidad de incorporar plugins de terceras compañías, exportación para web entre otros.

Photoshop se ha convertido, casi desde sus comienzos, en el estándar mundial en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño web, composición de imágenes bitmap, estilismo digital, fotocomposición, edición y grafismos de vídeo y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales.

Photoshop ha dejado de ser una herramienta únicamente usada por diseñadores / maquetadores, ahora Photoshop es una herramienta muy usada también por fotógrafos profesionales de todo el mundo, que lo usan para realizar el proceso de "positivado y ampliación" digital, no teniendo que pasar ya por un laboratorio más que para la impresión del material.

Con el auge de la fotografía digital en los últimos años, Photoshop se ha ido popularizando cada vez más fuera de los ámbitos profesionales y es quizá, junto a Windows y Flash (de Macromedia) uno de los softwares que resulta más familiar (al menos de nombre) a la gente que comienza a usarlo, sobre todo en su versión Photoshop Elements, para el retoque casero fotográfico.

Aunque el propósito principal de Photoshop es la edición fotográfica, este también puede ser usado para crear imágenes, efectos, gráficos y más en muy buena calidad.

### **Rational Rose**

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML. Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (Investigación Preliminar, Análisis, Diseño, Implementación e Instalación.). CASE es también definido como el Conjunto de métodos, utilidades y técnicas que facilitan el mejoramiento del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases.

Se puede ver al CASE como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

### **PgAdmin**

PgAdmin es una herramienta de propósito general para diseñar, mantener, y administrar las bases de datos de Postgres. Funciona bajo Windows 95/98 y NT.

Características incluidas:

Entradas SQL aleatorias.



1. Pantallas de información y 'Ayudas' para bases de datos, tablas, índices, secuencias, vistas, programas de arranque, funciones y lenguajes.
2. Preguntas y respuestas para configurar Usuarios, Grupos y Privilegios.
3. Control de revisión con mejora de la generación de script.
4. Configuración de las tablas de Microsoft MSysConf.
5. `Ayudas` para importar y exportar datos.
6. `Ayuda` para migrar Bases de datos.
7. Informes predefinidos en bases de datos, tablas, índices, secuencias, lenguajes y vistas.

#### **1.4 Conclusiones**

En este capítulo se abordaron las tecnologías y tendencias actuales mas importantes. Cuando se habla de tecnología de información, entramos en un territorio de fronteras cada vez más amplias y difusas (por la capacidad simbiótica de mezclarse con otras tecnologías), y de cambios cada vez más frecuentes.

## **Capítulo 2 Características del Sistema**

### **2.1 Introducción**

Con el objetivo de comprender la estructura y la dinámica de la organización en la cual se va a implementar el sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales se desarrolla este capítulo.

El modelo del negocio describe los procesos existentes u observados, con el propósito de comprenderlos. La información será expresada mediante actores, caso de uso y trabajadores.

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software. Durante muchos años muchas aplicaciones han fallado (no se culminaron o no se usaron) porque existieron incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene.

Aquí radica la importancia que en los últimos años se le ha dado a la identificación de los requerimientos como parte del proceso de desarrollo del software.

Las áreas de esfuerzo del análisis de requerimientos están en:

1. Reconocimiento del problema como lo ve el usuario.
2. Evaluación del problema y síntesis de la evaluación: Como evaluación se entiende a la definición de los datos observables, la evaluación del flujo y contenido de la información. La definición de las funciones del software, entender el comportamiento del software ante eventos, el establecimiento de las características de la interfaz y el descubrimiento de restricciones adicionales de diseño. Toda esta evaluación se sintetiza en la definición en detalle de los datos, funciones y el comportamiento del sistema.
3. Modelado: Creación de modelos que ayuden a entender la entidad a construir.
4. Construcción de un prototipo de alto nivel del sistema.
5. Revisión por parte del usuario.

6. Firma del contrato si las partes están de acuerdo.

## **2.2 Objeto de Estudio**

### **2.2.1 Problema y situación problemática.**

#### **Problema Científico**

¿Cómo lograr solucionar las dificultades que existen en la secretaría docente de la Facultad 2, para que la información esté clasificada, organizada y accesible para una mayor rapidez y eficiencia en el trabajo?

#### **Situación Problemática**

En la secretaría docente de la Facultad 2, existe un software instalado actualmente que se usa para el procesamiento de la información relacionada con la matrícula:

Gestión de estudiantes y estructuras, los Estudiantes

Herramienta que mantiene informado a los estudiantes sobre su desempeño académico, los Registros

Control del desarrollo del proceso docente, notas y asistencia, los Profesores

Gestión del claustro del centro, el Plan de estudio

Definición de los planes de estudio. Asignaturas, disciplinas, perfiles y los Reportes

Diseño y generación de reportes personalizados.

En este centro también se maneja diariamente mucho volumen de información acerca de los estudiantes que se les otorga el pase eventual, licencias de matrículas y licencias especiales pero no existe ningún software que gestione dicha información. Como hemos visto existen variantes que facilitan el trabajo de los trabajadores, pero aún hay trabajo que se hace manualmente provocando lentitud, difícil acceso, pérdida de información e ineficiencias en el trabajo.

### **2.2.2 Objeto de automatización.**

Los procesos objeto de automatización en el sistema son los procesos de optimización de la información, procesamiento de grandes volúmenes de datos, desarrollo de contenidos, solicitud de pases eventuales, los procesos de solicitud de licencias de matrícula y por último los procesos de otorgamiento de licencias especiales. Estos procesos actualmente se desempeñan de forma manual.

### **2.2.3 Información que se maneja.**

Los datos que se manejan son los referentes a los estudiantes que se le otorgue un pase eventual, licencia de matrícula y licencia especial.

1. Modelo de pase: Modelo que se entrega a los estudiantes cuando se les otorga el pase eventual.
2. Formato de Pase Eventual: Modelo por el cual se gestionan los datos de los estudiantes que se les otorgó pase eventual.
3. Solicitud de Licencia de Matrícula: Documento oficial mediante el cual se hace efectiva la solicitud de licencia de matrícula.
4. Formato de Licencia de Matrícula: Modelo por el cual se gestionan los datos de los estudiantes que se les otorgó licencia de matrícula.
5. Modelo de Tramitación de bajas: Modelo mediante el cual se gestiona la entrega de los medios básicos del centro.
6. Modelo SG\_3: Modelo en el cual se declaran los motivos por los que se otorgan las licencias de matrículas.
7. Resolución decanal para las licencias de matrículas: Documento oficial en el que se declara la licencia de matrícula.
8. Autorización de repitencia: Documento oficial para el otorgamiento de las licencias especiales.
9. Documento de licencia de matrícula incorporada: Documento oficial como constancia de la incorporación de la licencia.
10. Formato de Licencia especial: Modelo por el cual se gestionan los datos de los estudiantes que se les otorgó licencia especial.
11. Resolución decanal para las licencias especiales: Documento oficial en el que se declara la licencia especial.

### **2.2.4 Propuesta del sistema.**

Para lograr una mayor eficiencia en el trabajo y tener mejor control y acceso a la información que se maneja en la secretaría docente de la Facultad 2 vinculada al otorgamiento de pases eventuales, licencias especiales y licencias de matrículas, se va a desarrollar un sistema que resolverá los problemas antes

mencionados. Con el desarrollo de este sistema para la gestión de información se desea tener toda la información disponible, actualizada y organizada. El sistema a desarrollar brindará opciones como:

1. Gestionar pases eventuales: Permitirá que cuando el estudiante haga la petición de un pase eventual y le sea otorgado, se tenga un registro que contenga los datos del estudiante, la fecha solicitada y los motivos de la petición.
2. Gestionar licencias de matrícula: Permite que cuando a un estudiante se le otorgue la licencia de matrícula, se tenga un registro que contenga los datos del estudiante y motivo de la licencia.
3. Gestionar licencias especiales: Permite que cuando al estudiante se le otorgue licencia especial, se tenga un registro de todos los estudiantes que se les otorgó la licencia y en que año y la actividad a la cual han sido asignados.

## **2.3 Modelo del negocio actual**

En estos momentos los estudiantes se presentan en el decanato de la facultad o en la secretaría docente donde se les otorga licencias especiales, licencias de matrículas o pases eventuales. A los mismos se les suministran los formularios necesarios para tramitar las licencias y una vez finalizado el proceso se archivan estos documentos para un posterior uso cuando el estudiante se incorpore a las actividades docentes. En caso de solicitar pase eventual si sus argumentos son sólidos se procede a llenar el modelo de pase y archivar de alguna forma los datos de los estudiantes que están de pase.

### **2.3.1 Reglas del negocio a considerar**

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

#### **Pases Eventuales**

1. Únicamente la Decana, Vicedecana de Formación y el Vicedecano de Extensión Universitaria están autorizados a dar pases eventuales.

2. El estudiante tiene que presentarse personalmente a solicitar el pase eventual.
3. El estudiante tiene que presentar su solapín para llenar el modelo de pase eventual.

#### **Licencias de Matrículas**

1. Las solicitudes de las licencias de matrículas tienen que ser por escrito.
2. Las solicitudes de matrículas se entregarán a la decana personalmente.
3. Únicamente la decana está autorizada a aprobar las licencias de matrículas.
4. El estudiante tiene que tramitar personalmente el documento de tramitación de bajas.
5. El estudiante tiene que entregar el modelo de tramitación de bajas con todas las firmas correctas para poder continuar con el proceso de solicitud de licencia.
6. El estudiante tiene que recibir una copia del modelo SG-3

#### **Licencias Especiales**

1. El estudiante solo puede suspender 1 o 2 mundiales
2. Las licencias especiales solamente se otorgan una vez durante la carrera y en un caso muy excepcional dos veces.
3. Los estudiantes que se le otorgan las licencias especiales tienen que estar vinculados a un Trabajo Socialmente Útil (TSU) especial durante un curso.

### **2.3.2 Determinación y justificación de los actores y trabajadores del negocio.**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Son actores candidatos del negocio:

1. Socios
2. Proveedores
3. Autoridades (legales, reguladoras, etc.)
4. Propietarios sin no están dentro del negocio que se modela.
5. Sistemas de información externos al negocio.

6. Otras partes del negocio si este es grande y esas partes no están dentro del campo de acción del negocio que se modela.

<b>Actores del negocio</b>	<b>Descripción</b>
Estudiante	Solicita licencias de matriculas, licencias especiales y pases eventuales

**Tabla1.Actores del negocio.**

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

<b>Trabajadores del negocio</b>	<b>Descripción</b>
Decana de la Facultad	Autoriza las licencias de matriculas y licencias especiales.
Vicedecana Formación	Interviene en el proceso de autorizo de licencias especiales.
Secretaria Docente	Gestiona todo el proceso de autorizo de licencias de matriculas y licencias especiales
Responsable	Es la persona encargada de autorizar los pases eventuales( Decana de la Facultad, Vicedecano de Formación y Vicedecano de Extensión Universitaria )
Comisión	Es la encargada de decidir de los estudiantes que suspenden en el mundial cuales tienen derecho a obtener la licencia especial.

**Tabla 2. Trabajadores del negocio.**

### 2.3.3 Diagrama de Casos de Uso del Negocio

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio.

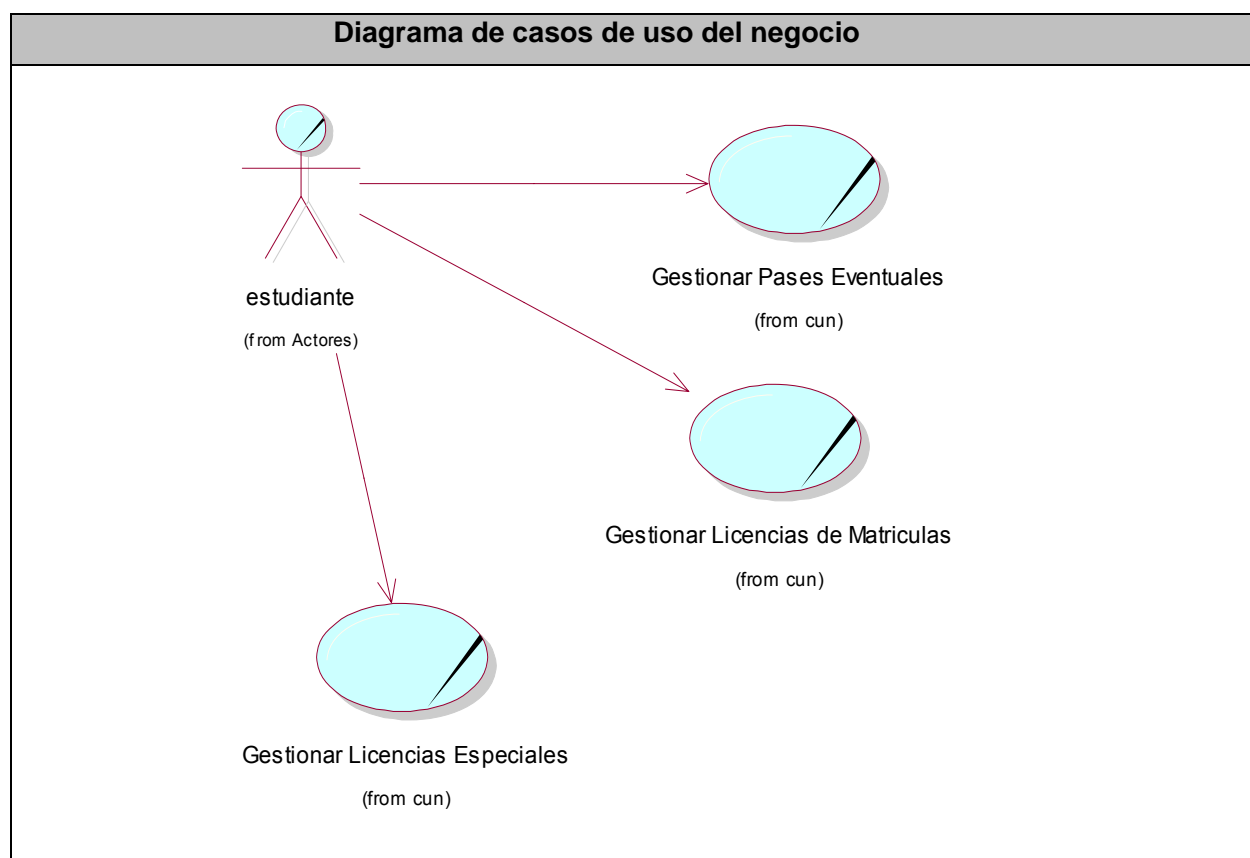


Tabla 3. Diagrama de Casos de Uso del Negocio

### 2.3.4 Descripción Textual del los Casos de Uso del Negocio

[Ver Anexo 1](#)

### 2.3.5 Diagramas de Actividades

El diagrama de actividad es un grafo (grafo de actividades) que contiene estados en que puede hallarse una actividad. Un estado de actividad representa la ejecución de una sentencia de un



procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En vez de esperar un evento, como en un estado de espera normal, un estado de actividad espera la terminación de su cómputo. Cuando la actividad termina, entonces la ejecución procede al siguiente estado de actividad dentro del grafo. Una transición de terminación es activada en un diagrama de actividades cuando se completa la actividad precedente.

[Ver Anexo 2](#)

### **2.3.6 Modelo de objetos**

Los modelos de objetos del dominio están asociados a cada uno de los casos de uso del negocio.

[Ver Anexo3](#)

### 2.4 Relación de los Requerimientos.

Se define un requerimiento como:

Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

1. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
2. Una representación documentada de una condición o capacidad como en 1 o 2.
3. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

#### **2.4.1 Definición de los requisitos funcionales**

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requerimientos funcionales no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

R 1- Gestionar pases eventuales.

R 1.1- Insertar estudiantes.

R 1.2- Eliminar estudiantes.

R 1.3- Modificar estudiantes.

R 1.4- Listar los estudiantes de pases.

**R 2- Gestionar licencias de matrículas.**

R 2.1- Insertar estudiantes.

R 2.2- Eliminar estudiantes.

R 2.3- Modificar estudiantes.

R 2.4- Listar los estudiantes que se les otorgó licencia de matrícula.

**R 3- Gestionar licencias especiales.**

R 3.1- Insertar estudiantes.

R 3.2- Eliminar estudiantes.

R 3.3- Modificar estudiantes.

R 3.4- Listar los estudiantes que se les otorgó licencia especial.

**R 4- Autenticarse.**

**R 5- Administración**

R 5.1 Adicionar usuario.

**2.4.2 Definición de los requisitos no funcionales.**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el

sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Para el uso de esta aplicación es necesario utilizar como navegador el firefox ya que el Internet Explorer presenta problemas con la utilización de los CSS (estilos) aplicados en el diseño y dificulta los resultados deseados.

**Requisitos de Diseño:**

Lenguaje de programación que será usado es PHP

El análisis y el diseño del sistema llevarán la metodología RUP, lenguaje de modelación UML y la herramienta el Rational Rose.

Gestor de base de datos será usado Postgre.

**Requisitos de Seguridad:**

1. Debe contar con varios niveles de acceso para permitir el trabajo organizado con el sistema.
2. Toda la información guardada de los estudiantes será confidencial, por lo que solo podrá ser consultada por personal autorizado y estará prohibida su divulgación.

**Requisitos de Usabilidad:**

1. Documentar bien la aplicación y proporcionar materiales de ayuda para hacer mejor uso de todos los servicios que este ofrece

**Requisitos de Interfaz:**

1. Interfaz con un diseño sencillo que contenga pocos gráficos, con vista a acelerar la velocidad de respuesta hacia el usuario.

**Requisitos de Soporte:**

1. El sistema debe ser integrado, de manera que se mantenga la colaboración entre todos los subsistemas que lo componen.
2. La base de datos que utilizará el sistema como medio de almacenamiento de la información estará soportada sobre un gestor de bases de datos PostgreSQL.
3. El diseño de la base de datos será multiplataforma.

**Requisitos de Rendimiento:**

1. El tiempo de respuesta de cada uno de las páginas debe ser menor que un minuto, excepto en aquellas que por las actividades que realizan, requieran más tiempo.
2. Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.

**Requisitos de Confiabilidad:**

1. El sistema debe funcionar las 24 horas del día.
2. Las salidas del sistema deben ser precisas.

**Requisitos Legales:**

1. Este sistema de Gestión de la Información para la Secretaría docente de la Facultad 2 pertenece a la Universidad de las Ciencias Informáticas, específicamente a la Secretaría Docente de la Facultad 2.

**2.5 Modelo de Casos de Uso del Sistema.**

### 2.5.1 Definición de los actores del sistema a automatizar.

Cada trabajador del negocio (inclusive si fuera un sistema ya existente) 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.

Los actores del sistema:

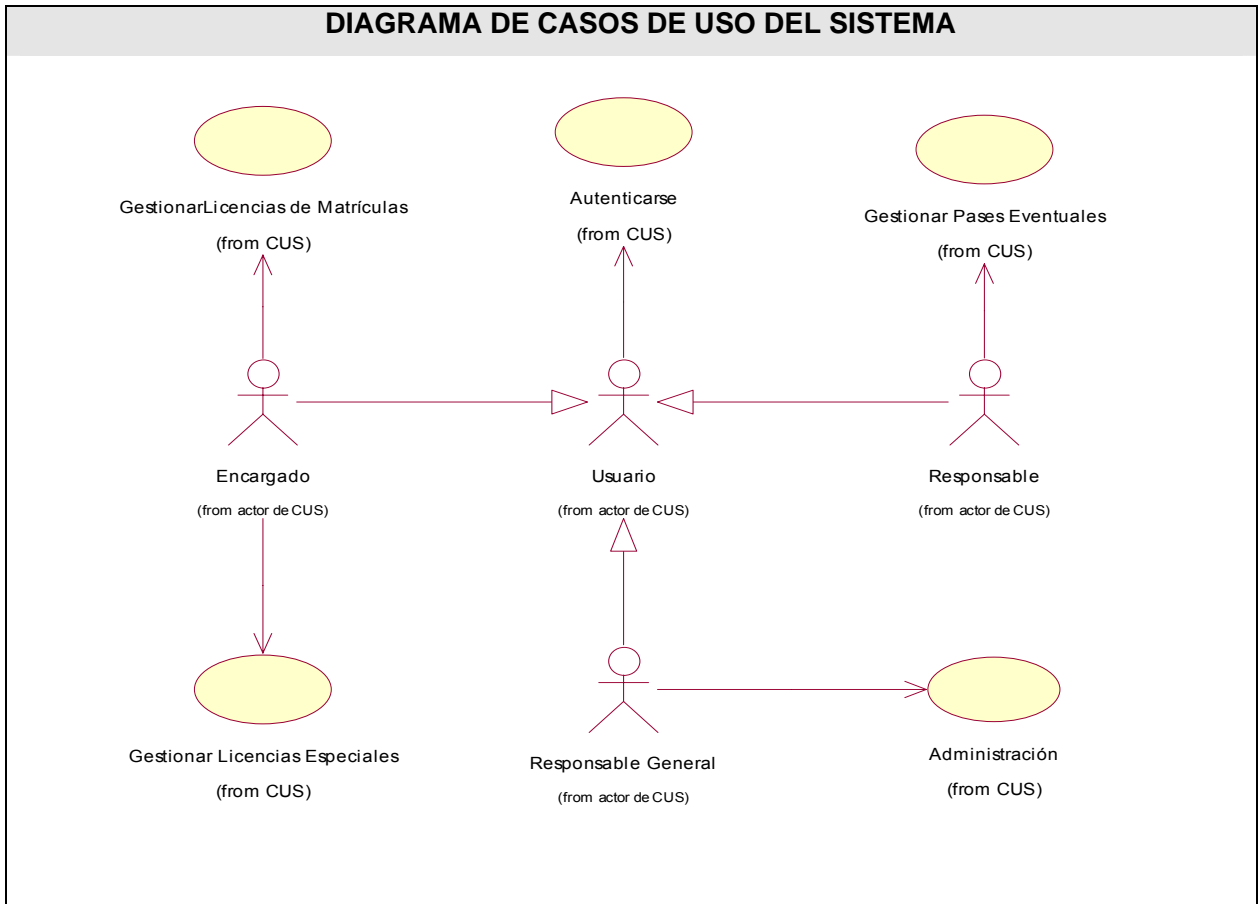
1. No son parte de él.
2. Pueden intercambiar información con él.
3. Pueden ser un recipiente pasivo de información.
4. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

Actores del Sistema	Justificación
Responsable	Personas autorizadas a dar pases eventuales (Decana, Vicedecana de Formación y Vicedecano de Extensión Universitaria). Esta encargado de recopilar los datos de los estudiantes y del pase en cuestión.
Encargado	Personas autorizadas de gestionar todo el proceso de licencias de matrículas y licencias especiales (Decana y Secretaria Docente).
Responsable General	Persona encargada de gestionar los usuario así como los permisos de cada uno

**Tabla 10. Definición de los actores del sistema a automatizar**

### 2.5.2 Diagrama de casos de uso del sistema a automatizar.

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.



**Tabla 11. Diagrama de casos de uso del sistema.**

### 2.5.3 Descripción de los casos de uso del sistema.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

Los casos de uso candidatos también se encuentran entre las actividades a automatizar. Esto no significa que una actividad se convierta en un caso de uso porque un caso de uso es un proceso que da un resultado de valor para un actor determinado y una secuencia de actividades a automatizar pueden implicar pasos dentro de un caso de uso.

[Ver Anexo 4](#)

## **2.6 Conclusiones**

El modelo del negocio da una visión de que es necesario hacer para dar respuesta a las solicitudes del usuario, lo cual se logra definiendo los procesos, roles y responsabilidades de la organización en los modelos de casos de usos del negocio. El modelo del negocio brinda una vía natural para determinar los requerimientos del sistema de la información.

Con el desarrollo de este capítulo se identificaron, enunciaron y clasificaron los requerimientos que el sistema debe cumplir. Se obtuvieron las propiedades o cualidades que el producto debe cumplir y un listado de las capacidades o funciones que el sistema debe cumplir.

Como resultado de este capítulo se obtiene una vista externa del sistema, que en el lenguaje del cliente, describe lo que se espera de él a través del diagrama de caso de uso del sistema.

## **Capítulo 3 Análisis y diseño del sistema.**

### **3.1 Introducción**

A partir de aquí se debe profundizar en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidos.

El objetivo de este capítulo es traducir los requisitos a una especificación que describe cómo implementar el sistema. Este capítulo consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace. Con este capítulo debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

### **3.2 Análisis**

#### **3.2.1 Modelo de clases de análisis.**

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. En general se siguen directrices muy parecidas a las que usamos en la construcción del modelo conceptual.

#### **3.2.2 Caso de uso Gestionar Pases Eventuales.**



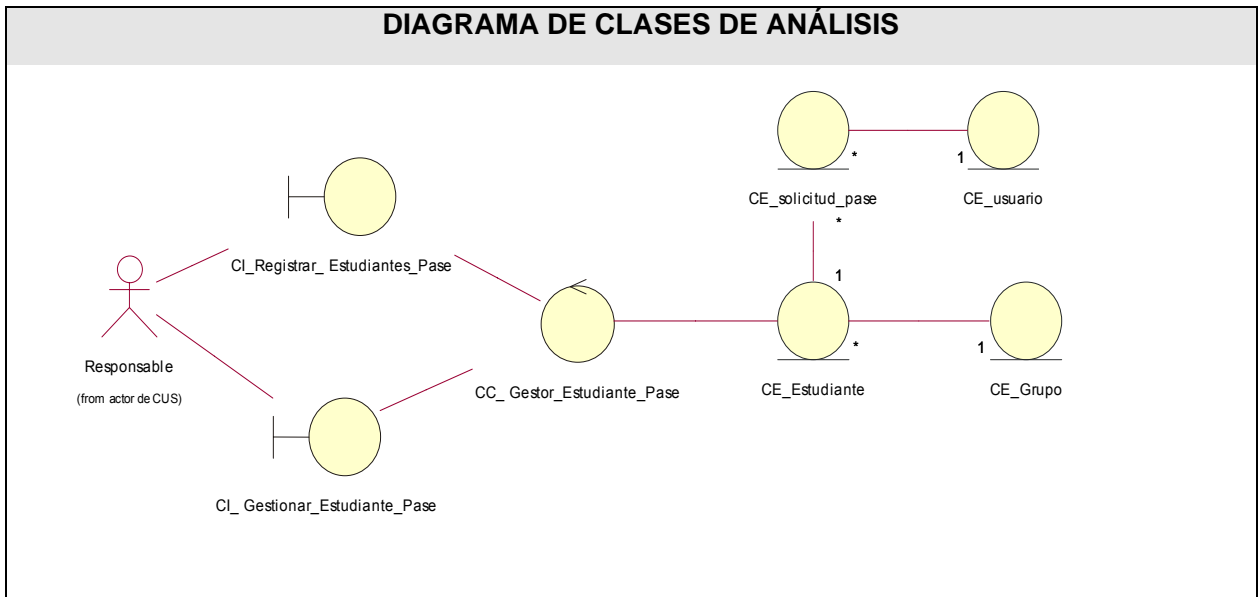


Tabla 17. Caso de Uso: "Gestionar Pases Eventuales".

### 3.2.3 Caso de uso Gestionar Licencias de Matrículas.

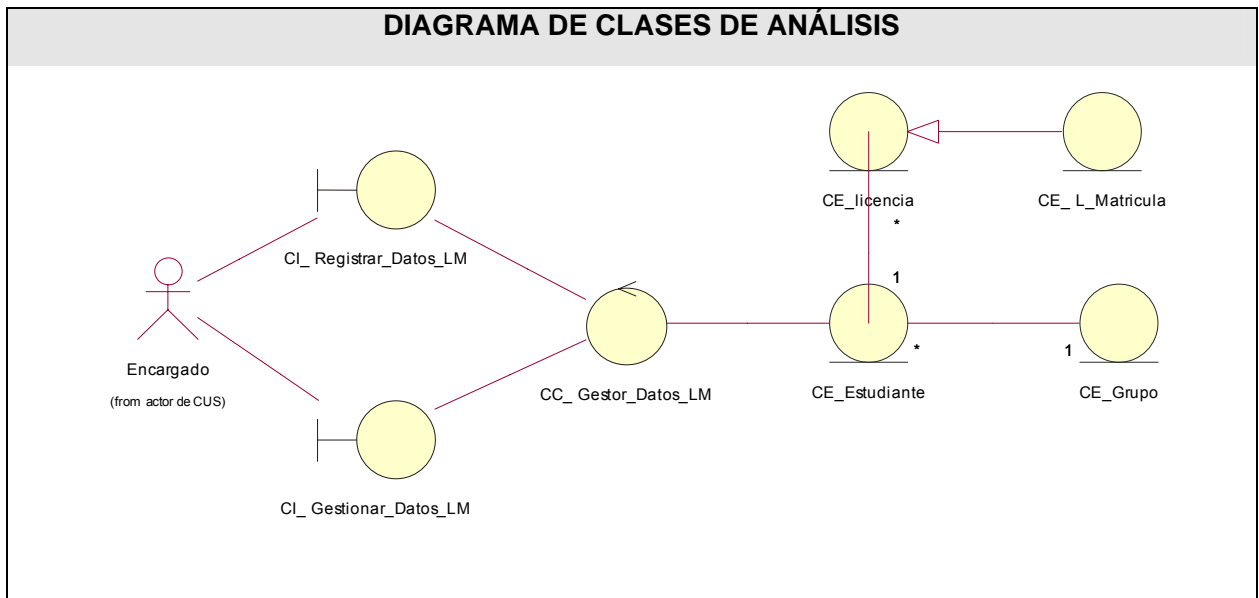


Tabla 18. Caso de Uso: "Gestionar Licencias de Matrículas".

### 3.2.4 Caso de uso Gestionar Licencias Especiales

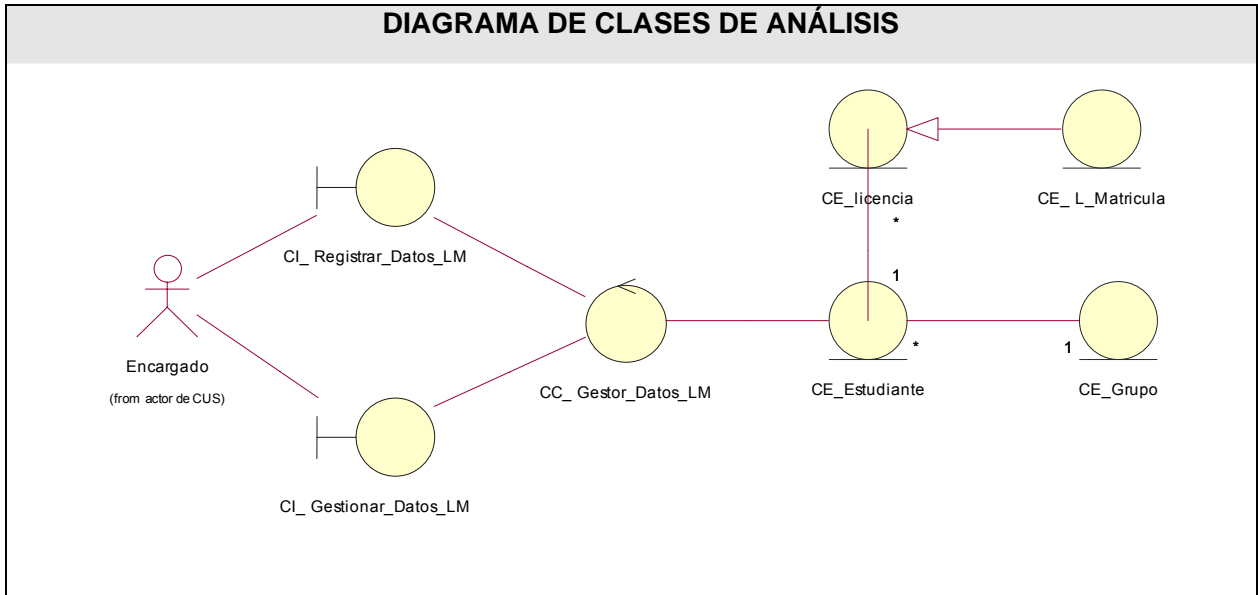


Tabla 19. Caso de Uso: "Gestionar Licencias Especiales".

### 3.2.5 Caso de uso Autenticarse

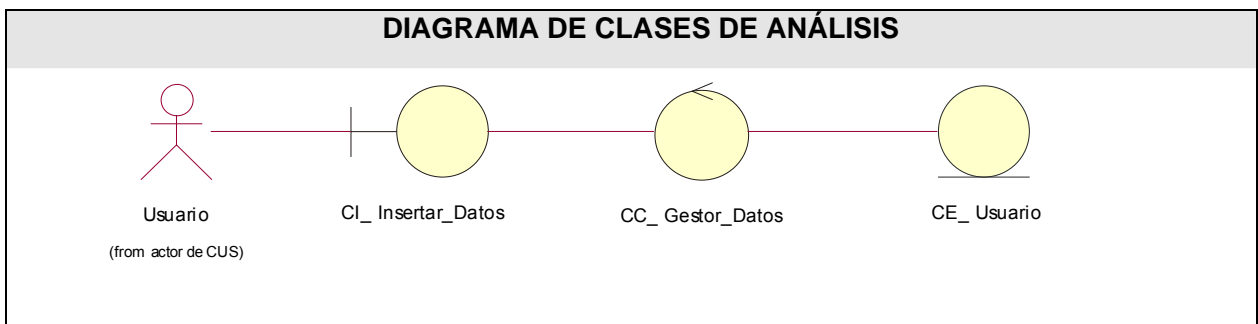


Tabla 20. Caso de Uso: "Autenticarse".

### 3.2.6 Caso de uso Administración.

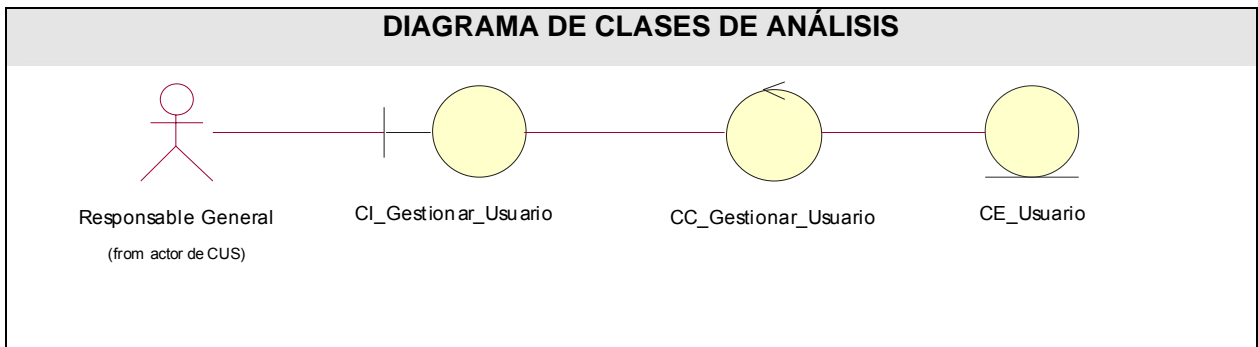


Tabla 21. Caso de Uso: "Administración".

### 3.2.7 Clases Bases

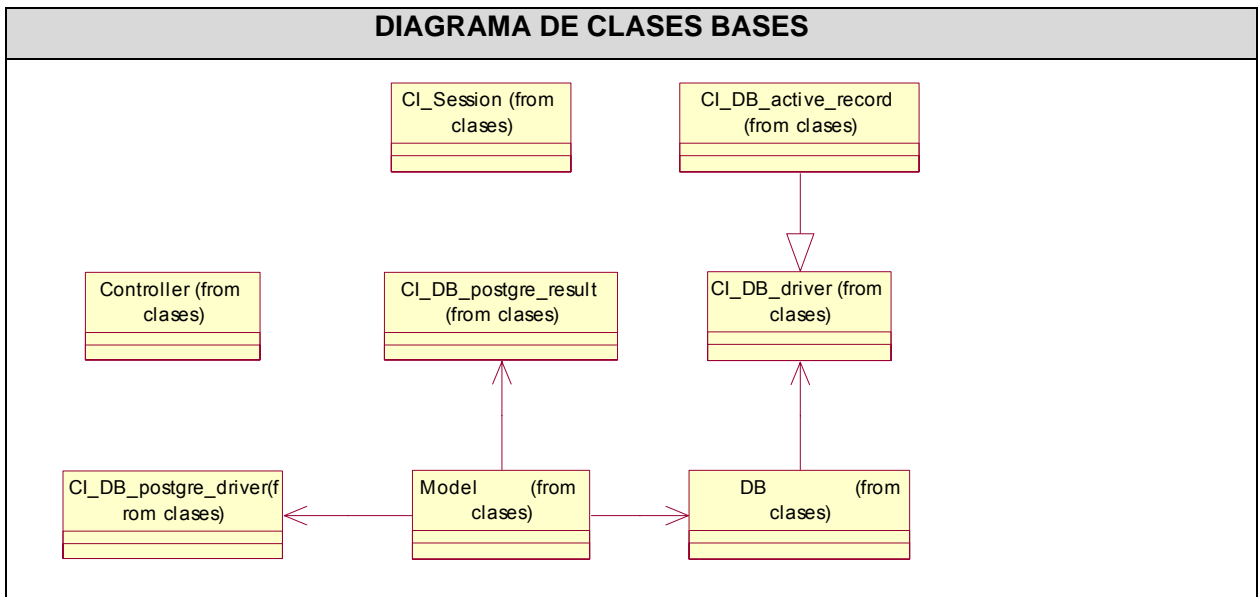


Tabla 22. Diagrama de clases bases.

#### 3.2.7.1 Descripción de las clases.

<b>Controller</b>	<b>Súper clase que se encarga de servir de base funcional de las clases controladoras.</b>
Métodos	
Controller	Constructor de la clase que se encarga de inicializar los elementos.
_ci_initialize	Asigna todas las clases bases cargadas por el inicio de la Controller por variables en esta clase. También se encarga de llamar al autoloading de la aplicación.

**Tabla 23. Descripción de las clases bases “Controller”.**

<b>Model</b>	<b>Súper clase que se encarga de servir de base funcional de las clases modelos.</b>
Métodos	
Model	Constructor de la clase que se encarga de inicializar los elementos.
_assign_libraries	Crea referencia locales a todos los objetos instanciados concurrentemente y permite que cualquier sintaxis se legalmente usado en la las controladoras o entre modelos.

**Tabla 24. Descripción de las clases bases “Model”.**

<b>Session</b>	<b>Clase que se encarga de tratar todos los métodos y funcionalidades para el trabajo con sesión</b>
Métodos	
Session	Constructor de la clase que se encarga de inicializar los elementos.
regenerate_id	Genera el id de la sesión
destroy	Destruye la sesión y borra todo lo esta guardado en la sesión
get	Devuelve lo guardado en la sesión
exist	Verifica si existe un atributo en la sesión
set	Guarda en atributos en la sesión

delete	Borra los atributos de las sesiones
_session_id_expired	Chequea si la sesión ha expirado

**Tabla 25. Descripción de las clases bases “Session”.**

<b>CI_DB_postgre_driver</b>	<b>Clase que se encarga de proveer métodos y funcionalidades importante a la hora del manejo de datos</b>
Métodos	
CI_DB_postgre_driver	Constructor de la clase que se encarga de inicializar los elementos.
db_connect	Conexión no persistente de a la base de datos.
db_pconnect	Conexión persistente a la base de datos.
_version	Muestra la versión.
_execute	Ejecuta un query.
trans_begin	Inicia transacciones.
trans_commit	Termina transacciones.
trans_rollback	Vuelve a atrás las transacciones.
Insert_id	Inserta un id.
count_all	Cuenta todas las consultas.
_list_tables	Muestra la consulta de una tabla.
_error_message	Muestra mensajes de error.
_insert	Dado los datos forma una consulta Insert.
_update	Dado los datos forma una consulta Update.
_delete	Dado los datos forma una consulta Delete.
_close	Cierra la conexión a la base de datos.

**Tabla 26. Descripción de las clases bases “CI\_DB\_postgre\_driver”.**

<b>CI_DB_postgre_result</b>	<b>Clase que se encarga de proveer métodos y funcionalidades importante a la hora del manejo de datos</b>
Métodos	

CI_DB_postgre_result	Constructor de la clase que se encarga de inicializar los elementos.
num_rows	Muestra el números de filas del resultado enviado
num_fields	Muestra el números de campos del resultado enviado
list_fields	Genera un arreglo de columnas con sus nombres
field_data	Obtiene los datos del campo

**Tabla 27. Descripción de las clases bases “CI\_DB\_postgre\_result”.**

<b>DB</b>	<b>Clase de gestionar el acceso a métodos de clases para el acceso a datos.</b>
Métodos	
DB	Constructor de la clase que se encarga de inicializar los elementos. Además carga todas las clase instanciadas a el.

**Tabla 28. Descripción de las clases bases “DB”.**

<b>CI_DB_driver</b>	<b>Clase que se encarga de proveer métodos y funcionalidades importante a la hora del manejo de datos</b>
Métodos	
CI_DB_driver	Constructor de la clase que se encarga de inicializar los elementos.
initialize	Inicializa archivos de la base de datos.
platform	Devuelve el nombre de la plataforma de base de datos que esta en uso.
version	Devuelve la versión de la base de datos
query	Ejecuta una consulta dada.
trans_off	Desabilita transacciones
trans_start	Habilita transacciones
trans_complete	Transacciones completadas
trans_status	Estado de las transacciones

is_write_type	Comprueba si la consulta es del tipo escritura
total_queries	Retorna el numero total de consulta
last_query	Retorna el ultimo query que ha sido ejecutado
list_tables	Retorna un arreglo con nombres de las tablas
table_exists	Determina si una tabla dada existe
field_exists	Determina si un campo dada existe
field_data	Retorna un objeto con los datos de los campos
insert_string	Genera una consulta Insert
update_string	Genera una consulta Update
display_error	Muestra un error en pantalla

**Tabla 29. Descripción de las clases bases “CI\_DB\_driver”.**

<b>CI_DB_active_record</b>	<b>Clase que se encarga de proveer métodos y funcionalidades importante a la hora del manejo de datos</b>
Métodos	
CI_DB_active_record	Constructor de la clase que se encarga de inicializar los elementos.
select	Genera la porción del select de la consulta
from	Genera la porción del from de la consulta
join	Genera la porción del join de la consulta
where	Genera la porción del where de la consulta
like	Genera la porción del %like% de la consulta
groupby	Genera la porción del groupby de la consulta
having	Genera la porción del having de la consulta
orderby	Genera la porción del orderby de la consulta
limit	Genera la porción del limit de la consulta
insert	Genera la porción del insert de la consulta
update	Genera la porción del update de la consulta
delete	Genera la porción del delete de la consulta

order_by	Genera la porción del order_by de la consulta
----------	---

**Tabla 30. Descripción de las clases bases “CI\_DB\_active\_record”.**

### 3.3 Diseño

#### 3.3.1 Diagramas de interacción.

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

Un Diagrama de Interacción muestra una interacción, que consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Un diagrama *de* secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes. Gráficamente, un diagrama de secuencia es una tabla que representa objetos, dispuestos a lo largo del eje X, y mensajes, ordenados según se suceden en el tiempo, a lo largo del eje Y. Un diagrama de colaboración es un diagrama de interacción que destaca la organización estructural de los objetos que envían y reciben mensajes. Gráficamente, un diagrama de colaboración es una colección de nodos y arcos.

[Ver Anexo 5](#)

#### 3.3.2 Diagrama de clases del diseño.

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación.

A diferencia del Modelo Conceptual, un Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real.



3.3.2.1 Caso de uso Gestionar Pases Eventuales.

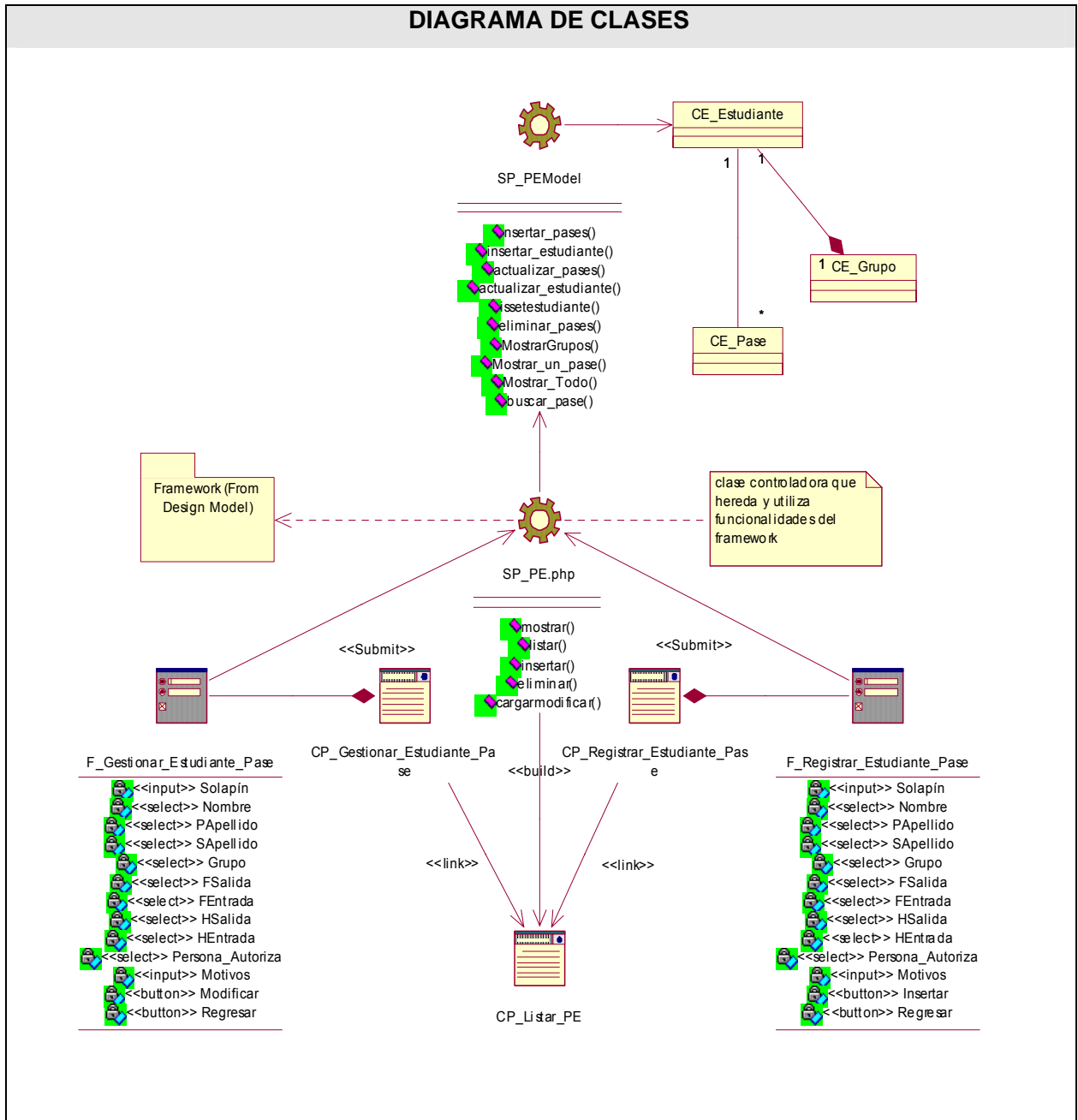


Tabla 42. Caso de uso “Gestionar Pases Eventuales”.

### **3.3.2.2 Caso de uso Gestionar Licencias de Matrículas**

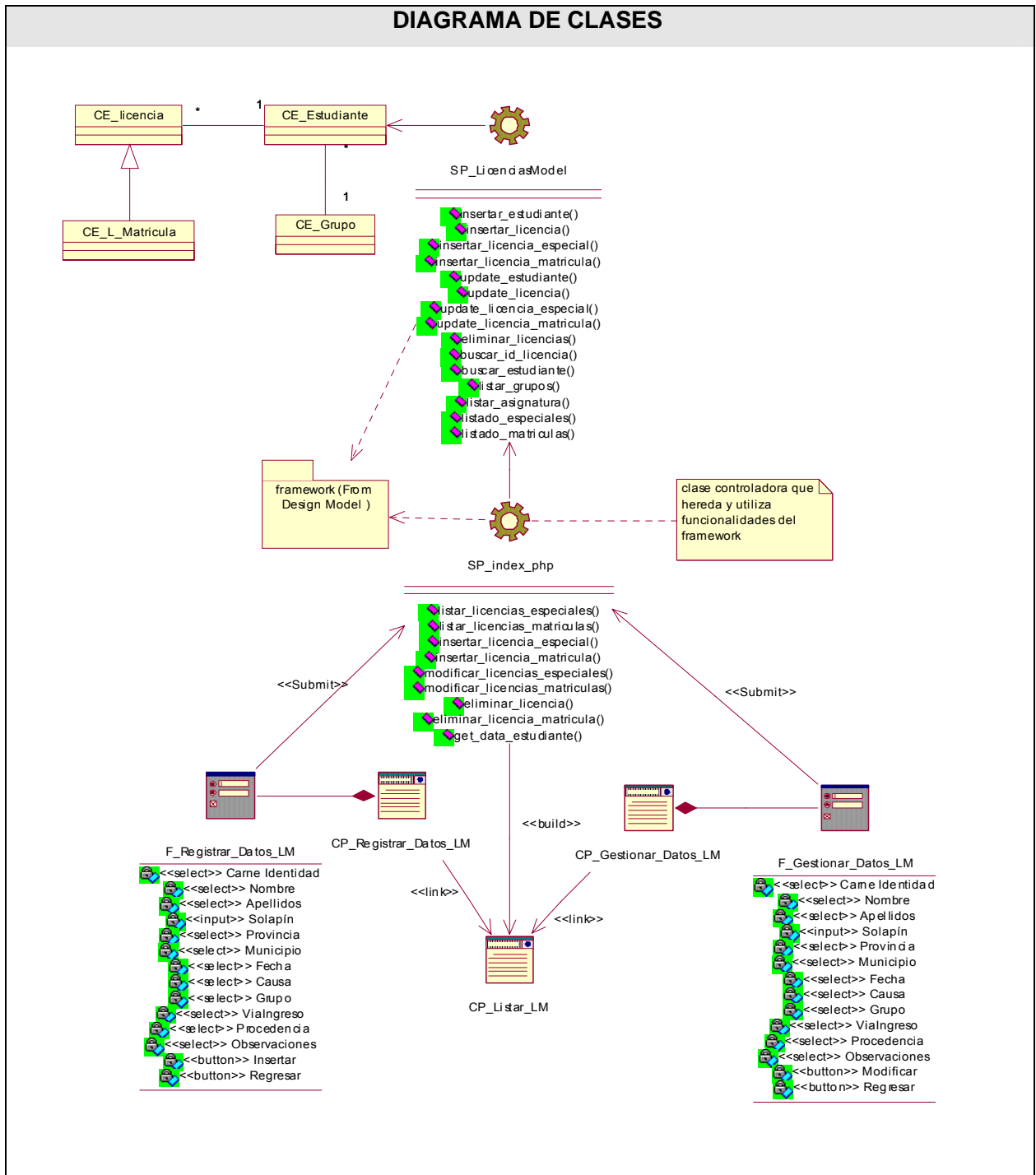


Tabla 43. Caso de uso “Gestionar Licencias de Matrículas”.

3.3.2.3 Caso de uso Gestionar Licencias Especiales

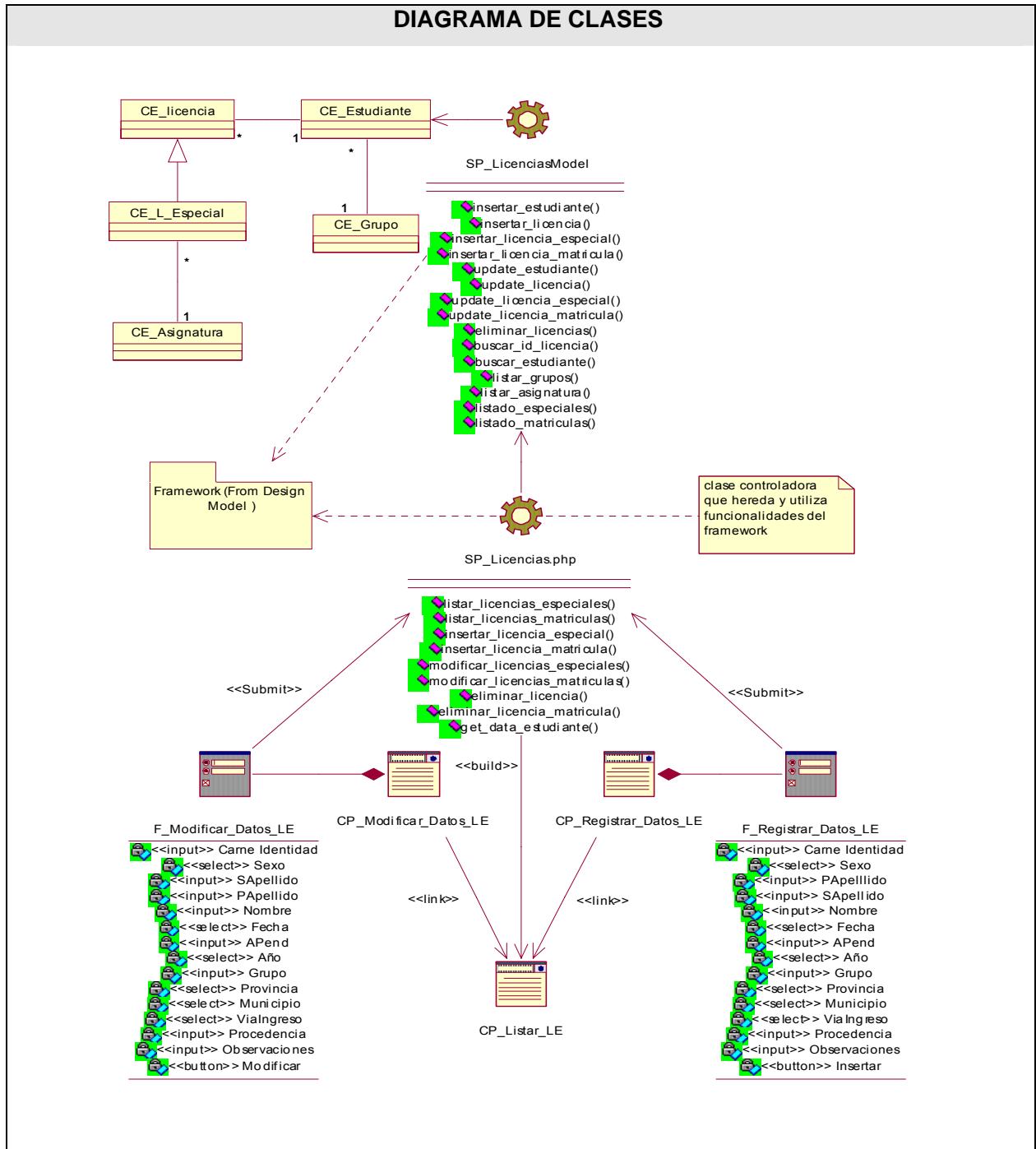


Tabla 44. Caso de uso “Gestionar Licencias Especiales”.

3.3.2.4 Caso de uso Autenticarse

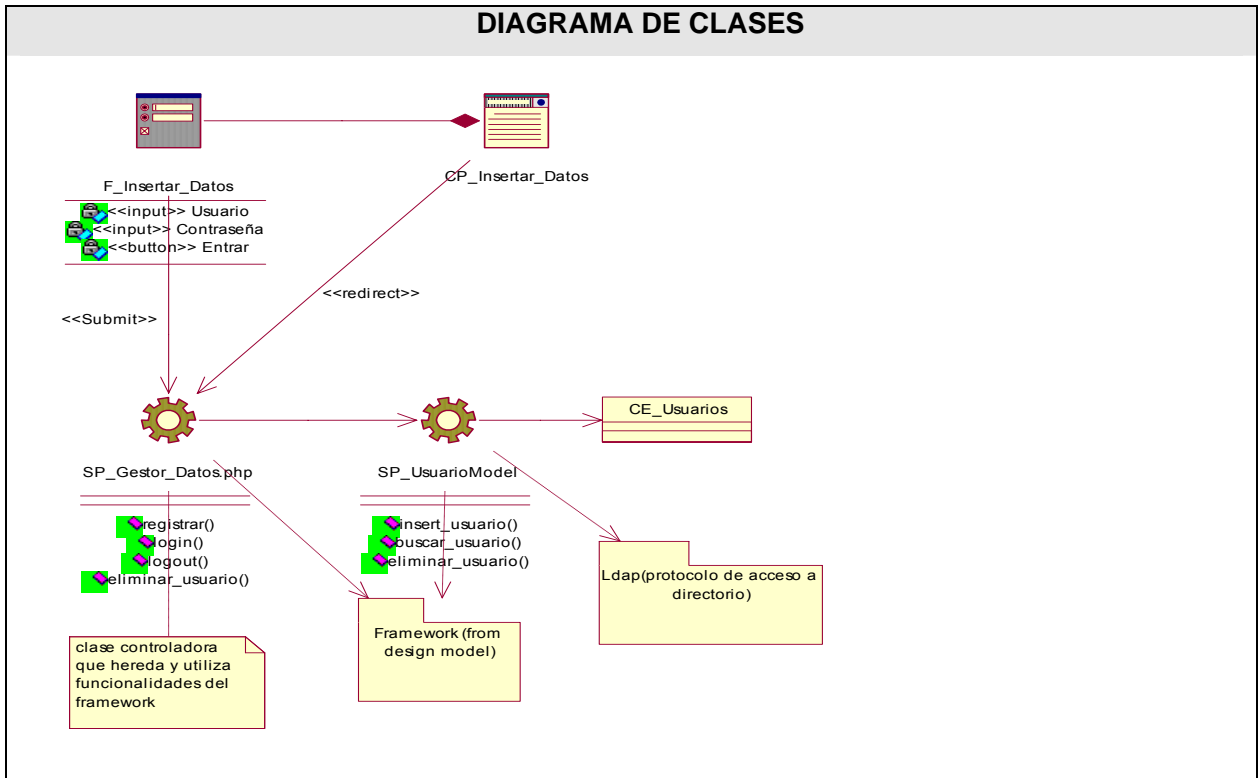


Tabla 45. Caso de uso "Autenticarse".

3.3.2.5 Caso de uso Administración

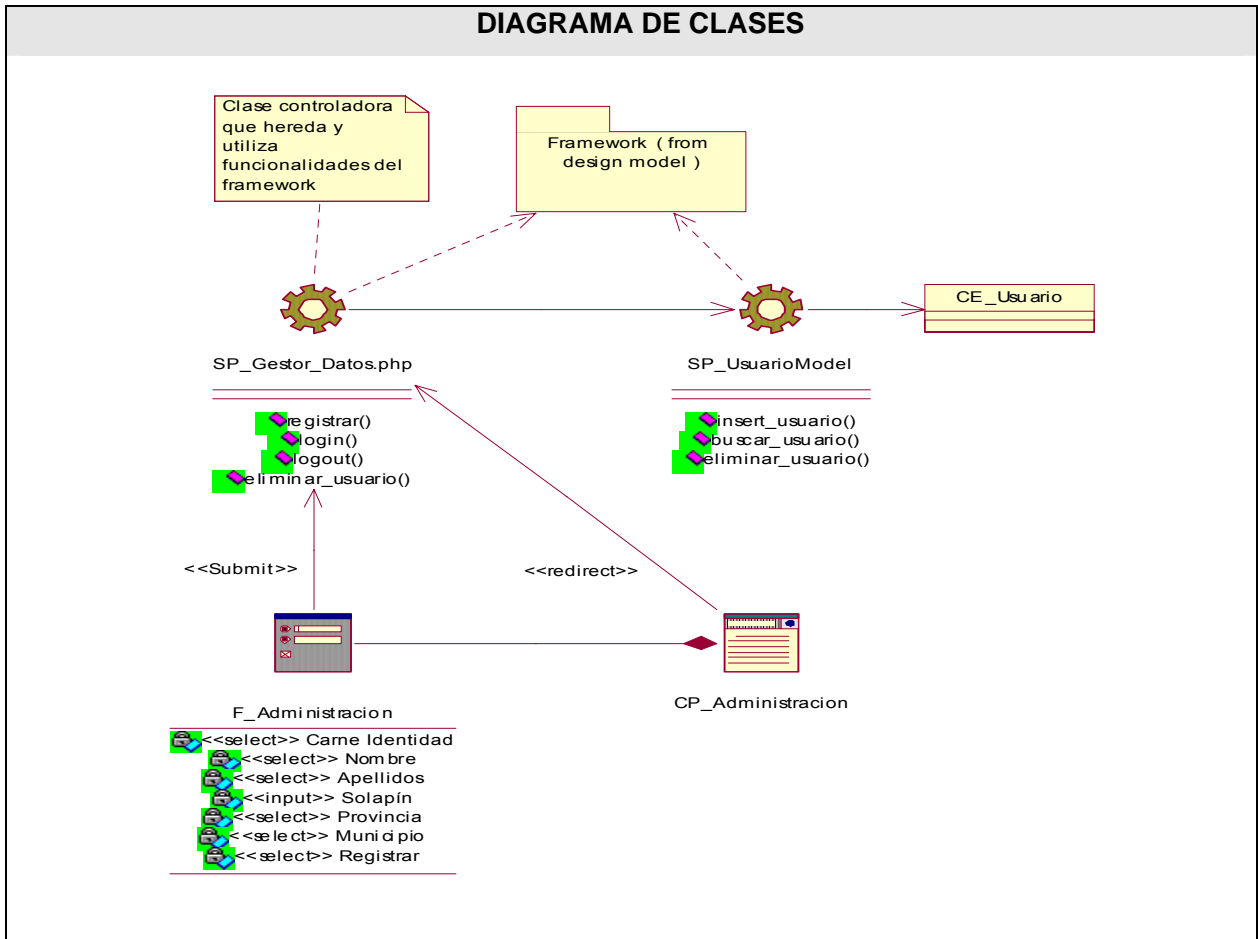


Tabla 46. Caso de uso "Administración".

3.4 Diseño de la base de datos.

3.4.1 Modelo lógico de datos (diagrama de clases persistentes).

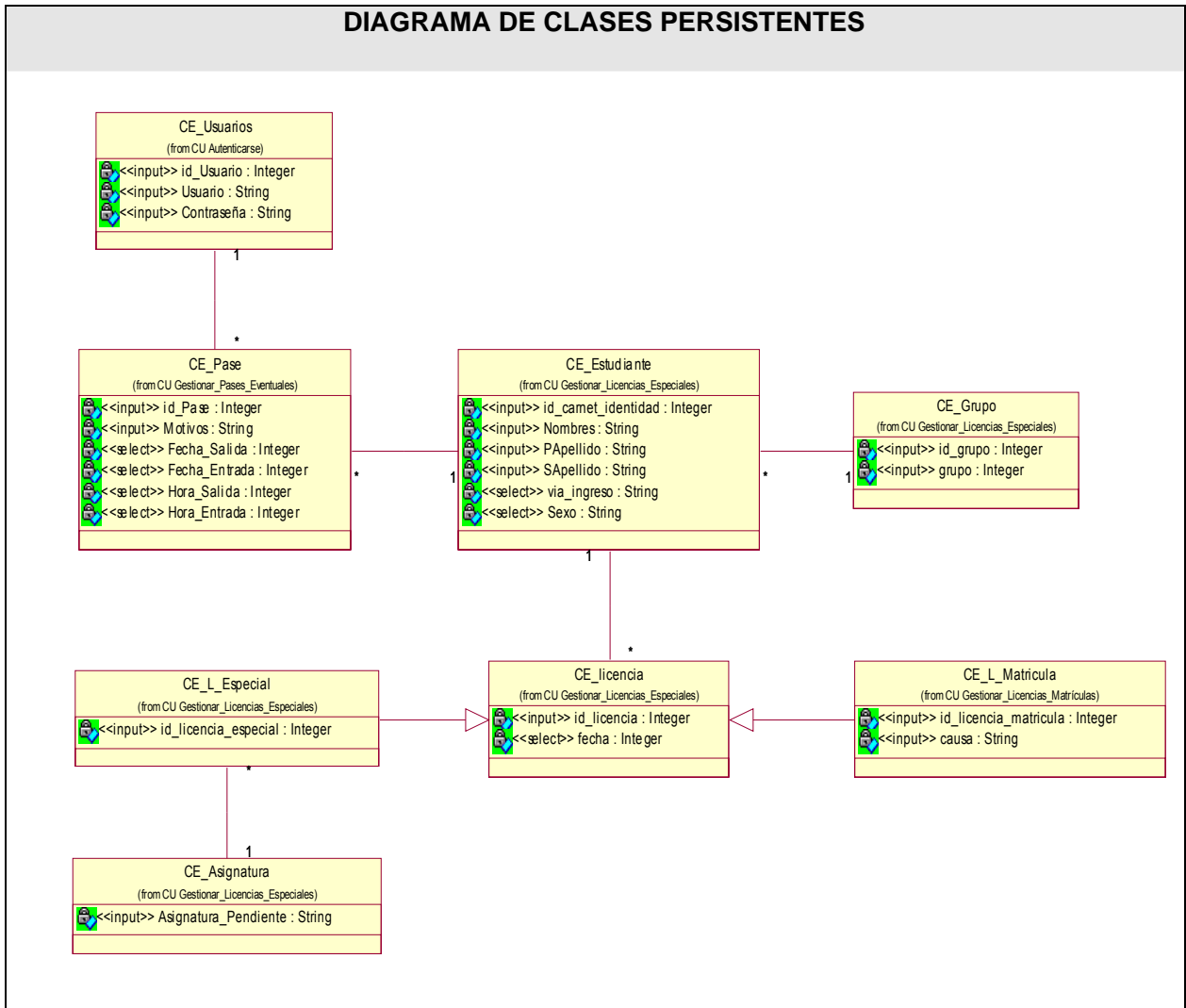


Tabla 47. Diagrama de clases persistentes.

3.4.2 Modelo físico de datos (modelo de datos).





### **3.5 Conclusiones**

Un proyecto de desarrollo de un Sistema de Información comprende varios componentes o pasos llevados a cabo durante la etapa del análisis, el cual ayuda a traducir las necesidades del cliente en un modelo de sistema que utiliza uno más de los componentes: Software, hardware, personas, base de datos, documentación y procedimientos.

El análisis y diseño de Sistemas, es el proceso de estudiar su situación con la finalidad de observar como trabaja y decidir si es necesario realizar una mejora.

Existen varios modelos o métodos para la realización del análisis y diseño de un sistema, lo primero del trabajo fue revisar que es el Análisis y el diseño y posteriormente presentar varios modelos que podemos utilizar para la realización y elaboración de un proceso y trabajo exhaustivo y dar solución o respuesta al problema que se ha generado.

## **Capítulo 4 Implementación.**

### **4.1 Introducción**

La implementación es la fase en la que se acometen todas las actividades planificadas. Antes de la implementación de un proyecto, los implementadores deben identificar sus puntos fuertes y débiles, oportunidades y obstáculos.

En esta fase de implementación es importante una supervisión que asegure que el proyecto se desarrolla según lo previsto.

Los plazos de implementación deben contar con evaluaciones periódicas que permitan detectar a tiempo problemas, debilidades y limitaciones e iniciar procesos formales para su corrección a tiempo. Algunos desarrolladores no ven con buenos ojos la introducción de cambios en el proyecto una vez iniciado el proceso de implementación, argumentando que introducirá retrasos e impactará el resto del proyecto.

Es por ello que debe definirse un protocolo formal (así como recursos y equipos de trabajo) para la revisión, corrección y adaptación de herramientas y sistemas en la fase de implementación sin que se produzca un impacto negativo significativo en la ejecución global del proyecto.

### **4.2 Diagrama de despliegue.**

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un mainframe. Las instancias de componentes software pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz).

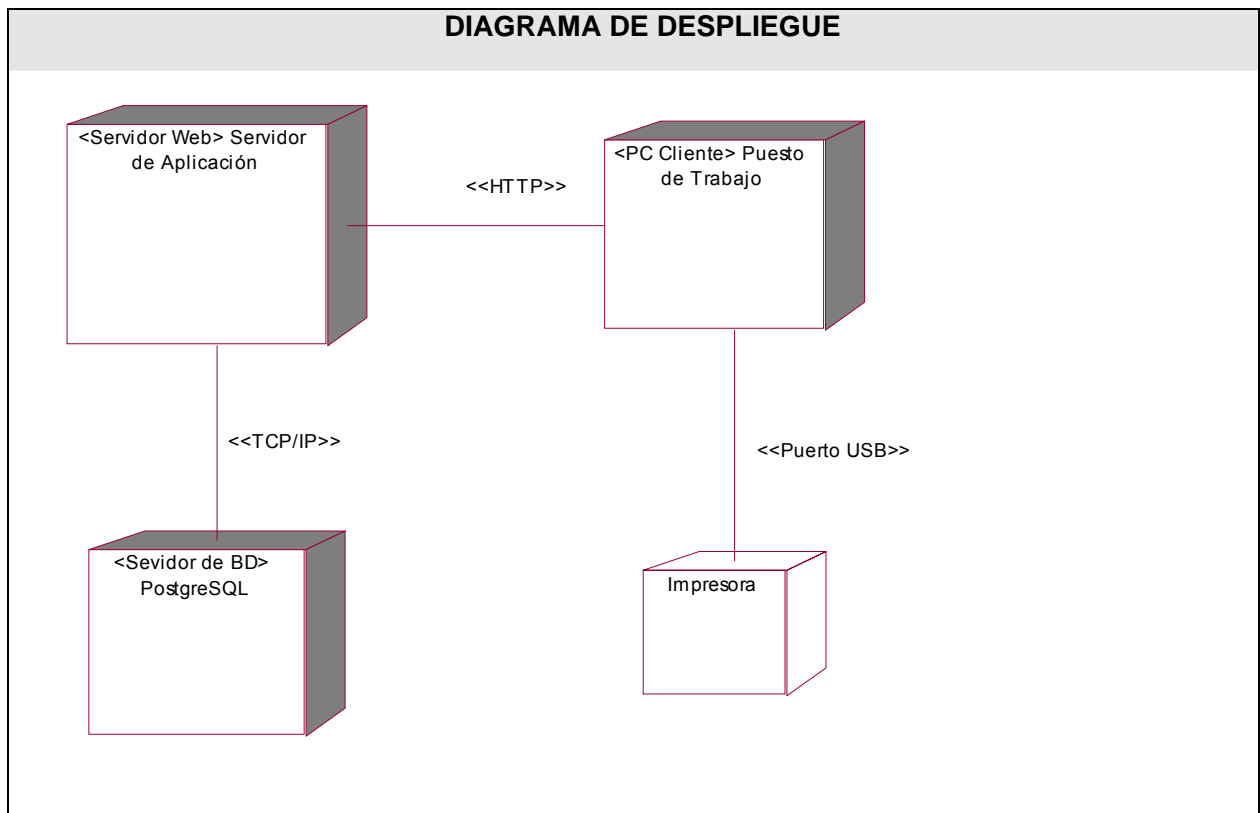


Tabla 49. Diagrama de despliegue.

### 4.3 Diagrama de componentes.

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

Dado que los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso.

Un paquete en un diagrama de componentes representa una división física del sistema. Los paquetes se organizan en una jerarquía de capas donde cada capa tiene una interfaz bien definida. Un ejemplo típico de una jerarquía en capas de este tipo es: Interfaz de usuario; Paquetes específicos de la aplicación; Paquetes reusables; Mecanismos claves; y Paquetes hardware y del sistema operativo.

Un diagrama de componentes se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (generalmente de compilación). Puede mostrar también que un componente software contiene una interfaz, es decir, la soporta.

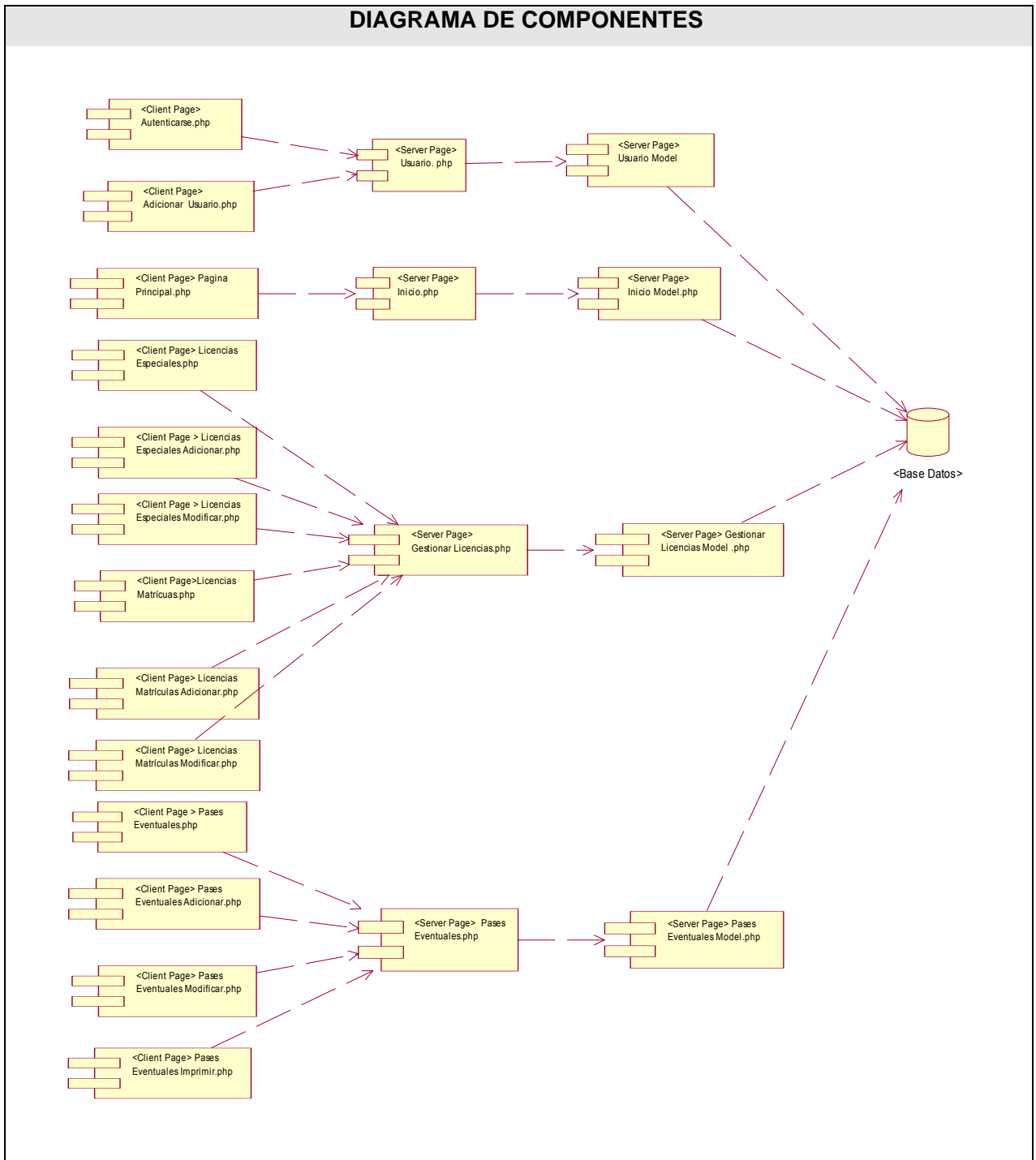


Tabla 50. Diagrama de componentes de código ejecutable.

#### 4.4 Diagrama de Componentes de BD.

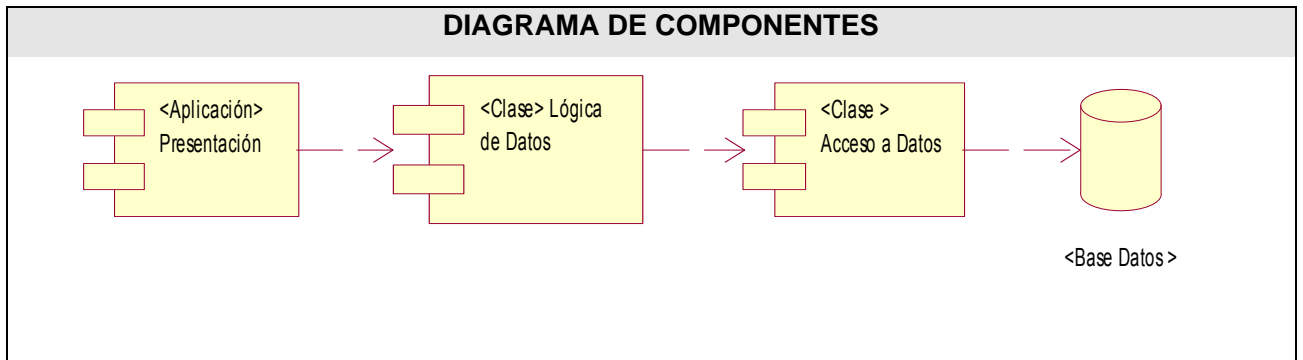


Tabla 51. Diagrama de componentes de base de datos

## Capítulo 5 Estudio de Factibilidad.

### 5.1 Introducción

La estimación del coste y del esfuerzo del software nunca será una ciencia exacta. Son demasiadas las variables humanas, técnicas, de entorno, políticas que pueden afectar al coste final del software y al esfuerzo aplicado para desarrollarlo. La estimación del proyecto de software puede dejar de ser un oscuro arte para convertirse en una serie de pasos sistemáticos que proporcionen estimaciones con un grado de riesgo aceptable.

### 5.2 Planificación basada en casos de uso.

#### Paso 1. Cálculo de los Puntos de casos de uso sin ajustar.

Se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

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.

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0

Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	3	9
Total			3	9

**Tabla 52. Factor de peso de los actores sin ajustar.**

$$UAW = \sum cant \ actores * peso$$

$$UAW = 9$$

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

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la tabla siguiente:

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	1	5
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	3	30
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
Total			4	35

**Tabla 53. Factor de peso de los casos de usos sin ajustar**



$$UUCW = \sum cantCU * peso$$

$$UUCW = 35$$

$$UUCW = 9 + 35$$

$$UUCW = 44$$

**Paso 3. Cálculo de los Puntos de casos de uso ajustados.**

$$UCP = UUCP * TCF * EF$$

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.

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	3	3
T4	Funcionamiento Interno complejo	1	2	2
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0,5	3	1,5
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	5	10

T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
Sumatoria			36.5	

**Tabla 54. Factor de Complejidad Técnica.**

$$TCF = 0.6 + 0.01 * \sum (\text{peso} * \text{valor\_asignado})$$

$$TCF = 0.6 + 0.01 * 36.5$$

$$TCF = 0.6 + 0.365$$

$$TCF = 0.965$$

**Paso 4. Cálculo del Factor de Ambiente (EF).**

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.

Factor	Descripción	Peso	Valor asignado	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	0	0
E3	Experiencia en la orientación a objetivos.	1	2	2
E4	Capacidad del analista líder.	0,5	5	2,5
E5	Motivación.	1	5	5
E6	Estabilidad de requerimientos	2	5	10
E7	Personal Part–Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	4	-4
Sumatoria				20

**Tabla 55. Factor de Ambiente.**

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{peso} * \text{valor\_asignado})$$

$$EF = 1.4 - 0.03 * 20$$

$$EF = 1.4 - 0.6$$

$$EF = 0.8$$

$$UCP = UUCP * TCF * EF$$

$$UCP = 44 * 0.965 * 0.8$$

$$UCP = 33.968$$

#### **Paso 5. Estimación de esfuerzo a través de los puntos de casos de uso.**

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas hombres.

UCP: Punto de casos de usos ajustados.

CF: Factor de conversión.

Para obtener el factor de conversión (CF) se cuentan cuantos valores de los que afectan el factor ambiente (E1...E6) están por debajo de la media (3), y los que están por arriba de la media para los restantes (E7, E8). Si el total es 2 o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso. Si el total es 3 o 4 se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso. Si el total es mayor o igual que 5 se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

En este caso se puede decir que:

CF = 28 Horas-Hombre / Punto de Casos de uso.

$$E = 33.968 * 28$$

$$E = 951.104$$

**Paso 6. Calcular esfuerzo del proyecto.**

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	237.776
Diseño	20	475.552
Implementación	40	951.104
Pruebas	15	356.664
Sobrecarga (otras actividades)	15	356.664
Total	100	2377.76

**Tabla 56. Esfuerzo del Proyecto.**

Con éste criterio, y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Si  $E_T = 2377.76$  horas-hombre y se estima que cada mes tiene como promedio 176 horas laborables, eso daría un  $E_T = 13.51$ mes-hombre.

Esto quiere decir que 1 persona puede realizar el problema analizado en 13 meses y medio aproximadamente.

Costo del Proyecto.

El salario promedio mensual es de \$100

CH: Cantidad de hombres.

Tiempo: Tiempo total del proyecto.

CH = 2 hombres

CHM = 2 \* Salario Promedio

CHM = 200.00 \$/mes

Costo = CHM \*  $E_T$  / CH

Costo = 200.00 \* 13.51 / 2

Costo = \$ 1351.00

Tiempo =  $E_T$  / CH

Tiempo =  $13.51 / 2$

Tiempo =  $6.755 \approx 6.8$  meses

Haciendo un análisis de los resultados que se obtuvieron se puede decir que con los 2 hombres que están trabajando en el proyecto este debe desarrollarse en 6.8 meses y su costo total esta estimado para \$ 1351.00

### **5.3 Beneficios tangibles e intangibles.**

El primer paso en este proceso es determinar la rentabilidad o beneficios que el nuevo sistema va a traernos. Los beneficios obtenidos pueden ser de 2 tipos – tangibles e intangibles. El sistema automatizado para la gestión de información de la secretaría docente sus beneficios inmediatos son intangibles.

Los beneficios tangibles son aquellos que están asociados con ahorros monetarios. Los beneficios intangibles en un sistema nuevo afecta no solo en lo externo sino que también en lo interno en términos de informes de resultados más rápidos y frecuentes además de la presión, la velocidad y la flexibilidad del equipo automático de procesamiento de datos le permite a la administración tener más tiempo para planear y organizar las actividades y así dirigir y controlar de acuerdo con el plan original. También tenemos el acceso a las mejores prácticas, mejora significativa en la calidad de los datos, flexibilidad del sistema, programa completo y actualizado, reducción de errores en la entrada de datos y rapidez en los reportes estadísticos finales.

### **5.4 Análisis de costos y beneficios.**

En el principio el costo del Software constituía un pequeño porcentaje del costo total de los sistemas basados en computadoras. Hoy en día el Software es el elemento más caro de la mayoría de los sistemas informáticos.

Un gran error en la estimación del costo puede ser lo que marque la diferencia entre beneficios y perdidas, la estimación del costo y del esfuerzo del software nunca será una ciencia exacta, son demasiadas las variables: humanas, técnicas, de entorno, políticas, que pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo.

Existen diferentes modelos de estimación como son:

1. Los modelos empíricos
2. El modelo cocomo.
3. Herramientas automáticas de estimación.

Para obtener estimaciones exactas para un proyecto, generalmente se utilizan al menos dos de las tres técnicas referidas anteriormente. Mediante la comparación y la conciliación de las estimaciones obtenidas con las diferentes técnicas, el planificador puede obtener una estimación más exacta. La estimación del proyecto de software nunca será una ciencia exacta, pero la combinación de buenos datos históricos y técnicas puede mejorar la precisión de la estimación.

## **5.5 Conclusiones**

El análisis de costos y beneficios constituye una ayuda importante en la toma de decisiones, ayuda que frecuentemente brinda la información necesaria para determinar si la actividad es deseable, o si, por el contrario viene a constituir un desperdicio.

En términos generales, el análisis de costos y beneficios es un instrumento para desarrollar en forma sistemática una información útil acerca de los efectos deseables e indeseables de los programas o proyectos

El análisis de costos involucra tanto costo y beneficio como costo y efectividad. Los costos y beneficios siempre deben ser considerados juntos.

## **Conclusiones**

El software se ha convertido en el elemento clave de la evolución de los sistemas y productos informáticos. El software se compone de programas, datos y documentos que de alguna forma han quedado reflejados en este documento. El componente de software deberá diseñarse e implementarse para que pueda volver a ser reutilizado en muchos programas diferentes. Un programa que funcione es solo una parte de una configuración del software que incluye muchos elementos. La documentación proporciona el fundamento para un buen desarrollo y, lo que es más importante, proporciona guías para la tarea del mantenimiento del software.

## **Recomendaciones**

Recomendamos:

1. Poner a prueba el sistema durante un período de tiempo significativo, para comprobar su desempeño y que las funcionalidades del sistema se correspondan con la realidad de lo que se desea.
2. Continuar el estudio con el objetivo de añadir nuevas funcionalidades.
3. Proponer, tras corroborar un desempeño exitoso, la utilización y generalización de este sistema en la UCI.



## Referencias Bibliográficas

- [1] Lenguaje de programación. In, 2007.  
<http://geeks.ms/blogs/sergiotarrillo/archive/2007/01/09/8420.aspx>
- [2] Servidor Web. In, 2007. [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)
- [3] M.A. Alvarez, Qué es PHP. In, 2007. <http://www.desarrolloweb.com/articulos/392.php>
- [4] W.U. Ayllapan, Planificación, proyección y diseño de páginas Web. Aspectos generales In, 2006. <http://www.desarrollodeweb.com.ar/planificacion-proyeccion-diseno-de-paginas-web-aspectos-generales.htm>
- [5] L.F.C. Calle, La tecnología informática y la educación. In, 2007.  
[http://www.wikilearning.com/la\\_tecnologia\\_informatica\\_y\\_la\\_educacion-wkccp-20281-1.htm](http://www.wikilearning.com/la_tecnologia_informatica_y_la_educacion-wkccp-20281-1.htm)
- [6] R.A. Fonseca, FAQ de postgreSQL In, 2005. <http://www.postgresql.org.mx/?q=node/8>
- [7] R. Good, Ajax: Un Abordaje A Mitad-De-Camino In, 2005.  
[http://www.masternewmedia.org/es/2005/07/28/ajax\\_el\\_nuevo\\_abordaje\\_de.htm](http://www.masternewmedia.org/es/2005/07/28/ajax_el_nuevo_abordaje_de.htm)
- [8] M. Mendez, Qué es oracle? In, 2007.  
<http://www.infranetworking.net/foros/index.php?showtopic=462&hl=oracle>
- [9] H.H. Perez, Modelo Cliente Servidor. In, 2001.  
<http://sipan.inictel.gob.pe/users/herrera/mcse.htm>
- [10] P. Pozo, Diagrama de Clases de Diseño. In, 2005.  
<http://www.clikear.com/manuales/uml/faseconstruccionbajonivel.asp>
- [11] P. Pozo, Diagramas de Interacción In, 2005.  
<http://www.clikear.com/manuales/uml/diagramasinteraccion.asp>
- [12] G.J. Reynoso, Manual del usuario de PostgreSQL. In, 2003. <http://es.tldp.org/Postgresql-es/web/navegable/user/app-pgadmin.html>
- [13] M.A.M. Sánchez, ¿Qué metodología debo usar para el desarrollo de un Software? In, 2004.  
[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
- [14] S. Tarrillo, Mi turno de hablar de AJAX, ventajas y desventajas. In, 2007.  
<http://geeks.ms/blogs/sergiotarrillo/archive/2007/01/09/8420.aspx>

[15] A.F. Vilas, Diagrama de componentes. In, 2001. <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>

[16] A.F. Vilas, Diagrama de despliegue. In, 2001. <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>

## Anexos

### Anexo 1 Descripción textual del caso de uso del negocio

#### Gestionar Pases Eventuales.

<b>Caso de uso del negocio: Gestionar Pases Eventuales</b>	
<b>Actores del negocio:</b> Estudiante (inicia)	
<b>Propósito:</b> Solicitar al personal autorizado un pase eventual	
<b>Resumen:</b> El caso de uso se inicia cuando el estudiante se presenta donde está la decana o los vicedecanos a solicitar un pase eventual. Después de informar los motivos que tiene para necesitar un pase eventual el responsable valora si los motivos son lo suficiente sólidos en caso afirmativo le pide al estudiante su solapín para registrar sus datos personales y la fecha de salida y entrada. El responsable llena el modelo de pase eventual, se lo entrega al estudiante y este se retira. En caso negativo el estudiante se retira sin el pase eventual.	
<b>Curso normal de los eventos:</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1-Solicita Pase Eventual exponiendo los motivos del mismo.	2-Analiza los argumentos del estudiante.
	3- Solicita el solapín del estudiante.
4- Entrega el solapín.	5- Registra los datos del estudiante y el periodo por el cual le dio el pase eventual.
	6-Elabora el modelo de pase y se lo entrega al estudiante.
6-El estudiante se retira	
<b>Flujo alternativo de los eventos:</b>	
<b>Acción del actor:</b>	<b>Respuesta del proceso de</b>

	<b>negocio:</b>
	2.1- Le informa al estudiante que no se le puede dar un pase eventual
1.1- El estudiante se retira	
<b>Prioridad:</b> Crítico	
<b>Mejoras:</b>	
<b>Otras secciones:</b>	

Tabla 4.Caso de uso del negocio "Gestionar Pases Eventuales".

#### Gestionar Licencias de Matrículas.

<b>Caso de uso del negocio: Gestionar Licencias de Matrículas</b>	
<b>Actores del negocio:</b> Estudiante (inicia)	
<b>Propósito:</b> Solicitar a la dirección de la facultad una licencia de matrícula	
<b>Resumen:</b> El caso de uso se inicia cuando el estudiante presenta a la decana una solicitud por escrito expresando los motivos que tiene para pedir una licencia. La decana analiza la solicitud y si son convincentes autoriza la licencia, en caso contrario el estudiante se marcha.	
<b>Curso normal de los eventos:</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1-El estudiante solicita por escrito la licencia de matrícula	3-Analiza los argumentos del estudiante.
	4-Entregar modelo de tramitación de bajas.
5-El estudiante realiza los pasos indicados en el modelo.	
6-Entrega el modelo	
	7-Verifica que el documento este correcto.

	8-Llena el modelo SG-3
	9-Archiva los documentos(solicitud licencia de matricula, modelo de tramitación de bajas y modelo SG-3)
	10-Entrega una copia del modelo SG-3
11-El estudiante se retira.	
<b>Flujo alternativo de los eventos:</b>	
<b>Acción del actor:</b>	<b>Respuesta del proceso de negocio:</b>
	3.1-Le informa al estudiante que no se le puede otorgar la licencia.
1.1- El estudiante se retira	
<b>Prioridad:</b> Crítico	
<b>Mejoras:</b>	
<b>Otras secciones:</b>	

Tabla 5. Caso de uso del negocio “ Gestionar Licencias de Matrículas”.

### Gestionar Licencias Especiales.

<b>Caso de uso del negocio: Gestionar Licencias Especiales</b>	
<b>Actores del negocio:</b> Estudiante (inicia)	
<b>Propósito:</b> Otorgar a los estudiantes que no aprueben un mundial el derecho de seguir siendo matrícula de la universidad.	
<b>Resumen:</b> El caso de uso se inicia cuando el estudiante suspende uno o dos mundiales y se reúne una comisión para decidir cuales de los estudiantes que suspendieron un mundial tiene el derecho de seguir siendo matricula de la escuela y cuales pasan a ser baja.	
<b>Curso normal de los eventos:</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de</b>

	<b>negocio</b>
1-El estudiante suspende uno o dos mundiales.	2-Analizan la trayectoria del estudiante durante los cursos anteriores.
	3-Deciden si le van a otorgar la licencia especial,
	4-Pasan la información a la vicerrectora de formación y a la secretaria docente.
	5- Le informan a los estudiantes que se le otorgó la licencia especial y las condiciones que tienen que cumplir.
6- El estudiante se presenta en secretaria y solicita las asignaturas que puede recibir durante el curso.	
	7- Consultan los planes de estudio y confeccionan un listado de asignaturas que el estudiante puede adelantar.
8- El estudiante selecciona solo 3 asignaturas por semestre y se retira.	
<b>Flujo alterno de los eventos:</b>	
<b>Acción del actor:</b>	<b>Respuesta del proceso de negocio:</b>
	3.1- Le informan al estudiante que no le otorgaron la licencia especial.
	3.2-Tramitan la baja del estudiante.
1.1- El estudiante se retira	
<b>Prioridad:</b> Crítico	

Mejoras:
Otras secciones:

Tabla 6. Caso de uso del negocio “Gestionar Licencias Especiales”.

## Anexo 2 Diagrama de actividad de los casos de uso del negocio

### Gestionar Pases Eventuales.

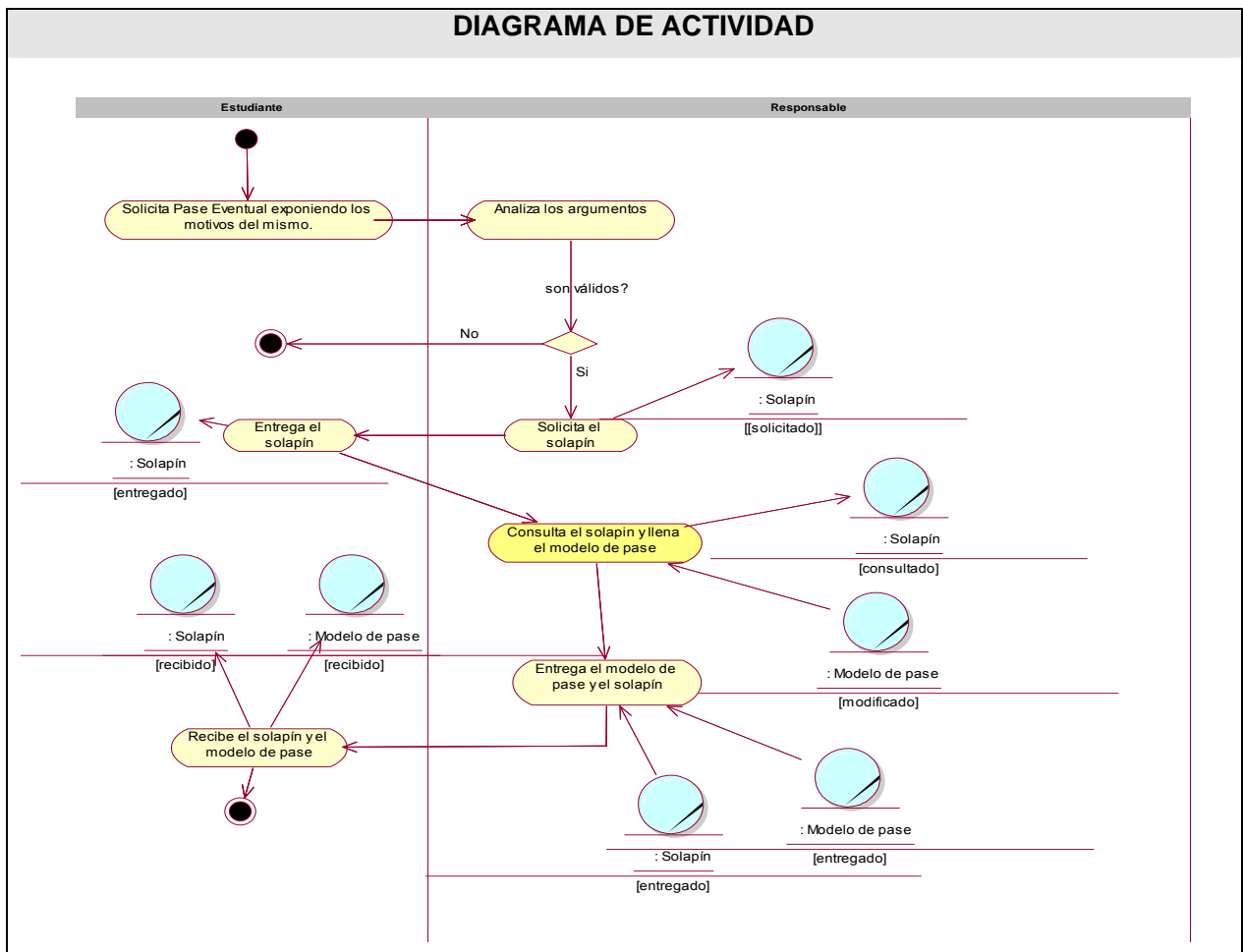


Tabla 7. Caso de uso del negocio “Gestionar Pases Eventuales”.

### Gestionar Licencias de Matriculas.

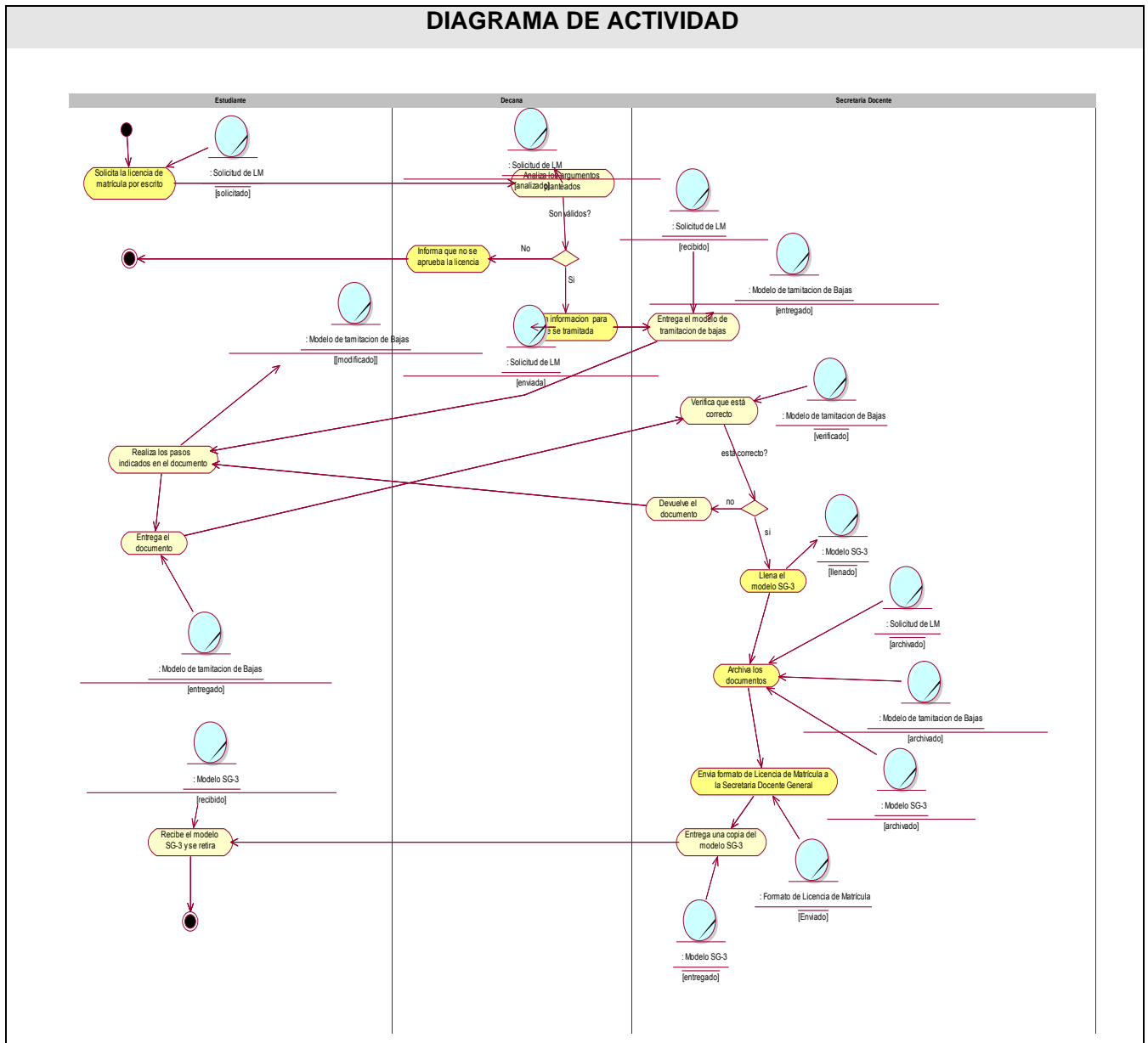


Tabla 8. Caso de uso del negocio “Gestionar Licencias de Matrículas”.

**Gestionar Licencias Especiales.**



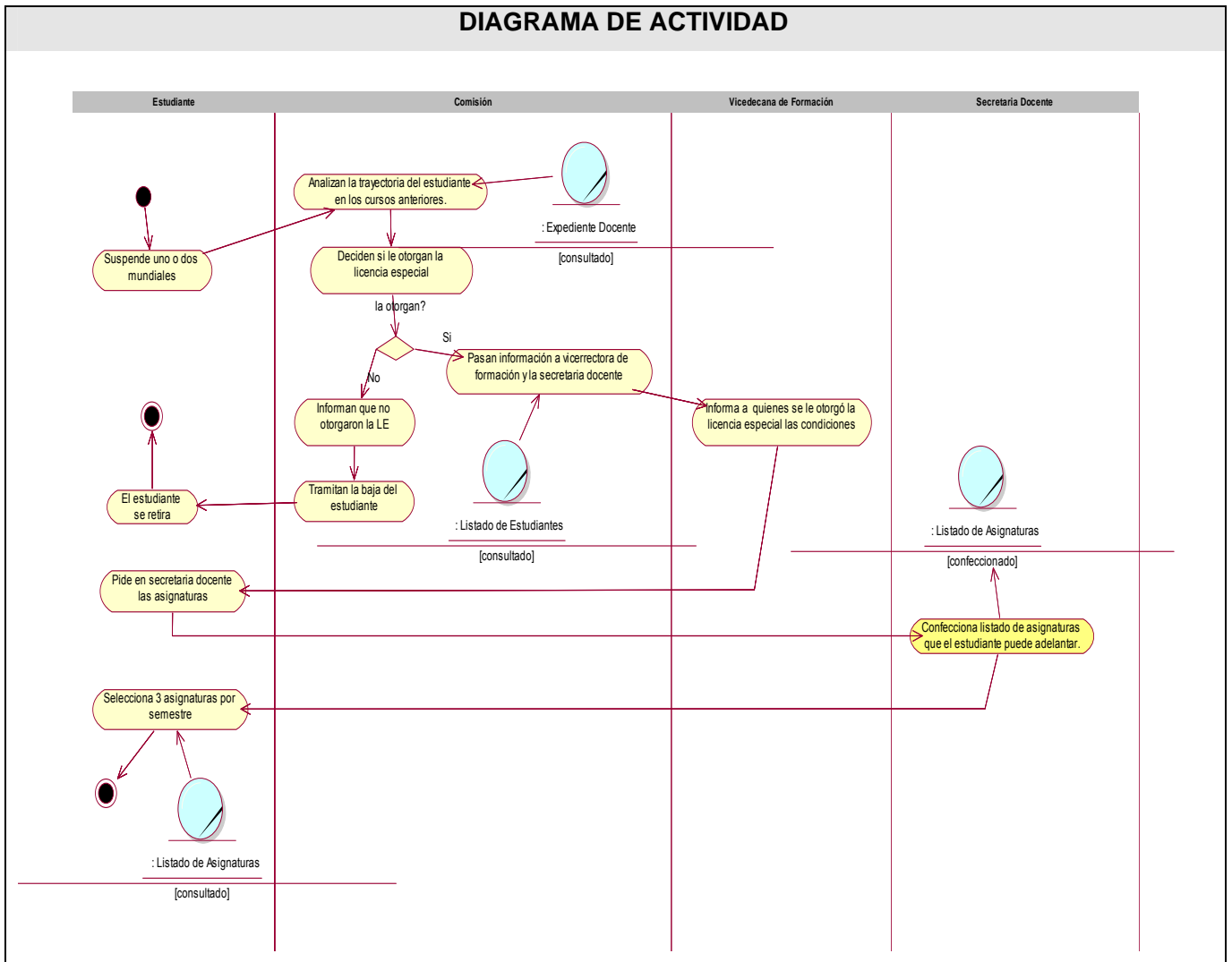


Tabla 8. Caso de uso del negocio “Gestionar Licencias Especiales”.

Anexo3 Modelo de objetos.

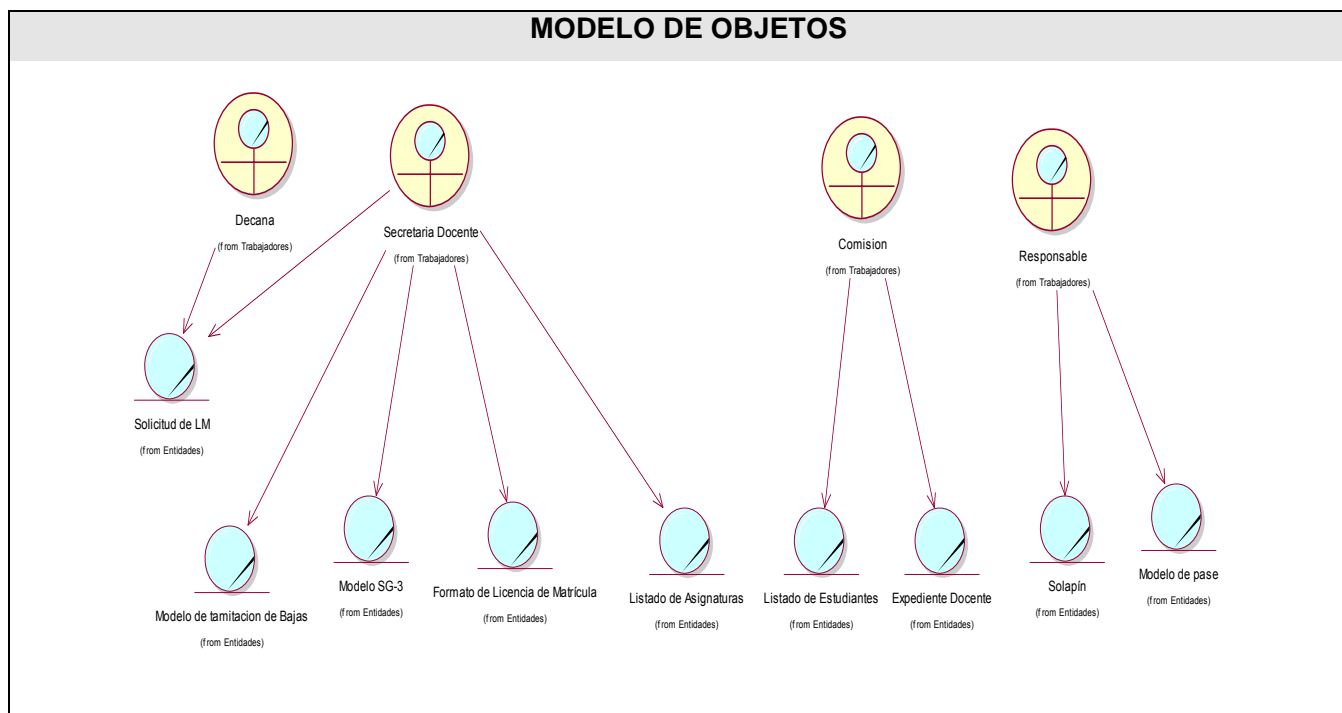


Tabla 9. Modelo de Objetos.

## Anexo 4 Descripción textual de los casos de uso del sistema.

### Gestionar pases eventuales.

<b>Caso de uso</b>	Gestionar pases eventuales
<b>Propósito</b>	Permite insertar, eliminar, modificar estudiantes y listar los estudiantes de pase dada una fecha determinada.
<b>Actores</b>	Responsable
<b>Resumen:</b>	El caso de uso se inicia cuando el responsable accede al sistema para insertar, modificar o eliminar los datos del estudiante o para obtener un listado de los estudiantes que están de pase dada una fecha determinada.
<b>Referencias</b>	R1

<b>Precondiciones</b>	El usuario debe introducir su usuario y contraseña.	
<b>Poscondiciones</b>	Según los privilegios del usuario se habilitan las funcionalidades.	
<b>Curso Normal de los Eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
El responsable necesita insertar, modificar o eliminar estudiantes	<p>1.1- El sistema ejecuta las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- Si decide insertar un estudiante, ir a la sección "Insertar estudiante "</li> <li>- Si decide modificar un estudiante, ir a la sección "Modificar estudiante "</li> <li>- Si decide eliminar un estudiante, ir a la sección "Eliminar estudiante "</li> </ul>	
<b>Sección "Insertar estudiante "</b>		
2-El responsable introduce los datos del estudiante.	<p>2.1- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.</p> <p>2.2- Se almacenan los datos en el sistema.</p> <p>2.3- Se muestra un mensaje informando al responsable que los datos del estudiante se agregaron satisfactoriamente, dando la posibilidad de insertar datos nuevos y finaliza así el caso de uso.</p>	
<b>Curso Alternativo de los Eventos</b>		
	2.1- Se muestra un mensaje de error informando que faltan campos o que hay errores en los datos.	

<b>Sección "Modificar estudiante "</b>	
3- El responsable selecciona el estudiante que quiere modificar.	<p>3.1- El sistema muestra la interfaz con los datos del estudiante que se pueden modificar.</p> <p>3.2- El sistema da la posibilidad de modificar los datos.</p>
4- El responsable registra los datos modificados	<p>4.1- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.</p> <p>4.2- Se almacenan los datos en el sistema.</p> <p>4.3- Se muestra un mensaje informando al responsable que los datos del estudiante se modificaron satisfactoriamente y finaliza así el caso de uso.</p>
<b>Curso Alternativo de los Eventos</b>	
	4.1- Se muestra un mensaje de error informando que faltan campos o que hay errores en los datos.
<b>Sección "Eliminar estudiante "</b>	
5- El responsable selecciona el estudiante que quiere eliminar.	5.1- El sistema muestra la interfaz con los datos del estudiante que se quiere eliminar.
6- El responsable elimina los datos mostrados	6.1- El sistema da la posibilidad de eliminar los datos.

	6.2- Se eliminan los datos en el sistema. 6.3- Se muestra un mensaje informando al responsable que los datos del estudiante se eliminaron satisfactoriamente y finaliza así el caso de uso.
<b>Curso Alternativo de los Eventos</b>	
	6.1- Se muestra un mensaje para verificar si esta de acuerdo con eliminar esos datos.
<b>Prioridad</b>	Crítico

Tabla 12. Caso de uso del sistema “Gestionar Pases Eventuales”.

#### Gestionar licencias de matrículas.

<b>Caso de uso</b>	Gestionar licencias de matrículas.
<b>Propósito</b>	Permite insertar, modificar y eliminar datos de los estudiantes.
<b>Actores</b>	Secretaria Docente
<b>Resumen:</b> El caso de uso se inicia cuando la secretaria docente accede al sistema para insertar, modificar o eliminar datos de los estudiantes.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe introducir su usuario y contraseña.
<b>Poscondiciones</b>	Según los privilegios del usuario se habilitan las funcionalidades.
<b>Curso Normal de los Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- La secretaria docente necesita insertar, modificar o eliminar datos de los estudiantes que solicitan licencia de matrícula.	1.1- El sistema ejecuta las siguientes acciones:  - Si decide insertar un estudiante, ir a la sección “ Insertar estudiante ”  - Si decide modificar un estudiante, ir a

	<p>la sección “Modificar estudiante “</p> <p>- Si decide eliminar un estudiante, ir a la sección “ Eliminar estudiante “</p>
<b>Sección “Insertar estudiante ”</b>	
<p>2- La secretaria docente introduce los datos del estudiante.</p>	<p>2.1- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.</p> <p>2.2- Se almacenan los datos en el sistema.</p> <p>2.3- Se muestra un mensaje informando a la secretaria docente que los datos del estudiante se agregaron satisfactoriamente, dando la posibilidad de insertar datos nuevos y finaliza así el caso de uso.</p>
<b>Curso Alternativo de los Eventos</b>	
	<p>2.1- Se muestra un mensaje de error informando que faltan campos o que hay errores en los datos.</p>
<b>Sección “Modificar estudiante “</b>	
<p>3- La secretaria docente selecciona el estudiante que quiere modificar.</p>	<p>3.1- El sistema muestra la interfaz con los datos del estudiante que se pueden modificar.</p>
<p>4- La secretaria docente cambia los datos.</p>	<p>4.1- El sistema da la posibilidad de modificar los datos.</p> <p>4.2- El sistema verifica que los datos introducidos sean correctos y que no</p>

	<p>existan campos vacíos.</p> <p>4.3- Se almacenan los datos en el sistema.</p> <p>4.4- Se muestra un mensaje informando a la secretaria docente que los datos del estudiante se modificaron satisfactoriamente y finaliza así el caso de uso.</p>
<b>Curso Alternativo de los Eventos</b>	
	4.2- Se muestra un mensaje de error informando que faltan campos o que hay errores en los datos.
<b>Sección “Eliminar estudiante “</b>	
5- La secretaria docente selecciona la opción de eliminar un estudiante.	5.1- El sistema muestra la interfaz con los estudiantes que se pueden eliminar.
6- La secretaria docente selecciona el estudiante que quiere eliminar.	<p>6.1- El sistema da la posibilidad de eliminar los datos.</p> <p>6.2- Se eliminan los datos en el sistema.</p> <p>6.3- Se muestra un mensaje informando a la secretaria docente que los datos del estudiante se eliminaron satisfactoriamente y finaliza así el caso de uso.</p>
<b>Curso Alternativo de los Eventos</b>	
	6.1- Se muestra un mensaje para verificar si esta de acuerdo con eliminar esos datos.
<b>Prioridad</b>	Crítico

Tabla 13. Caso de uso del sistema “Gestionar Licencias de Matrículas”.

**Gestionar licencias especiales.**

<b>Caso de uso</b>	Gestionar licencias especiales.	
<b>Propósito</b>	Permite insertar, modificar y eliminar datos de los estudiantes.	
<b>Actores</b>	Secretaria docente	
<b>Resumen:</b>	El caso de uso se inicia cuando la secretaria docente accede al sistema para insertar, modificar o eliminar datos de los estudiantes.	
<b>Referencias</b>	R6	
<b>Precondiciones</b>	El usuario debe introducir su usuario y contraseña.	
<b>Poscondiciones</b>	Según los privilegios del usuario se habilitan las funcionalidades.	
<b>Curso Normal de los Eventos</b>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
1- La secretaria docente necesita insertar, modificar o eliminar estudiantes.	<p>1.1- El sistema ejecuta las siguientes acciones:</p> <ul style="list-style-type: none"> <li>- Si decide insertar un estudiante, ir a la sección "Insertar estudiante "</li> <li>- Si decide modificar un estudiante, ir a la sección "Modificar estudiante "</li> <li>- Si decide eliminar un estudiante, ir a la sección "Eliminar estudiante "</li> </ul>	
<b>Sección "Insertar estudiante "</b>		
2- La secretaria docente introduce los datos del estudiante.	<p>2.1- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos.</p> <p>2.2- Se almacenan los datos en el sistema.</p> <p>2.3-Se muestra un mensaje informando a la secretaria docente que los datos del</p>	



	estudiante se agregaron satisfactoriamente, dando la posibilidad de insertar datos nuevos y finaliza así el caso de uso.
<b>Curso Alternativo de los Eventos</b>	
	2.1- Se muestra un mensaje de error informando que faltan campos o que hay errores en los datos.
<b>Sección "Modificar estudiante "</b>	
3- La secretaria docente selecciona la opción modificar.	3.1- El sistema muestra la interfaz con los estudiantes que se pueden modificar.
4- La secretaria docente selecciona el estudiante que quiere modificar.	4.1- El sistema da la posibilidad de modificar los datos. 4.2- El sistema verifica que los datos introducidos sean correctos y que no existan campos vacíos. 4.3- Se almacenan los datos en el sistema. 4.4- Se muestra un mensaje informando a la secretaria docente que los datos del estudiante se modificaron satisfactoriamente y finaliza así el caso de uso.
<b>Curso Alternativo de los Eventos</b>	
	4.2- Se muestra un mensaje de error

	informando que faltan campos o que hay errores en los datos.
<b>Sección “Eliminar estudiante “</b>	
5- La secretaria docente selecciona la opción de eliminar un estudiante.	5.1- El sistema muestra la interfaz con los estudiantes que se pueden eliminar.
6- La secretaria docente selecciona el estudiante que quiere eliminar.	6.1- El sistema da la posibilidad de eliminar los datos. 6.2- Se eliminan los datos en el sistema. 6.3- Se muestra un mensaje informando a la secretaria docente que los datos del estudiante se eliminaron satisfactoriamente y finaliza así el caso de uso.
<b>Curso Alternativo de los Eventos</b>	
	6.1- Se muestra un mensaje para verificar si esta de acuerdo con eliminar esos datos.
<b>Prioridad</b>	Crítico

Tabla 14. Caso de uso del sistema “Gestionar Licencias Especiales”.

#### Autenticarse.

<b>Caso de uso</b>	Autenticarse
<b>Propósito</b>	Verificar que todos los usuarios se registren antes de entrar al sistema.
<b>Actores</b>	Usuario
<b>Resumen:</b> El caso de uso se inicia cuando algún usuario quiere entrar al sistema pero para poder asignarle los privilegios tiene que autenticarse previamente.	
<b>Referencias</b>	R7
<b>Precondiciones</b>	EL usuario entre su usuario y contraseña.

<b>Poscondiciones</b>	Según los privilegios del usuario se habilitan las funcionalidades.
<b>Curso Normal de los Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El usuario decide entrar al sistema.	1.1- El sistema muestra la interfaz de autenticarse.
2- El usuario llena las casillas con el nombre de usuario y su contraseña correspondiente.	2.1- El sistema valida los datos entrados y ejecuta una de las siguientes acciones:  - Si los datos son correctos, ir a la sección "Correctos".  - Si los datos son incorrectos, ir a la sección "Incorrectos".
<b>Sección "Correctos"</b>	
	3.1- El sistema permite que el usuario acceda a sus funcionalidades y finaliza el caso de uso.
<b>Sección "Incorrectos"</b>	
	4.1- El sistema muestra un mensaje de fallo en la autenticación.
	4.2- El sistema procede nuevamente a mostrar la interfaz de autenticarse y finaliza el caso de uso.
<b>Prioridad</b>	Crítico

Tabla 15. Caso de uso del sistema "Autenticarse".

**Administración.**

<b>Caso de uso</b>	Administrar Usuarios
<b>Propósito</b>	Garantizar que cada Usuario se registre en el sistema posteriormente a su autorizo.
<b>Actores</b>	Responsable General
<b>Resumen:</b> El caso de uso se inicia cuando el Responsable General accede al sistema y solicita administrar usuarios, inmediatamente se visualiza una pantalla donde el sistema brinda la posibilidad de efectuar las operaciones de agregar usuarios. Finaliza una vez que el administrador realiza las operaciones.	
<b>Referencias</b>	R5
<b>Precondiciones</b>	Si se desea agregar un usuario debe existir la instancia del rol que va a ejercer.
<b>Poscondiciones</b>	Insertar: se crea una instancia de la clase usuario.
<b>Curso Normal de los Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El Responsable General decide administrar usuarios y selecciona la opción.	1.1 El sistema muestra una pantalla que da la posibilidad de agregar un usuario
2-El Responsable General introduce el solapín del usuario.	2.1 El sistema verifica que los datos introducidos sean válidos y que los requeridos no estén vacíos. 2.2 El sistema verifica que el usuario no exista. 2.3 El usuario se registra en el sistema. 2.4 El sistema muestra un mensaje comunicando que el usuario se agregó satisfactoriamente y finaliza así el caso

	de uso.
<b>Curso Alternativo de los Eventos</b>	
	2.1 Se emite un mensaje de error para que se llenen los campos obligatorios o se introduzcan correctamente los datos. 2.2 Se emite un mensaje informando de la existencia del usuario.
<b>Prioridad</b>	Crítico

Tabla 16. Caso de uso del sistema “Administración”.

Anexo 5 Diagramas de interacción.

Gestionar pases eventuales.

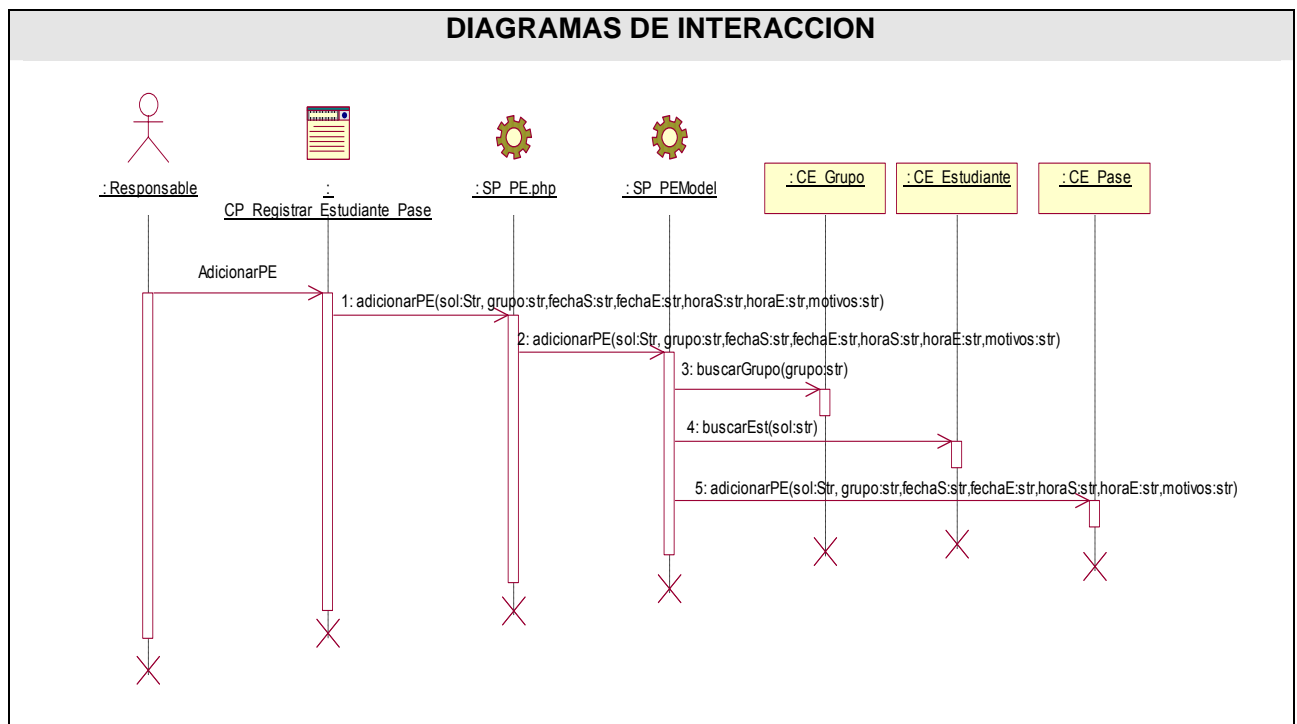
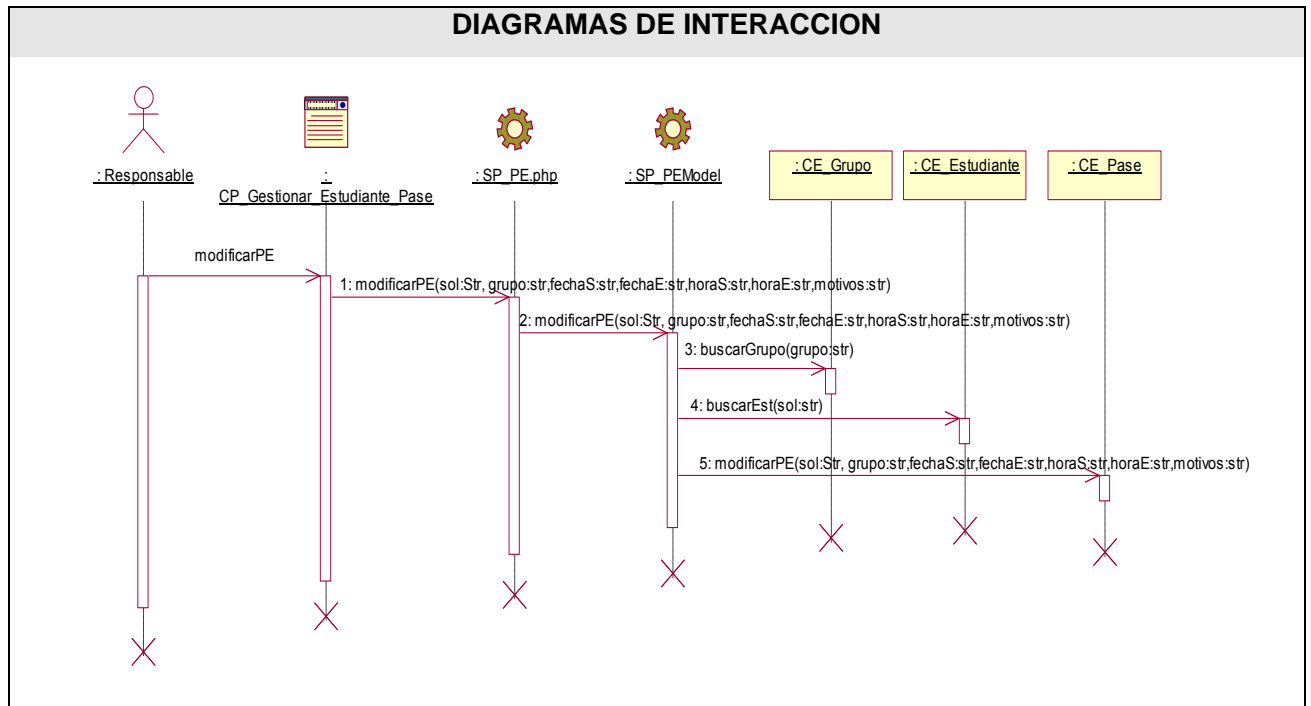


Tabla 31. Caso de Uso: “Gestionar Pases Eventuales”: Escenario: Adicionar Pases Eventuales.



**Tabla 32. Caso de Uso: “Gestionar Pases Eventuales”: Escenario: Modificar Pases Eventuales.**

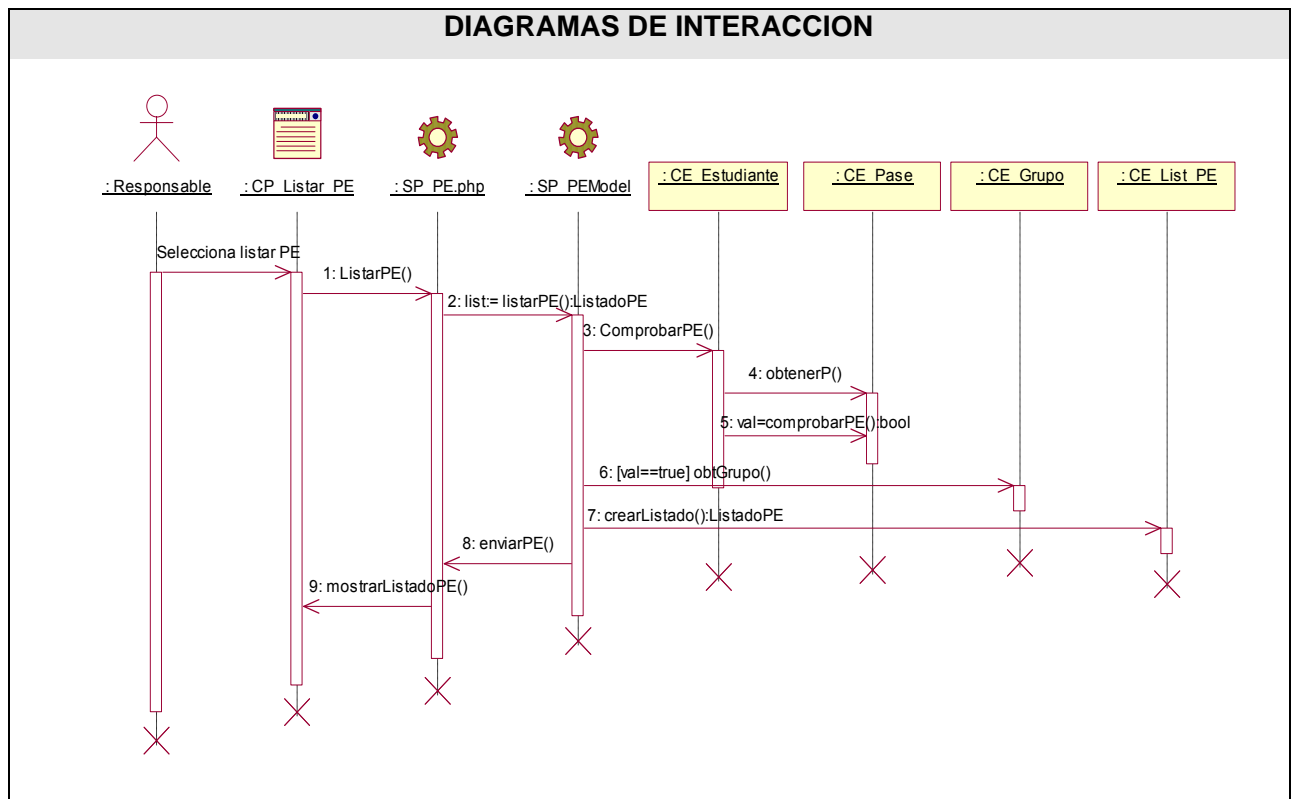


Tabla 33. Caso de Uso: “Gestionar Pases Eventuales”: Escenario: Listar Pases Eventuales.

**Gestionar licencias de matrículas.**

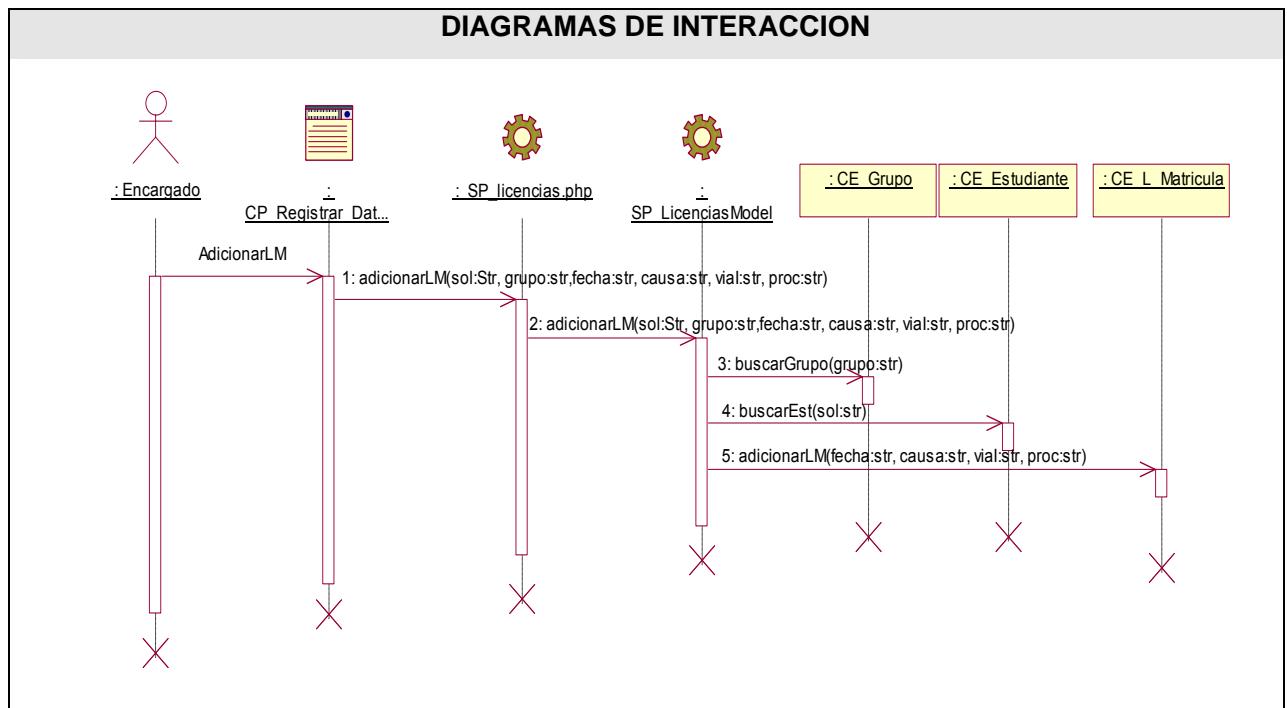
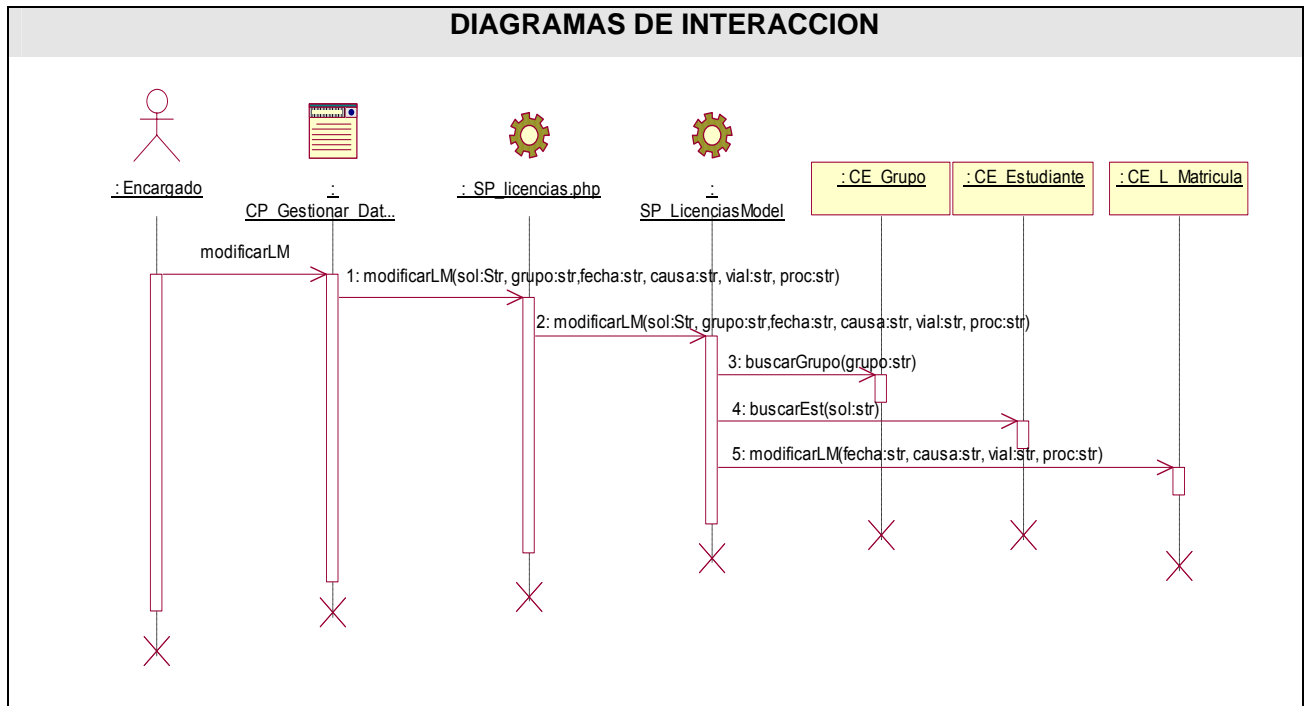


Tabla34. Caso de Uso: "Gestionar Licencias de Matrículas": Escenario: Adicionar Licencias Matrículas.





**Tabla35. Caso de Uso: "Gestionar Licencias de Matrículas": Escenario: Modificar Licencias de Matrículas.**

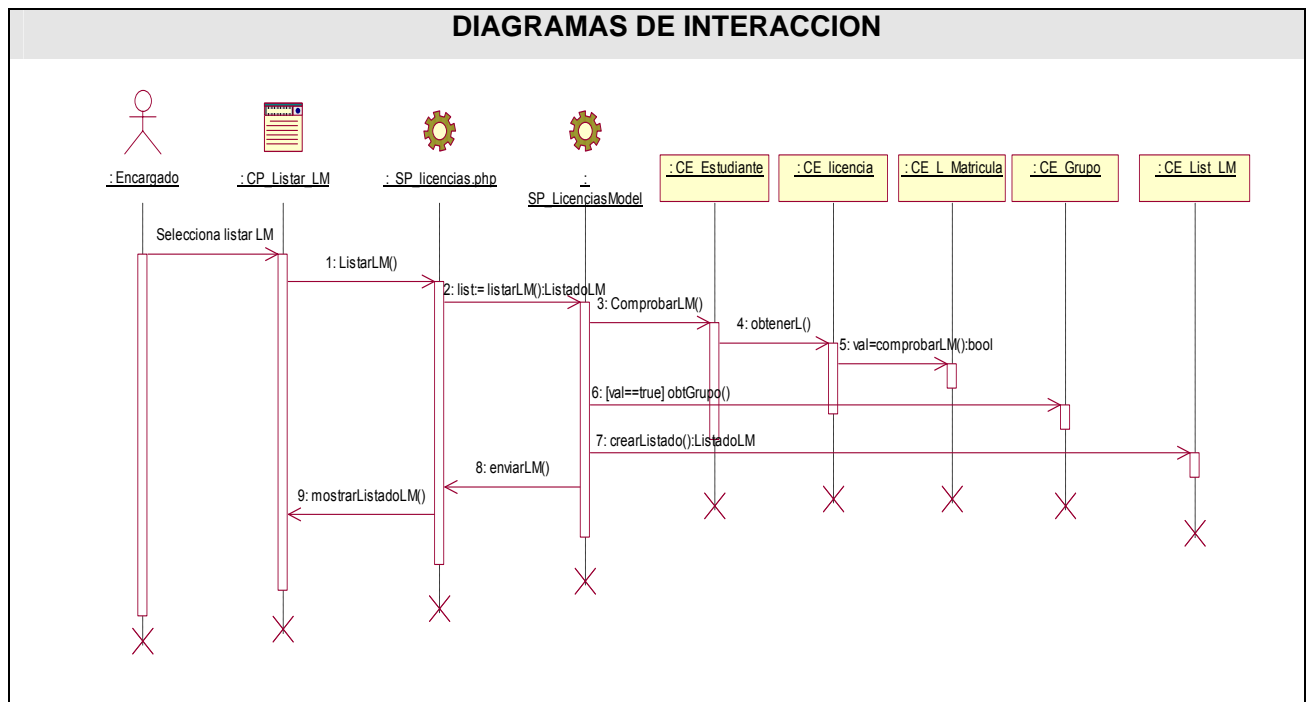
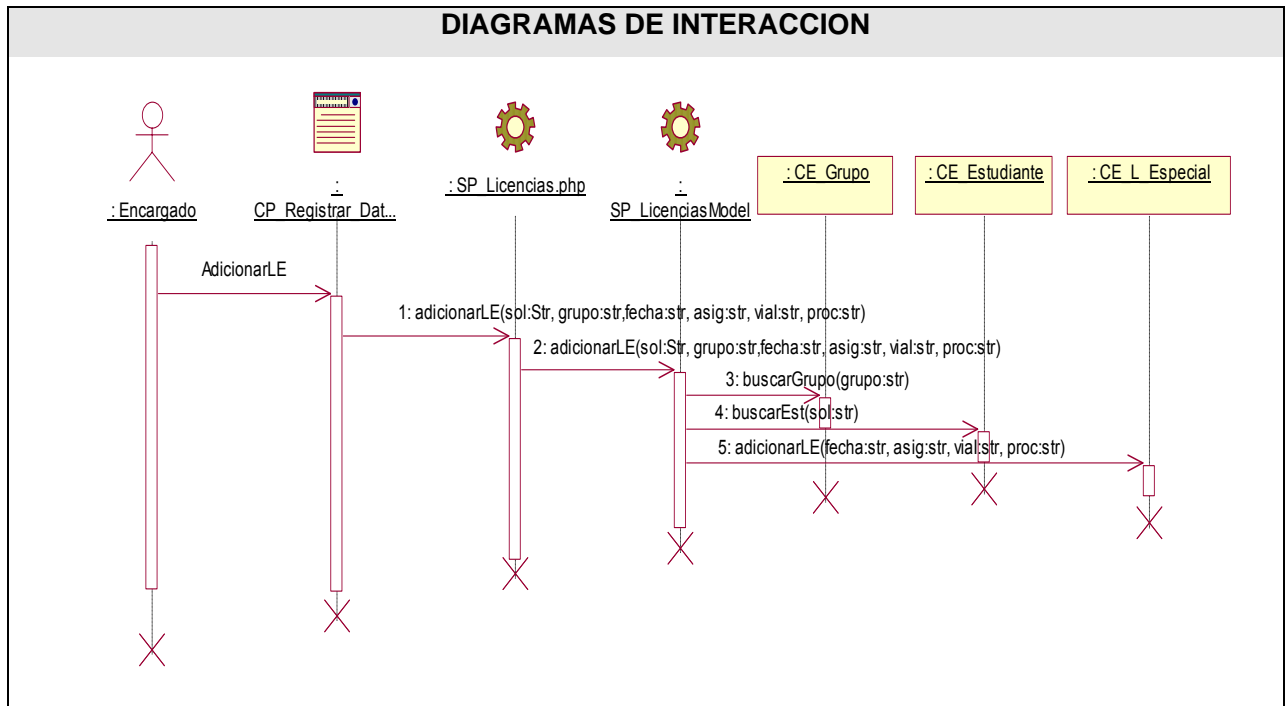
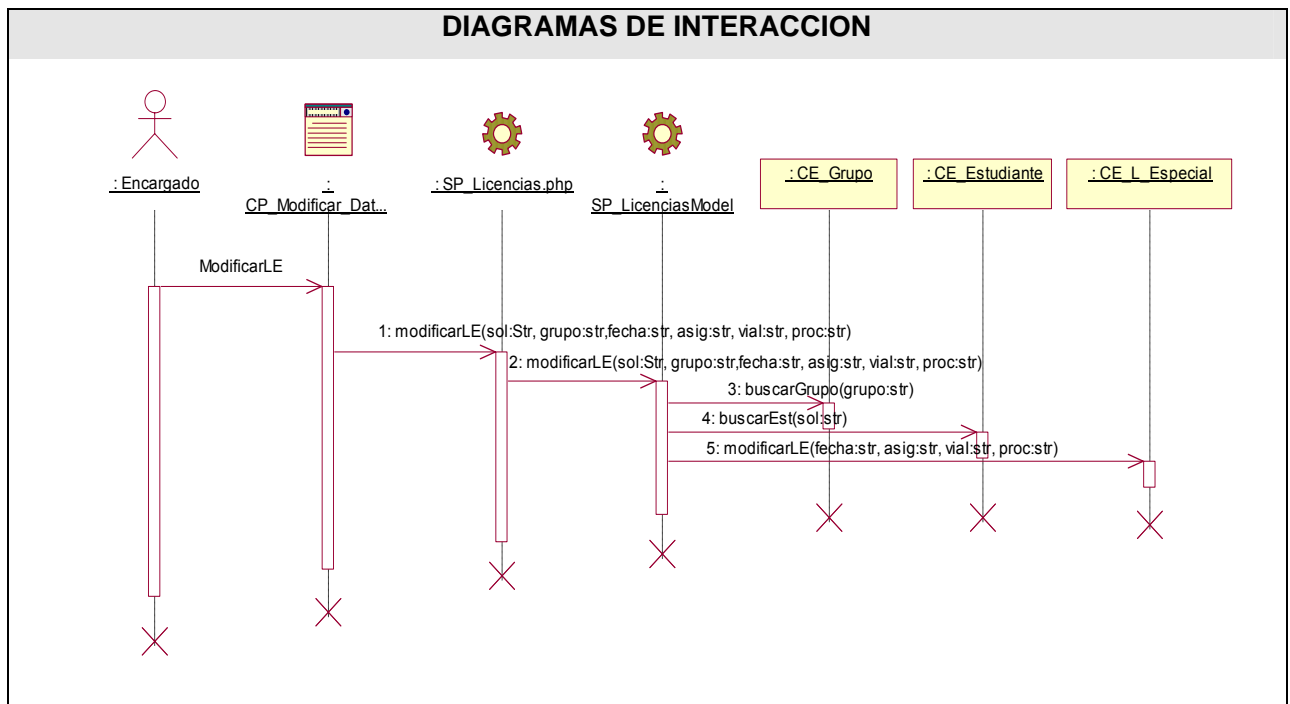


Tabla 36. Caso de Uso: “Gestionar Licencias de Matrículas”: Escenario: Listar Licencias de Matrículas.

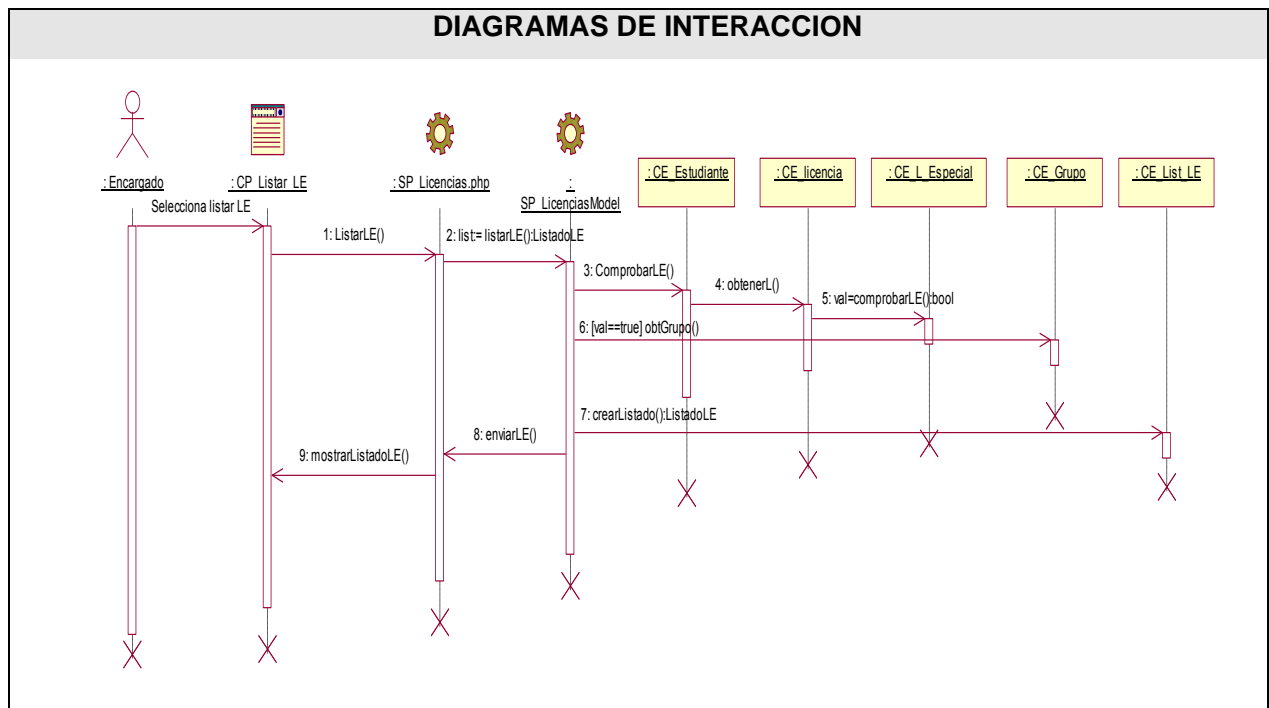
**Gestionar licencias especiales.**



**Tabla 37. Caso de Uso: “Gestionar Licencias Especiales”: Escenario: Adicionar Licencias Especiales.**



**Tabla 38. Caso de Uso: “Gestionar Licencias Especiales”: Escenario: Modificar Licencias Especiales.**



**Tabla 39. Caso de Uso: “Gestionar Licencias Especiales”: Escenario: Listar Licencias Especiales.**

**Autenticarse.**

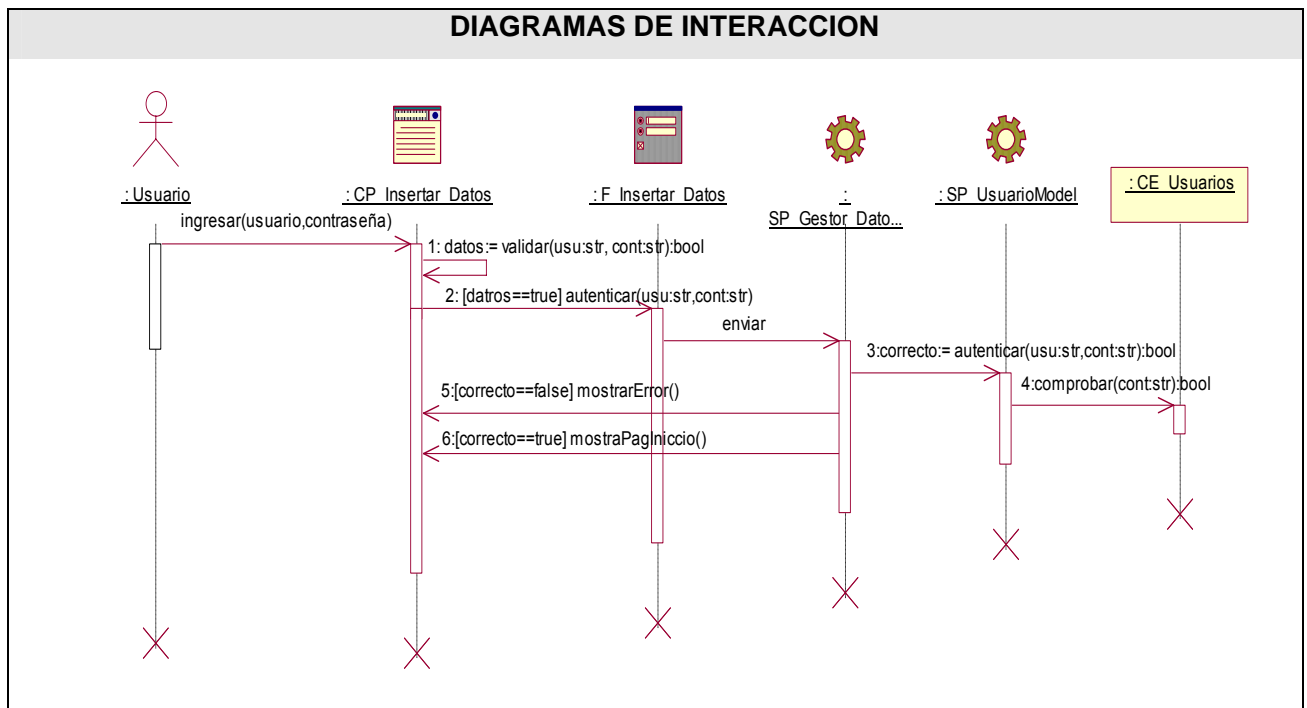


Tabla 40. Caso de Uso: "Autenticarse".

Administración.

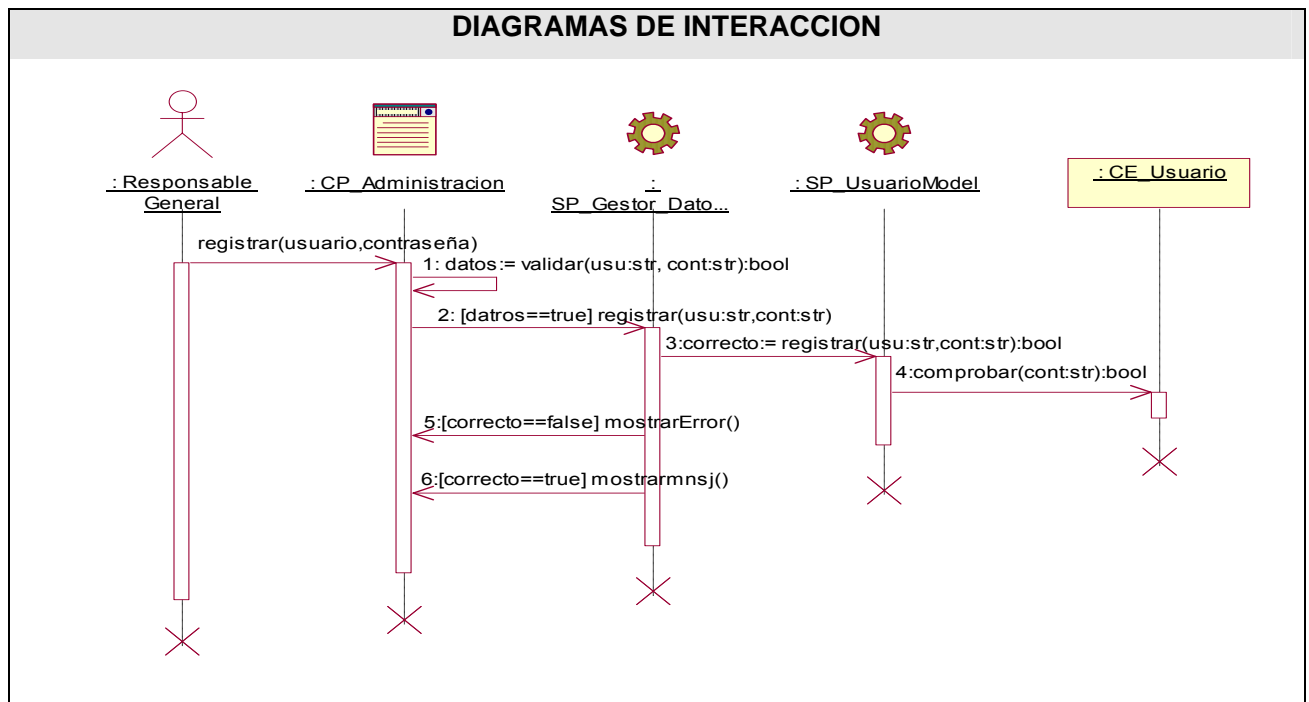


Tabla 41. Caso de Uso: "Administración".

## Glosario de Términos

**Akademios:** Sistema de gestión académica.

**AJAX:** Acrónimo de *Asynchronous JavaScript And XML* (JavaScript y XML asíncronos).

**BSD:** Distribución de Software Berkeley.

**Callback:** Gestor de eventos.

**CASE:** Computer Aided Software Engineering

**CES:** Centros de Educación Superior.

**CPU:** Unidad central de proceso.

**C/S:** Cliente/Servidor.

**CSS:** Hojas de estilos.

**DBMS:** DataBase Management System.

**DHTML:** HTML Dinámico.

**DOM:** Document Object Model.

**EBML:** Acrónimo de Extensible Binary Meta Language (Meta Lenguaje Binario Extendible)

**ECMAScript:** Especificación de lenguaje de programación publicada por ECMA International.

Define un lenguaje de tipos dinámicos ligeramente inspirado en Java y otros lenguajes del estilo de C.

**Firefox:** Navegador de Internet.

**Framework:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

**FTP:** Protocolo de transferencia de ficheros entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor

**Gestacad:** Sistema para la gestión académica.

**GNU/Linux:** Sistema operativo que utiliza el kernel Linux en conjunto con las aplicaciones de sistema creadas por el proyecto GNU y de varios otros proyectos/grupos de software. Comúnmente este sistema operativo es denominado como Linux.

**Handler:** Representación interna de Apache de una acción que se va a ejecutar cuando hay una llamada a un fichero.

**Hosting:** Servicio ofrecido por algunos proveedores, **que** brindan a sus clientes (individuos o empresas) un espacio en su servidor



**HTML:** Hypertext Markup Language

**HTTP:** Hypertext Transfer Protocol

**HTTDP:** Programa que corre de fondo en un servidor web y espera peticiones de entrada para responderles.

**IIS: Internet Information Services** (Serie de servicios para los ordenadores que funcionan con Windows).

**Iframe:** Elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML.

**JavaScript:** Lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web.

**JScript:** Lenguaje de secuencias de comandos moderno con una gran variedad de aplicaciones.

**JSON:** Acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos.

**LAMP:** Conjunto de subsistemas software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un esfuerzo reducido.

**MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras.

**NCSA:** Acrónimo del National Center for Supercomputing Applications (Centro Nacional de Aplicaciones de Supercomputación).

**MVC:** Modelo Vista Controlador.

**ODBC:** Estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation

**Opera:** Navegador de Internet creado por la empresa noruega **Opera** Software en 1994.

**Patchy server:** Un servidor "parcheado".

**P2P:** Se refiere a una red que no tiene clientes ni servidores fijos, sino una serie de nodos que se comportan simultáneamente como clientes y como servidores de los demás nodos de la red.

**PHP:** Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

**Plugins:** Tienen la función de reproducir determinados formatos de gráficos, reproducir datos multimedia, codificar/decodificar emails

**PostgreSQL:** Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

**Request:** Permite el acceso a toda la información que pasa desde el navegador del cliente al servidor.

**RUP:** Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

**Safari:** Navegador web desarrollado por Apple para su sistema operativo Mac OS X

**Scripts:** Ofrecen a los autores la posibilidad de extender los documentos HTML de maneras activas e interactivas.

**SGBD:** Sistema de Gestión de Bases de Datos. 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.

**SGISD:** Software de gestión de la información de la secretaria docente.

**SQL:** Structured Query Language. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas.

**UCI:** Universidad de las Ciencias Informáticas.

**Unix:** Sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell

**URL:** Uniform Resource Locator, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar

**WEB (WWW):** Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

**XHTML:** Acrónimo inglés de extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto).

**XML:** Extensible Markup Language. Es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium. Orientado principalmente al almacenamiento, procesamiento y transmisión de mensajes.

**XMLHttpRequest:** Interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB.