

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Facultad 2

TÍTULO: Sistema de gestión de información del Área de Extensión Universitaria.

TESISTAS:

Yisel Rosabal Sánchez

Nosbel Méndez Cordero

TUTORA:

Ing. Anabel Parra Vázquez

Ciudad de la Habana, Junio 2007

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad #2 de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Autores:

\_\_\_\_\_  
Yisel Rosabal Sanchez

\_\_\_\_\_  
Nosbel Mendez Cordero

Tutor:

\_\_\_\_\_  
Ing. Anabel Parra Vázquez

## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado: **Sistema de gestión de información del Área de Extensión Universitaria**, fue realizado en **la Facultad #2**. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

Totalmente

Parcialmente en un \_\_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

1. Automatiza los procesos que se generan en el vicedecanato, como son la guardia obrero-estudiantil y la cuartería.
2. Agiliza el proceso de búsqueda de los estudiantes en la residencia, contando con una búsqueda avanza a partir de varios criterios.
3. Permite tener actualizada la situación real de cada uno de los estudiantes a poder establecer parámetros como lo son ubicación por grupos, y si pertenece al movimiento de deportistas.

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a \$ 1272.

Y para que así conste, se firma la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Representante de la entidad: Abdel Eduardo Martínez Valverde.

Cargo: Vicedecano de Extensión Universitaria de la Facultad #2.

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño

**"No basta dar pasos que puedan conducir hasta la meta;  
sino que cada paso sea una meta, sin dejar de ser un paso".**

**Johann Peter Eckerman.**

## **AGRADECIMIENTOS**

Ante todo agradecer a la Revolución Cubana por darme la oportunidad de formarme como un hombre de bien, dispuesto a poner los conocimientos adquiridos a lo largo de la vida para el desarrollo de la revolución.

A nuestros familiares especialmente a padres, abuelos, tíos, por siempre apoyarnos y darnos fuerzas y confianza para seguir adelante y vencer todas las dificultades que se presenten.

A la tutora por estar disponible y brindarnos su ayuda siempre que la necesitemos.

A los compañeros de quinto año especialmente a los muchachos del laboratorio 301, entre ellos: Maikel, Yuri, Giorbis, Yasser, Elvis.

A nuestros amigos Anglada, Yennis, Leonardo y Yoanky por la ayuda brindada.

A todos aquellos que de una forma u otra han contribuido a la realización de esta tesis.

## **DEDICATORIA**

Especialmente a nuestros padres, abuelos, tíos y amistades por el apoyo, amor y cariño que nos brindaron en todo momento.

**RESUMEN**

Con el desarrollo de la informática y el aumento de su impacto social, son cada vez más las instituciones u organizaciones que optan por incorporar aplicaciones que gestionen su información, logrando así una mayor dinámica en sus procesos de negocio.

Este trabajo ha sido realizado en la Universidad de las Ciencias Informáticas, centro en el cual la Extensión Universitaria es el vínculo natural que, a través de las manifestaciones culturales y la acción social, potencia los valores éticos y espirituales como contribución al desarrollo cultural de los estudiantes y trabajadores.

Con este trabajo se persigue brindar una solución automatizada, flexible y única a todo el proceso de control y gestión de información del Área de Extensión Universitaria.

Es por ello, que para la realización del trabajo se decidió desarrollar una aplicación Web, basada en tecnología PHP y con gestor de base de datos PostgreSQL.

Los objetivos concretos trazados son: Realizar una aplicación Web de fácil acceso y con una interfaz cómoda que facilite la obtención de los datos de los estudiantes de la facultad # 2, automatizar la gestión de la guardia y la cuartelaría en la facultad # 2, diseñar e implementar una base de datos capaz de almacenar de manera organizada la información que se manipula durante estos procesos, desarrollar el análisis y diseño de los procesos seleccionados, implementar el sistema con las características definidas en los procesos de análisis y diseño.

El sistema propuesto contribuirá al control y almacenamiento de la información, a que exista una mayor rapidez en las búsquedas de la misma y a mejorar favorablemente las condiciones de trabajo del personal que lleva a cabo el control y la gestión de la información relacionada con esta área.

**ÍNDICE**

INTRODUCCIÓN ..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ..... 5

    1.1 Introducción..... 5

    1.2 Sistemas automatizados existentes vinculados al campo de acción ..... 5

    1.3 Tendencias y tecnologías actuales ..... 6

        1.3.1 Las aplicaciones Web ..... 6

        1.3.2 Modelo Cliente Servidor ..... 8

        1.3.3 PHP (Hypertext Preprocessor)..... 9

        1.3.4 Servidor Web Apache ..... 10

        1.3.5 AJAX (Asynchronous JavaScript And XML)..... 12

        1.3.5 Modelo Vista Controlador (MVC) ..... 14

        1.3.6 Sistemas de Gestión de Base de Datos..... 15

            1.3.6.1 PostgreSQL..... 16

        1.3.7 Proceso de Desarrollo de Software..... 18

        1.3.8 Lenguaje Unificado de Modelado (UML)..... 19

    1.4 Herramientas utilizadas..... 20

        1.4.1 Diseño de interfaz ..... 20

        1.4.2 Adobe Photoshop..... 21

        1.4.3 Rational Rose..... 21

    1.5 Conclusiones..... 22

CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA..... 23

    2.1 Introducción..... 23

    2.2 Descripción de los procesos que serán objeto de automatización. .... 24

    2.3 Actores y trabajadores del negocio ..... 24

    2.4 Diagrama de casos de uso del negocio ..... 25

    2.5 Requerimientos funcionales ..... 26

    2.6 Requerimientos no funcionales ..... 27

    2.7 Actores del sistema a automatizar ..... 29

    2.8 Paquetes y relaciones entre ellos ..... 30

    2.9 Diagrama de Caso de Uso del Sistema a automatizar ..... 30

    2.10 Conclusiones..... 31

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA..... 33



3.1 Introducción.....	33
3.2 Diagramas de clases del análisis.....	33
3.3 Diagramas de clases del diseño.....	36
3.4 Diseño de la base de datos.....	40
3.4.1 Diagrama de clases persistentes.....	40
3.4.2 Modelo de datos.....	41
3.5 Conclusiones.....	41
CAPÍTULO 4 IMPLEMENTACIÓN.....	42
4.1 Diagrama de despliegue.....	42
4.2 Diagramas de componentes por paquetes.....	43
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.....	45
5.1 Introducción.....	45
5.2 Estimación del esfuerzo.....	45
5.3 Beneficios tangibles e intangibles.....	49
5.4 Análisis de costos y beneficios.....	49
5.5 Conclusiones.....	50
CONCLUSIONES.....	51
RECOMENDACIONES.....	52
REFERENCIAS BIBLIOGRÁFICAS.....	53
Bibliografía.....	55
GLOSARIO DE TÉRMINOS Y SIGLAS.....	56
ANEXOS.....	59
Anexo # 1.....	59
Anexo # 2.....	63
Anexo # 3.....	73

**ÍNDICE DE FIGURAS**

Figura 1.1 Esquema general de la tecnología Web..... 7

Figura 1.3 Flujos de Trabajo de RUP..... 19

Figura 2.1 Diagrama de Casos de Uso del Negocio..... 26

**ÍNDICE DE TABLAS**

Tabla 2.1 Descripción de los actores del negocio ..... 25

Tabla 2.2 Descripción de los trabajadores del negocio ..... 25

## INTRODUCCIÓN

En América Latina se habló por vez primera de "Extensión Universitaria", como función social inherente e inseparable de la universidad, en la Reforma de Córdoba, Argentina (1918), en respuesta a la crisis motivada por la confrontación entre una sociedad que se movía hacia el cambio y una universidad que se aferraba a sus esquemas obsoletos.

El Manifiesto de Córdoba entre sus demandas exigía: "La Extensión Universitaria, el fortalecimiento de la función social de la universidad, proyección al pueblo de la cultura universitaria y preocupación por los problemas nacionales", poniendo fin a la política de aislamiento de la universidad respecto al resto de la sociedad. Este movimiento propició el desarrollo cultural interno de las universidades y el acercamiento de la intelectualidad de avanzada y los estudiantes a los sectores obrero, campesino, y al resto de la sociedad. Así mismo se le reconoció a la extensión un enfoque más abarcador en la función social de la universidad. La extensión universitaria se integraría a la universidad para formar parte de ella.

En la Universidad de las Ciencias Informáticas específicamente en la Facultad # 2 no existe un software que permita aplicar las tecnologías de la informática y las comunicaciones para agilizar la generación y el flujo de información en el área de Extensión Universitaria de dicha facultad.

Es por ello que entre las actividades que tiene que llevar a cabo el ViceDecano de Extensión están: el control de la guardia estudiantil y la cuartería, estos procesos se realizan manualmente; en el caso de la cuartería, las instructoras son las encargadas de distribuirla por edificios, apartamentos, y pasos de escalera por lo que el VD Extensión cuando necesita saber que estudiante estuvo o está de cuarterero, o alguna otra información que sea de su interés tiene que ir a ver a las instructoras para obtener estos datos. El control de la guardia también se torna engorroso porque además de que se realiza manualmente como ya se había dicho, el encargado de llevar a cabo este proceso tiene que tener en cuenta muchos aspectos para distribuir la misma, y en ocasiones comete errores, esto traería consigo muchos problemas, también pueden ocurrir pérdidas de información, etc.

Actualmente cuando el VD Extensión desea buscar algún dato referente a algún estudiante de la facultad por ejemplo: provincia, municipio, edificio, apartamento, tiene que realizar una búsqueda en un Excel con un filtro y este proceso demora mucho, además que no le brinda datos que son de su interés como: si el estudiante es deportista y artista, y en caso de serlo en que deporte y manifestación cultural se desempeña, todo esto provoca que las condiciones laborales no sean las mas propicias. Con el objetivo de mejorar este proceso en la facultad, se decidió realizar una aplicación Web que permita controlar y gestionar la información relacionada con el área de extensión universitaria.

El **problema** a resolver de esta investigación surge con la necesidad de solucionar lo antes expuesto, y es la no existencia de un software que permita aplicar las tecnologías de la informática y las comunicaciones para agilizar la generación y el flujo de información en el área de extensión universitaria en la Facultad # 2 de la Universidad de las Ciencias Informáticas.

El **objeto de estudio** es todo el trabajo relacionado con el área de Extensión Universitaria.

De donde se deriva que el **campo de acción** comprende el control y gestión de información, de procesos tales como: la guardia estudiantil y la cuarterería en el área de Extensión Universitaria, en la Facultad # 2.

El **problema científico** del trabajo es: ¿Cómo erradicar las dificultades en las actividades que tienen relación con el VD Extensión, entre ellas el control y gestión de la guardia estudiantil y la cuarterería, y de esta forma mejorar las condiciones laborales en la facultad # 2?

Se ha propuesto como **objetivo general**: realizar un sistema informático para controlar y gestionar la información del área de Extensión Universitaria.

Con la realización de este trabajo se esperan los siguientes aportes prácticos:

- ✓ Lograr un sistema que controle y gestione la información del VD Extensión, es decir, que facilite obtener los datos fundamentales de los estudiantes de la facultad # 2, así como el control de la guardia estudiantil y la cuartelería.
- ✓ Rapidez en las búsquedas de información.
- ✓ Información actualizada rápidamente.

Como objetivos Específicos se plantean los siguientes:

- ✓ Realizar una aplicación Web de fácil acceso y con una interfaz cómoda que facilite la obtención de los datos de los estudiantes de la facultad # 2.
- ✓ Automatizar la gestión de la guardia y la cuartelería en la facultad # 2.
- ✓ Diseñar e implementar una base de datos capaz de almacenar de manera organizada la información que se manipula durante estos procesos.
- ✓ Desarrollar el análisis y diseño de los procesos seleccionados.
- ✓ Implementar el sistema con las características definidas en los procesos de análisis y diseño.

Para cumplir los objetivos se desarrollaron las siguientes tareas:

- ✓ Selección y revisión bibliográfica de la información precedente sobre el tema.
- ✓ Procesamiento y evaluación de la información obtenida sobre el tema.
- ✓ Selección de la metodología con la cual se trabajará.
- ✓ Interacción con el cliente donde se definen las bases del Modelo de Negocio.
- ✓ Especificación de los Requerimientos definidos a partir del Modelo de Negocio.
- ✓ Especificación de la Interfaz de Usuario del sistema.
- ✓ Diseño de una base de datos que soporte la mayoría de las funcionalidades del sistema entre ellas la cuartelería y la guardia estudiantil.
- ✓ Implementación del software haciendo uso de herramientas de código abierto como PostgreSQL como gestor de base de datos y PHP como lenguaje de programación.

El presente trabajo, estructurado en 5 capítulos, resume la siguiente información:

Capítulo 1: Fundamentación Teórica. En este capítulo se incluye un estado del arte del tema tratado, tanto a nivel internacional, nacional y de la Universidad, además de las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

Capítulo 2: Características del sistema. Se muestra el problema y la situación problemática donde se especifican los objetivos estratégicos de la organización y procesos de negocio que los soportan, el flujo actual de los procesos involucrados en el campo de acción y un análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. También se escriben los sistemas automatizados que existen y que están vinculados con el campo de acción. Se indica quiénes son los actores y trabajadores del negocio y por qué lo son, se representa el diagrama de caso de uso del negocio y sus descripciones textuales, los diagramas de actividades y modelos de objetos. Además se muestra un listado de los requerimientos funcionales enumerados y se definen los no funcionales, el modelo de casos de uso del sistema y la definición de los actores del sistema a automatizar y la descripción de los casos de uso, por ejemplo: el caso de uso Gestionar Cuartelería y Gestionar Guardia Estudiantil.

Capítulo 3: Análisis y diseño del sistema. En este capítulo se muestran los modelos de clases de análisis, los diagramas de interacción en el diseño y los diagrama de clases del diseño. También está el modelo lógico de datos (diagrama de clases persistentes) y el modelo físico de datos (modelo de datos).

Capítulo 4: Implementación. En este capítulo se muestra el diagrama de despliegue y el de componentes.

Capítulo 5: Estudio de factibilidad. Se muestran los cálculos de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo. También están los beneficios tangibles e intangibles y se cuantifican los tangibles y se analiza el costo.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

Este capítulo brinda un enfoque general de sistemas automatizados existentes vinculados al campo de acción y el análisis comparativo de las soluciones existentes con la propuesta dada en este trabajo. Se describen además las tecnologías actuales de desarrollo utilizadas para el análisis, diseño e implementación del sistema sobre las cuales se apoya la propuesta.

### **1.2 Sistemas automatizados existentes vinculados al campo de acción**

Actualmente en la universidad existen algunos sistemas vinculados al campo de acción pero cada uno de ellos tiene características particulares y no responden totalmente a lo que se necesita.

- 1) El directorio es una aplicación en la cual se obtienen datos de los estudiantes tales como: nombre(s) y apellidos, grupo, municipio y provincia, pero no muestra otro tipo de información de los estudiantes que es de interés para todos como edificio, apto, si el estudiante es deportista o no, si es artista aficionado o no, el sexo. Además el directorio está hecho en tecnología .NET y usa como gestor de Base Datos, SQLServer. .NET es un software propietario, el servidor solo funciona sobre Windows NT, Windows 2000 o superior, el código es cerrado, es decir, solo visto por sus desarrolladores.
- 2) El nuevo directorio nos ofrece una mayor posibilidad ya que puede realizar búsquedas por varios criterios. Al igual que el anterior está hecho utilizando tecnología .NET, aunque la mayor parte del código es en Java Script, del lado del cliente y se usa como Base de datos SQLServer. Este nuevo buscador se piensa migrar a PHP y como gestor de Base de Datos se usará MySQL. Las ventajas de este nuevo directorio con respecto al otro es que es más amigable, más rápido, y permite búsquedas más complejas.
- 3) Otro de los sistemas vinculados al campo de acción es Akademos, el estudiante puede consultar su expediente docente, el mismo contiene toda la información existente en la Secretaría Docente de su facultad, cada estudiante podrá consultar sus evaluaciones así como las de cualquier otro estudiante, los profesores, personal de secretaría y cuadros de dirección podrán consultar el registro de evaluaciones de cualquier estudiante, pero un estudiante no puede



acceder a los datos de los profesores. De forma similar deberá responder el sistema de gestión y control de la cuartería y la guardia estudiantil acorde con nuestras especificaciones, o sea los estudiantes solo podrán acceder a un grupo de informaciones referidas a ellos, con permiso de solo de lectura, así como no tendrán permiso a las acciones del administrador, estos últimos tendrán todo los privilegios, es decir el control total del sistema. Además este sistema a diferencia de los antes mencionados se realizará en PHP y este lenguaje de programación es más ventajosos que .NET.

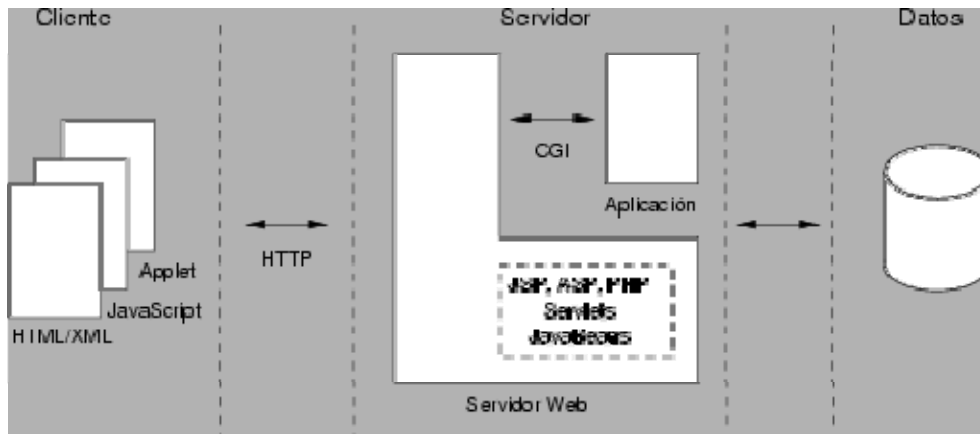
### **1.3 Tendencias y tecnologías actuales**

Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio de las tendencias y tecnologías actuales posibles a emplear, descritas a continuación.

#### **1.3.1 Las aplicaciones Web**

Las aplicaciones Web son una especialización y concreción de las aplicaciones cliente servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son estándar, y no han de ser creados por el desarrollador. [1]

La parte del cliente de las aplicaciones Web está formada por el código HTML (HyperText Markup Language) que forma la página Web, con opción a código ejecutable mediante los lenguajes script de los navegadores (JavaScript, VBScript, PerlScript) o mediante pequeños programas (applets) en Java. La parte del servidor está formada por un programa o script que es ejecutado por el servidor Web, y cuya salida se envía al navegador del cliente. [1]



**Figura 1.1 Esquema general de la tecnología Web.**

La creciente popularidad de las aplicaciones Web se debe a las múltiples ventajas, que estas tienen entre las cuales podemos citar: [2]

- **Multiplataforma:** Con un solo programa, un único ejecutable, nuestras aplicaciones pueden ser utilizadas a través de múltiples plataformas, tanto de hardware como de software.
- **Actualización instantánea:** Debido a que todos los usuarios de la aplicación hacen uso de un sólo programa que radica en el servidor, los usuarios siempre utilizarán la versión más actualizada del sistema.
- **Suave curva de aprendizaje:** Los usuarios, como utilizan la aplicación a través de un navegador, hacen uso del sistema tal como si estuvieran navegando por Internet, por lo cual su acceso es más intuitivo.
- **Fácil de integrar con otros sistemas:** Debido a que se basa en protocolos estándares, la información manejada por el sistema puede ser accedida con mayor facilidad por otros sistemas.
- **Acceso móvil:** El usuario puede acceder a la aplicación con la única restricción de que cuenta con un acceso a la red privada de la organización o a Internet, dependiendo de las políticas de dicha organización; puede hacerlo desde una computadora de escritorio, una laptop o desde una agenda electrónica; desde su oficina, hogar u otra parte del mundo.

El desarrollo de aplicaciones Web está siendo utilizado en muchas organizaciones, ésta situación va ir creciendo indefinidamente. Es por ello que día a día se requieren más programadores capacitados para desarrollos basados en el World Wide Web (WWW).

No obstante a la serie de ventajas que presenta tiene además algunas desventajas, las cuales son:

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera algunas veces excesivo hasta que se obtiene la reacción del sistema.
- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

### **1.3.2 Modelo Cliente Servidor**

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes: [3]

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

Ventajas de la arquitectura cliente-servidor: [2]

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

### 1.3.3 PHP (Hypertext Preprocessor)

El PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre las que se pueden mencionar InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML (extensible Markup Language) y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+. Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas. Es muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado. [2]

Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra. Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages). [2]

PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos. Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes. Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y

se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

El funcionamiento del PHP se puede describir a través de los pasos siguientes: [4]

- Escribir en las páginas HTML pero con el código PHP dentro.
- Guardar la página en el servidor Web.
- Un navegador solicita una página al servidor.
- El servidor interpreta el código PHP.
- El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario, el código PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML. Por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar. [10]

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. Además se encuentra libre en el mercado y se puede acceder a él por medio de Internet. [11]

Además de ser la petición de nuestro cliente y después de visto todo esto el PHP es el adecuado para implementar la propuesta de sistema de este trabajo.

#### **1.3.4 Servidor Web Apache**

Un servidor de páginas Web es un programa que permite acceder a páginas Web alojadas en un ordenador. Hoy en día Apache es el servidor Web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que nos serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable.

Los servidores Web suministran páginas Web a los navegadores que lo solicitan. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones Web. Usando HTTP, un servidor Web envía páginas Web en HTML y Common Gateway Interface (CGI), así como otros tipos de scripts a los navegadores o browsers cuando éstos los requieren. Cuando un usuario hace clic sobre un enlace a una página Web, se envía una solicitud al servidor Web para localizar los datos nombrados por ese enlace. El servidor Web recibe esta solicitud y suministra los datos que le han sido solicitados o bien devuelve un mensaje de error.

El servidor Apache es un software que esta estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías: [3]

- Módulos Base: Módulo con las funciones básicas del Apache.
- Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

### **1.3.5 AJAX (Asynchronous JavaScript And XML)**

Es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la velocidad de interacción en la misma. [8]

AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente y que se muestra a continuación:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto IFrame en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Existen diferencias significativas entre las aplicaciones web tradicionales y las aplicaciones desarrolladas en AJAX, sí como sus patrones de interacción sincrónica para una aplicación Web tradicional y asincrónica para una aplicación AJAX.

#### **Ventajas del AJAX:**

- Interactividad:

Las aplicaciones AJAX se ejecutan en la máquina del usuario, manipulando la página actual dentro de sus navegadores usando métodos de Document Object Model. Puede

ser usado para multitud de tareas como actualizar o eliminar registros, expandir formularios Web, devolver peticiones simples de búsqueda, o editar árboles de categorías; todo sin tener la necesidad de recargar toda la página de HTML cada vez que se realiza un cambio. Generalmente solo requiere enviar pequeñas peticiones al servidor, y se devuelven respuestas relativamente cortas. Esto permite el desarrollo de aplicaciones interactivas con más interfaces de usuario más responsivas gracias al uso de las técnicas DHTML.

- Portabilidad:

Las aplicaciones construidas con AJAX utilizan características bien documentadas presentes en todos los navegadores importantes en la mayoría de las plataformas existentes. Aunque esta situación podría cambiar en el futuro, en este momento, los usos de AJAX son efectivos entre plataformas. Mientras que la plataforma de AJAX está más restringida que la plataforma de Java, las aplicaciones actuales de AJAX llenan con eficacia la parte de los Java Applets: ampliar el navegador con mini-aplicaciones ligeras. [9]

### **Desventajas del AJAX:**

- Críticas de usabilidad:

Una de las mayores críticas contra el uso de AJAX en aplicaciones Web es que puede fácilmente acabar con el comportamiento normal del botón atrás del navegador. Las diversas expectativas entre volver a una página que se ha modificado dinámicamente y la vuelta a una página estática pueden ser sutiles. Otro problema relacionado es que las actualizaciones dinámicas hacen difícil al usuario agregar a los marcadores/favoritos un momento particular de la aplicación. [9]

- JavaScript:

Aunque AJAX no necesita ningún tipo de plug-in para el navegador, requiere que los usuarios tengan el JavaScript activado. Esto se aplica a todos los navegadores que soportan esta tecnología excepto para Microsoft Internet Explorer 6 y anteriores los cuales necesitan también tener el ActiveX activado, ya que el objeto XMLHttpRequest está implementado junto con el ActiveX en este navegador. [9]

Como ocurre con las aplicaciones DHTML, las de AJAX deben de ser probadas rigurosamente para adaptarse a los diferentes navegadores y plataformas.



### **1.3.5 Modelo Vista Controlador (MVC)**

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.

Son muchas las empresas que deciden pasar sus aplicaciones a la arquitectura modelo vista controlador para documentar más fácilmente el código, ahorrar espacio y en caso de no disponer de diseñadores Web, poder contratar los servicios de un diseñador que no sepa mucho de programación que les haga las vistas.

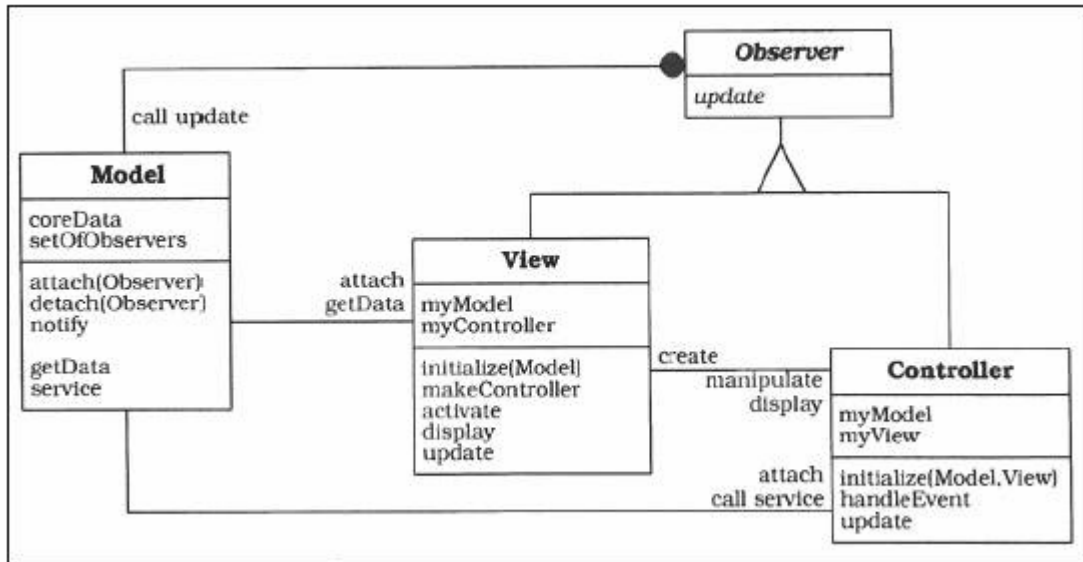
El Modelo es todo acceso a datos, y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo modelo (por ejemplo, una aplicación "tienda" y una "contabilidad" que accedan a las bases de datos de inventario, ventas, etc.).

La Vista, en una aplicación Web, es el HTML y lo necesario para convertir datos en HTML. O sea muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.

El Controlador es lo que une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las entradas del usuario. Recibe los eventos de entrada (un clic, un cambio en un campo de

texto, etc.). Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que "hace algo" está perfectamente separado del código dedicado a crear HTML, lo que ayuda a evitar el spaghetti. [15]



**Figura 1.2 Estructura del Modelo Vista Controlador**

### 1.3.6 Sistemas de Gestión de Base de Datos

Un Sistema de Gestión de Bases de Datos (SGBD) puede definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

Un SGBD tiene los siguientes objetivos específicos: [12]

- Independencia de los datos y los programas de aplicación.
- Minimización de la redundancia.
- Integración y sincronización de las bases de datos.
- Integridad de los datos.
- Seguridad y protección de los datos.

- Facilidad de manipulación de la información.
- Control centralizado.

La información es representada a través de tuplas, las cuales describen el fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Todos los sistemas mencionados anteriormente facilitan el trabajo con la base de datos y tienen características que los diferencian, por ejemplo:

- Oracle: requiere de una licencia para poderlo utilizar, es decir, es necesario pagar para poder utilizarlo.
- Microsoft SQL Server: no es multiplataforma, solo puede ser utilizado con el sistema operativo Windows que está patrocinado por la compañía Microsoft.
- MySQL: PostgreSQL soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL.

#### **1.3.6.1 PostgreSQL**

PostgreSQL es un sistema de gestión de bases de datos Objeto-Relacionales (ORDBMS) libre, liberado bajo la licencia BSD (Berkeley Software Distribution). Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle y SQLServer. El mismo ha sido desarrollado de varias formas desde 1977. [12]

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL

declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Es altamente extensible, soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL.

Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL (Tool Command Language) como lenguaje procedural embebido. Además usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

Las principales mejoras en PostgreSQL incluyen: [13]

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg\_dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido características adicionales que cumplen el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadenas literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

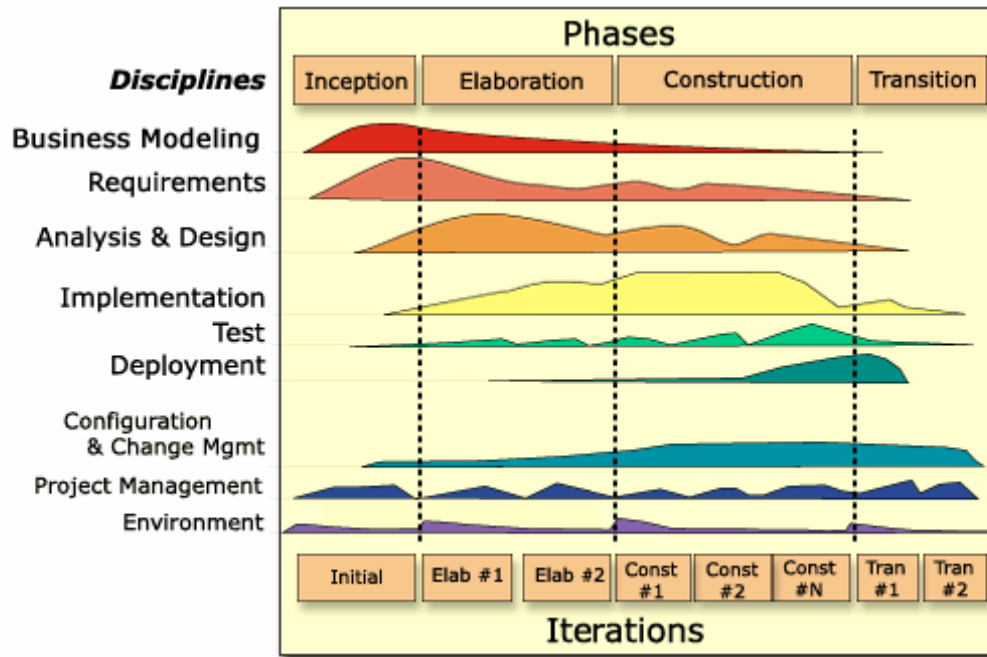
### **1.3.7 Proceso de Desarrollo de Software**

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo.

Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP). [6]

El Proceso Unificado de Rational, (Rational Unified Process, de ahí las siglas RUP), fue publicado en 1998 como resultado de varios años de experiencia. RUP es un proceso de desarrollo de software, o sea, conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. [7]

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, que parten de los casos de uso; centrado en la arquitectura y es iterativo e incremental. [14]



**Figura 1.3 Flujos de Trabajo de RUP**

Está acompañado de una herramienta muy buena que soporta cada uno de los procesos que necesitamos: Rational Rose Enterprise Edition 2003. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

### 1.3.8 Lenguaje Unificado de Modelado (UML)

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas. [10]

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas

generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos y es ésta precisamente la que se utiliza en la modelación de este proyecto. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto.

## **1.4 Herramientas utilizadas**

### **1.4.1 Diseño de interfaz**

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web. [5]

Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir una línea de código, así como también la codificación manual.

Dreamweaver ayuda además a construir aplicaciones Web dinámicas apoyadas en bases de datos. Es completamente personalizable. Se pueden crear objetos y

comandos propios, modificar los accesos directos de teclado, e incluso escribir código JavaScript para extender las capacidades del Dreamweaver con nuevos comportamientos. Soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP, Microsoft ASP.NET, Sun JavaServer Pages (JSP) y PHP.

#### **1.4.2 Adobe Photoshop**

Adobe Photoshop CS para el tratamiento de los gráficos. Es una herramienta muy poderosa para crear cualquier tipo de gráficos, su integración con Adobe ImageReady hacen que crear complicados gráficos para la Web sea una tarea muy fácil.

#### **1.4.3 Rational Rose**

Existen herramientas CASE de trabajo visuales como el Analise, el Designe, el Racional Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este proyecto.

Rational Rose cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

Es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

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

Se decidió que se utilizaría el Rational Rose Enterprise Edition 2003, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling



Language) para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades.

### **1.5 Conclusiones**

En este capítulo se exponen las condiciones y problemas que rodean el objeto de estudio; a través de los conceptos y definiciones planteadas. Se evidencia la necesidad de implementar un software que permita llevar a cabo el control y gestión de la información del Área de Extensión Universitaria. Para desarrollar el sistema se hace uso del lenguaje de programación PHP5 y con soporte de base de datos en PostgreSQL. El proceso de desarrollo es RUP y para el modelado visual se usará UML, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos y cambios.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

### 2.1 Introducción

El modelo del negocio posibilita obtener una visión más clara del proceso en cuestión, por ello en este capítulo se describen los actores y trabajadores del negocio y el modelo de objetos. Además se identifican los requisitos funcionales y no funcionales del sistema que darán solución al problema planteado; quiénes interactuarán con él (actores del sistema) y las distintas funcionalidades que ofrecerá a cada uno de los actores.

En la actualidad en la Facultad # 2 no existe un software que permita aplicar las tecnologías de la informática y las comunicaciones para agilizar la generación y el flujo de información en el área de Extensión Universitaria de esta facultad. Es por ello que entre las actividades que tiene que llevar a cabo el ViceDecano de Extensión están: el control de la guardia estudiantil y la cuartelería, los cuales se realizan de manualmente, en el caso de la cuartelería, las instructoras son las encargadas de distribuirla por edificios, apartamentos, y pasos de escalera por lo que el VD Extensión cuando necesita saber que estudiante estuvo o está de cuartelero, o alguna otra información que le interese tiene que ir a ver a las instructoras para obtener estos datos. El control de la guardia también se torna engorroso porque además de que se realiza manualmente como ya se había dicho, el encargado de llevar a cabo este proceso tiene que tener en cuenta muchos aspectos para distribuir la misma, y en ocasiones comete errores, esto traería consigo muchos problemas, también pueden ocurrir pérdidas de información, etc.

Actualmente cuando el VD Extensión desea buscar algún dato referente a algún estudiante de la facultad por ejemplo: provincia, municipio, edificio, apartamento, tiene que realizar una búsqueda en un Excel con un filtro y este proceso demora mucho, además que no le brinda datos que son de su interés como: si el estudiante es deportista y artista, y en caso de serlo en que deporte y manifestación cultural se desempeña, todo esto provoca que las condiciones laborales no sean las mas propicias. Con el objetivo de mejorar este proceso en la facultad, se decidió realizar una aplicación Web que permita controlar y gestionar la información relacionada con el área de extensión universitaria.

El problema a resolver de esta investigación surge con la necesidad de solucionar lo antes expuesto, y es la no existencia de un software que permita aplicar las tecnologías de la informática y las comunicaciones para agilizar la generación y el flujo de información en el área de extensión universitaria en la Facultad # 2 de la Universidad de las Ciencias Informáticas.

## **2.2 Descripción de los procesos que serán objeto de automatización.**

Los procesos que serán objeto de automatización en la aplicación son:

🚦 Gestionar la Guardia Estudiantil.

En este proceso a cada grupo se le asigna una fecha para realizar la guardia estudiantil, en caso de que la cantidad de estudiantes del grupo sea mayor que 20 este se divide en dos subgrupos, cada estudiante tiene asignado un subgrupo, un turno y posta. Una vez que todos los grupos de la facultad hayan realizado la guardia y que comience nuevamente por el primer grupo que la realizó, a los estudiantes no le corresponden los mismos turnos.

🚦 Gestionar la Cuartelería:

En este proceso al estudiante le corresponde una fecha para realizar la cuartelería, en correspondencia con su edificio, paso de escalera y apartamento, además se tiene en cuenta que el domingo no se realiza dicha actividad.

## **2.3 Actores y trabajadores del negocio**

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados.

De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor.

A continuación se muestra en la tabla 2.1, los actores del negocio y su correspondiente justificación:

Actores del Negocio	Justificación
Vice Decano de Extensión	El Vice Decano de Extensión es el que inicia las acciones que dan lugar a planificar la guardia estudiantil y la cuarterería, y al mismo tiempo es el principal beneficiado con el resultado de dichos procesos.

**Tabla 2.1 Descripción de los actores del negocio**

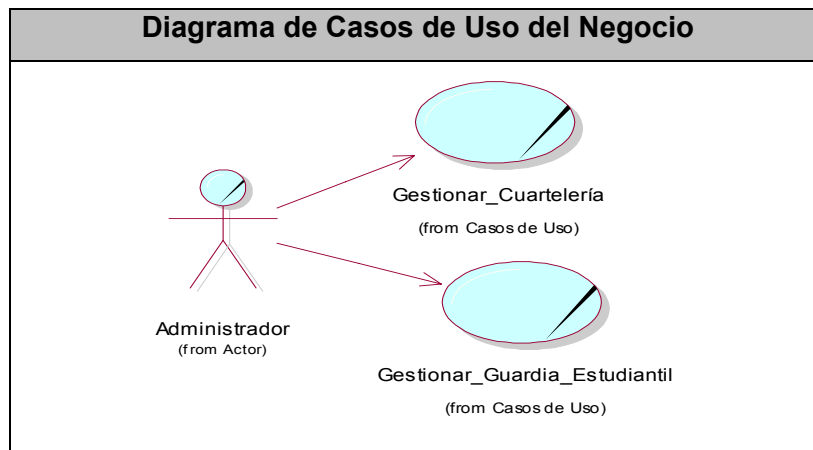
Trabajadores del negocio	Justificación
Estudiante	Es la persona que le corresponde realizar la guardia y la cuarterería.
Instructora	Es la encargada de chequear la realización de la cuarterería y entregar la evaluación correspondiente para que ésta sea archivada.
Profesor	Es el encargado de chequear la realización de la guardia estudiantil y entregar dicha evaluación para que sea archivada.

**Tabla 2.2 Descripción de los trabajadores del negocio**

### 2.4 Diagrama de casos de uso del negocio

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

A continuación se muestra la figura 2.1 correspondiente al diagrama de casos de uso del negocio.



### **Figura 2.1 Diagrama de Casos de Uso del Negocio**

Los diagramas de actividades asociados a cada caso de uso, así como la descripción textual y el modelo de objetos se pueden encontrar en el Anexo 1

#### **2.5 Requerimientos funcionales**

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

- ✓ R 1 Autenticar Usuario.
  - R 1.1 Definir niveles de acceso.
  - R 1.2 Gestionar usuarios.
    - R 1.2.1 Insertar usuario.
    - R 1.2.2 Eliminar usuario.
  
- ✓ R 2 Generar cuartelería.
  - R 2.1 Mostrar cuartelería.
  - R 2.2 Imprimir cuartelería.
  - R 2.3 Insertar edificios.
  - R 2.4 Insertar apartamentos.
  - R 2.5 Insertar estudiantes en apartamentos.
  - R 2.6 Eliminar apartamentos.
  - R 2.7 Eliminar estudiantes.
  
- ✓ R 3 Generar guardia estudiantil.
  - R 3.1 Mostrar guardia estudiantil.
  - R 3.2 Imprimir guardia estudiantil.
  - R 3.3 Insertar grupos.
  - R 3.4 Insertar profesores.
  - R 3.5 Eliminar profesores.

✓ R 4 Reportes

R 4.1 Ver la cuartelería.

R 4.2 Ver la guardia.

R 4.3 Buscar información.

R 4.3.1 Buscar información simple.

R 4.3.2 Buscar información avanzada.

## 2.6 Requerimientos no funcionales

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto.

A continuación se muestran los requerimientos no funcionales:

- Apariencia o interfaz externa

La interfaz no contiene muchas imágenes para no demorar las respuestas al usuario. El diseño de la interfaz es sencillo y claro de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Es formal, serio y con una navegación sugerente, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación.

- Usabilidad

El sistema ha sido diseñado de tal forma que permite a los usuarios acceder a la aplicación Web mediante el dominio UCI. Requiere tener el JavaScript activado.

- Rendimiento

La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve (de segundos), aunque en algunos momentos como se tiene que conectar al servidor LDAP esto puede demorar un poco.

- Soporte

Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación.

- Ayuda y documentación en línea

El sistema brinda a los usuarios una buena ayuda en línea de modo de si el usuario presenta algún problema pueda acudir al mismo, además cuando el usuario desea realizar una búsqueda existe una ayuda disponible.

- Software

Para el funcionamiento del sistema en el servidor es necesario el S.O Windows 98 o superior, Linux o Unix, en sus versiones de S.O servidores. Para el funcionamiento del sistema en las terminales cliente es necesario el S.O Windows 98 o superior, Linux o Unix.

- Hardware

Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior.

- Portabilidad

El producto es usado bajo los S.O Windows, Linux y Unix. El producto corre sobre una plataforma Web, codificada en "PHP" y sus sistemas de bases de datos en PostgreSQL.

- Seguridad

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios al sitio, de identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema y de asignarles el privilegio que le corresponde. Garantiza que la información sea insertada y eliminada únicamente por quien tiene derecho a hacerlo.

- Confidencialidad

Toda la información está protegida del acceso no autorizado, los administradores de sistema son los únicos que podrán insertar o eliminar la información.

- Disponibilidad

Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento (si tiene permiso para ello).

- Políticos-Culturales

Toda modificación al funcionamiento establecido en los requerimientos será realizada por la Dirección de la Facultad # 2 (VD Extensión).

- Restricciones en el diseño y la implementación

Es una aplicación Web desarrollada con la tecnología para creación de páginas Web dinámicas PHP y base de datos en PostgreSQL.

- Confiabilidad

La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

- Restricciones

Se utiliza UML para lograr una mejor documentación del sistema y como herramienta de apoyo Rational Rose. Se utiliza como lenguaje de programación PHP y el gestor de base de datos PostgreSQL.

## 2.7 Actores del sistema a automatizar

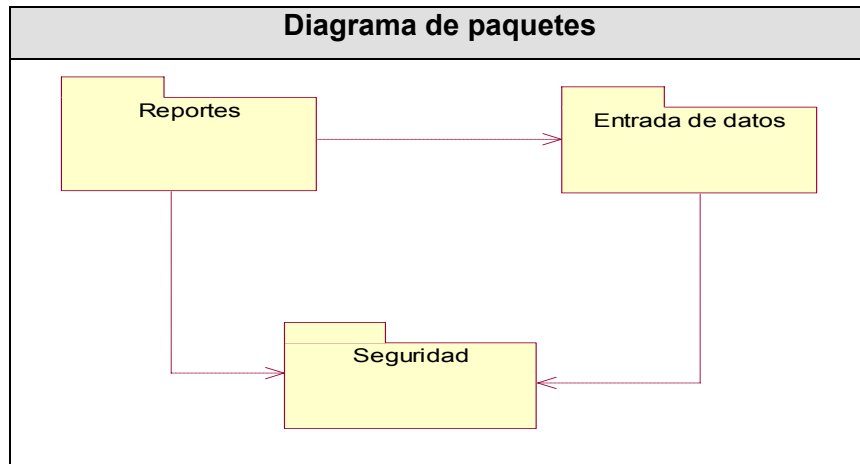
Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. En este caso los actores que interactúan con el sistema se definen a continuación en la tabla 3.1.

Actores	Justificación
Usuario	Es una generalización del administrador, lleva a cabo el proceso de autenticación.
Administrador	Puede realizar todas las acciones que realiza un usuario. Es una generalización del Vice Decano de Extensión y la Decana, además es el responsable de Insertar la información del estudiante, Generar la cuartelería y Generar la guardia estudiantil.

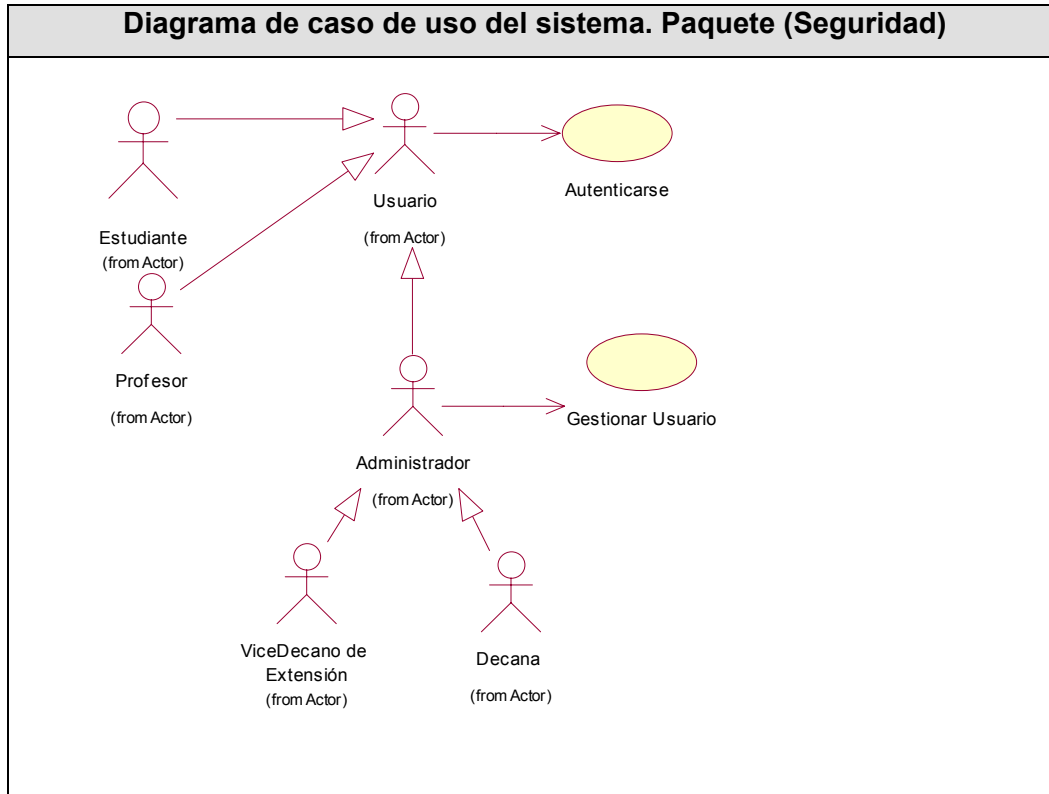


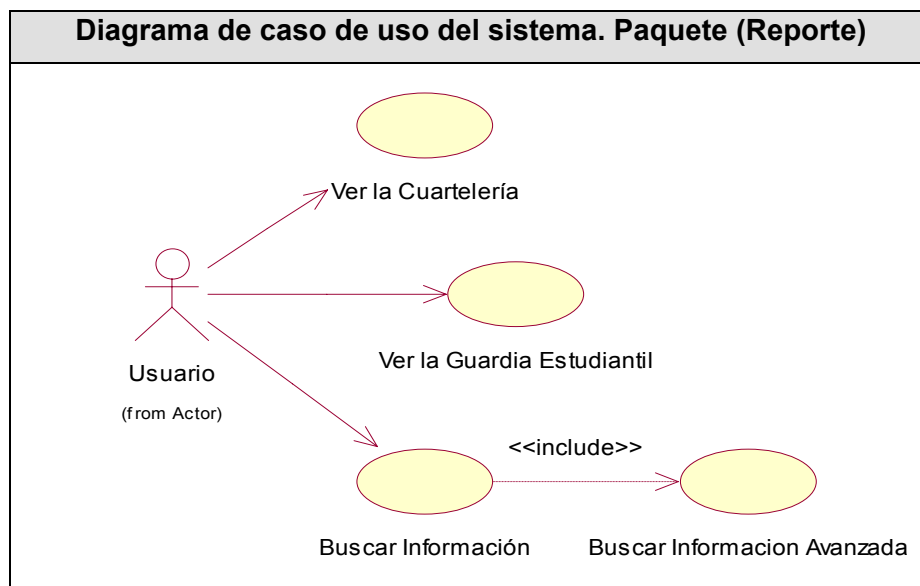
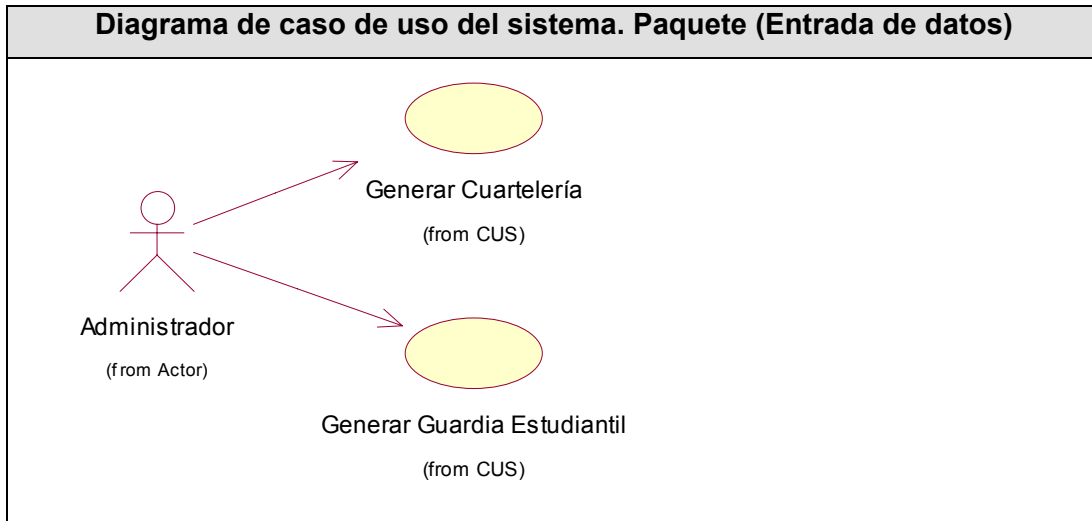
Vice Decano de Extensión	Es un administrador, que a su vez puede realizar todas las acciones que realiza un administrador.
Decana	Es un administrador, que a su vez puede realizar todas las acciones que realiza un administrador.

### 2.8 Paquetes y relaciones entre ellos



### 2.9 Diagrama de Caso de Uso del Sistema a automatizar





Las descripciones textuales asociadas a los casos de uso críticos del sistema se pueden encontrar en el Anexo 2.

## 2.10 Conclusiones

En este capítulo se identificaron los roles, entidades u objetos del negocio, así como su relación con este proceso. Esta descripción fue realizada mediante el modelo del negocio, para lo cual se elaboraron los modelos de casos de uso y objetos del negocio.

Después de haber realizado el modelado de negocio, se pudo lograr una mejor comprensión del problema que el sistema tiene que resolver. Además se comenzó a

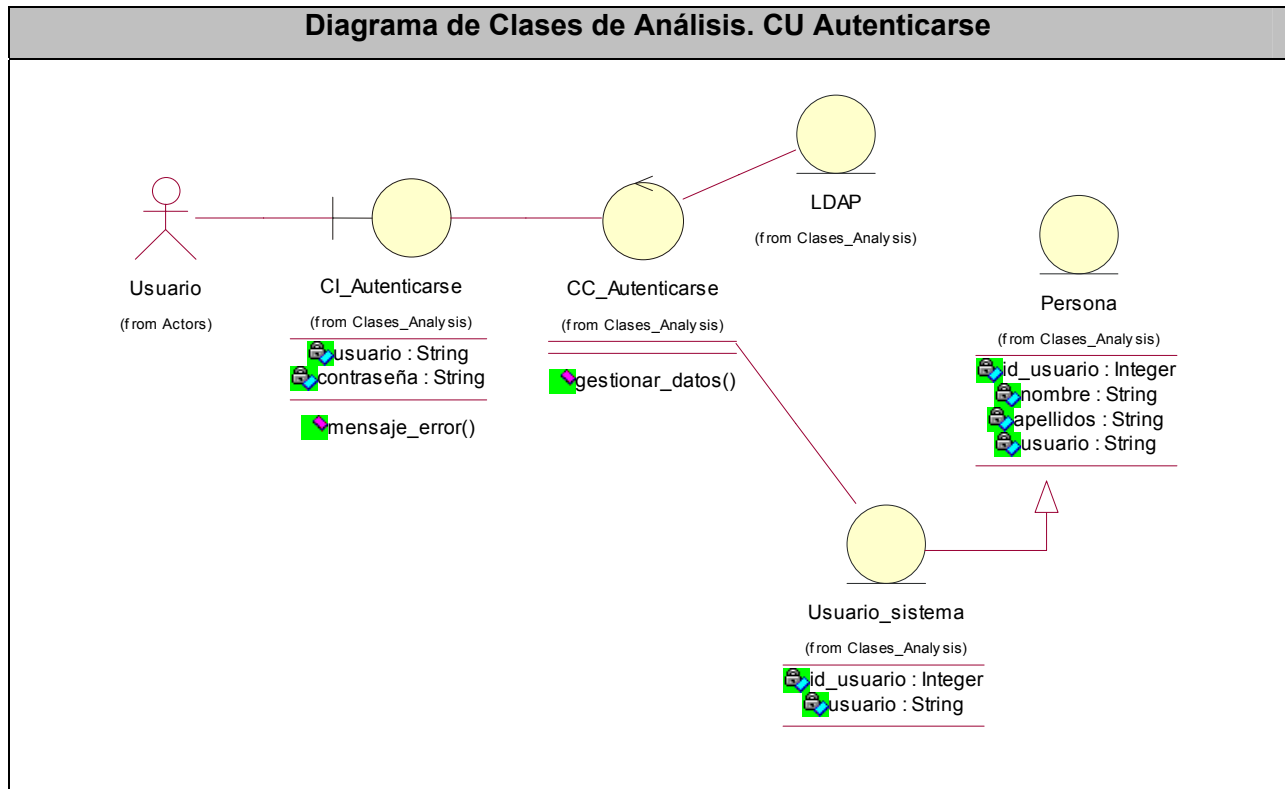
desarrollar la propuesta de solución, obteniéndose a partir del análisis de los procesos del negocio, un listado con las principales funcionalidades que debe tener el sistema y los requisitos adicionales, se representaron los diagramas de casos de uso del sistema, y finalmente se describieron las acciones de los actores del sistema con los casos de uso con los que interactúan. Gracias a esto, ahora se puede empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funcionalidades que han sido consideradas necesarias en este capítulo.

## CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL SISTEMA

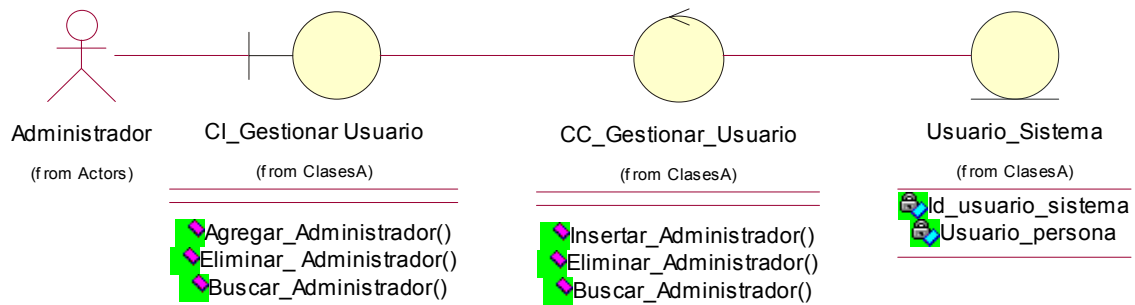
### 3.1 Introducción

Tras la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto, se hace necesario definir cómo se desarrollará. Este capítulo tiene el objetivo de plantear la concepción general del análisis y diseño del sistema propuesto. Así, se presentan los diagramas de clases de análisis, se estructura la información que se desea persista a través del diseño de la base de datos y la descripción de las tablas.

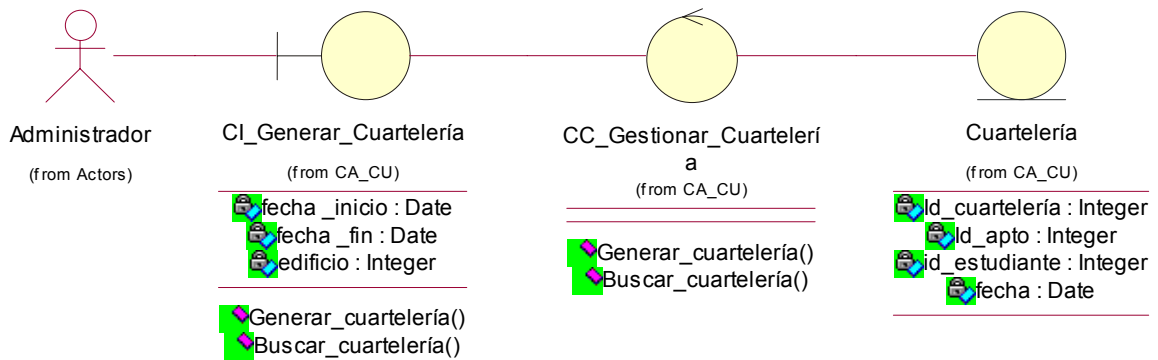
### 3.2 Diagramas de clases del análisis.



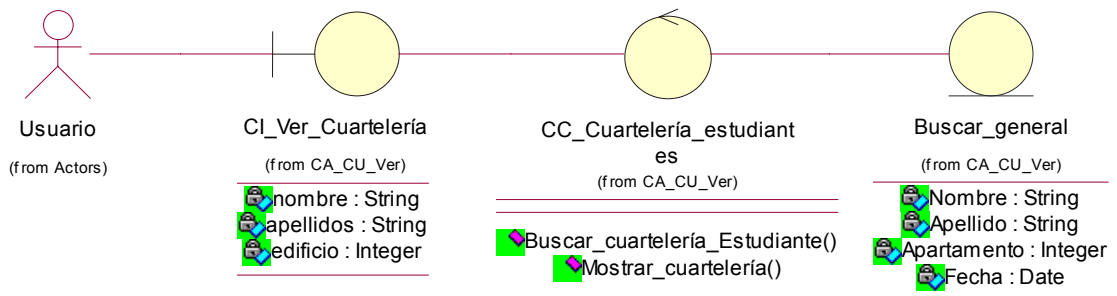
**Diagrama de Clases de Análisis. CU Gestionar Usuario**

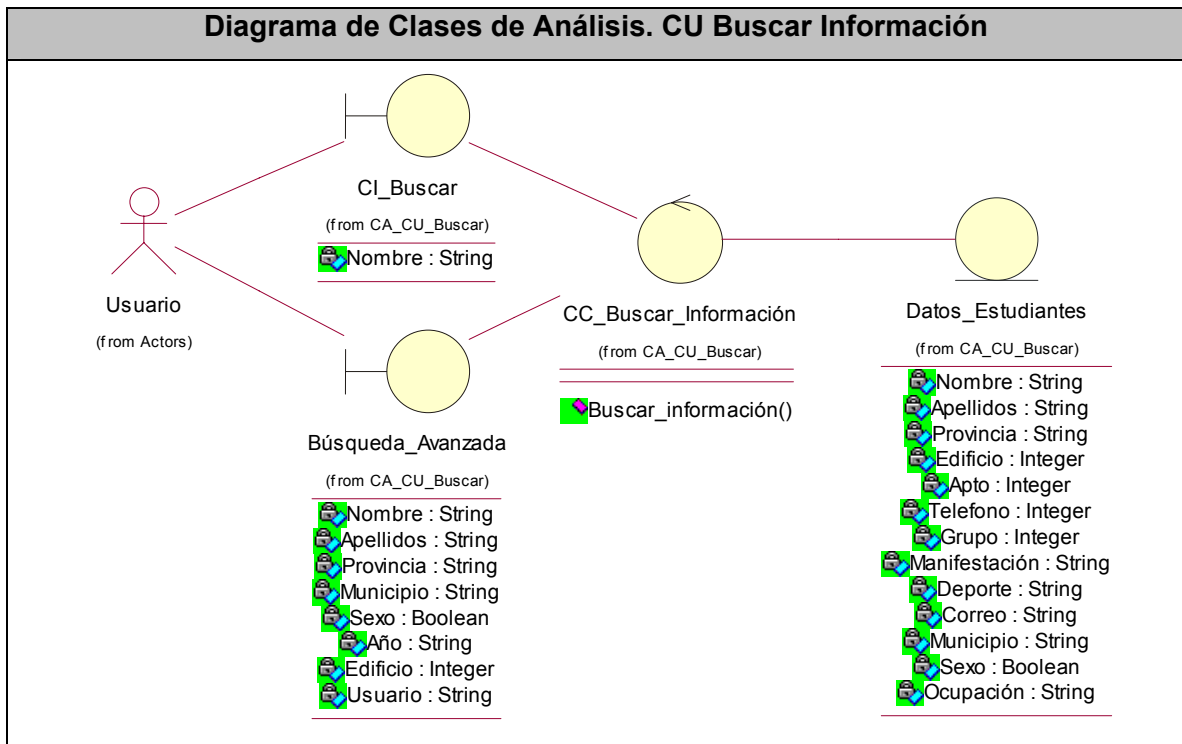
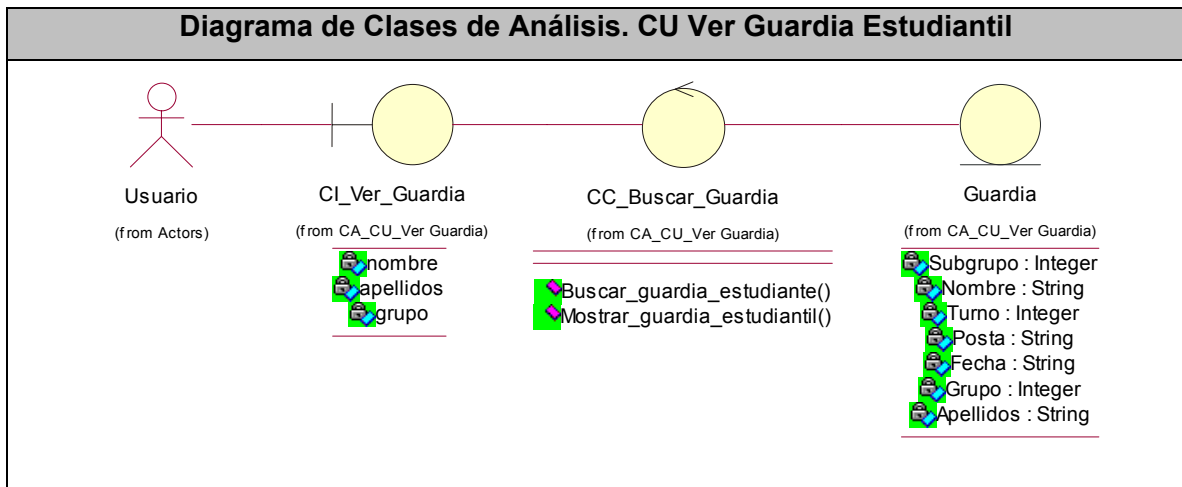
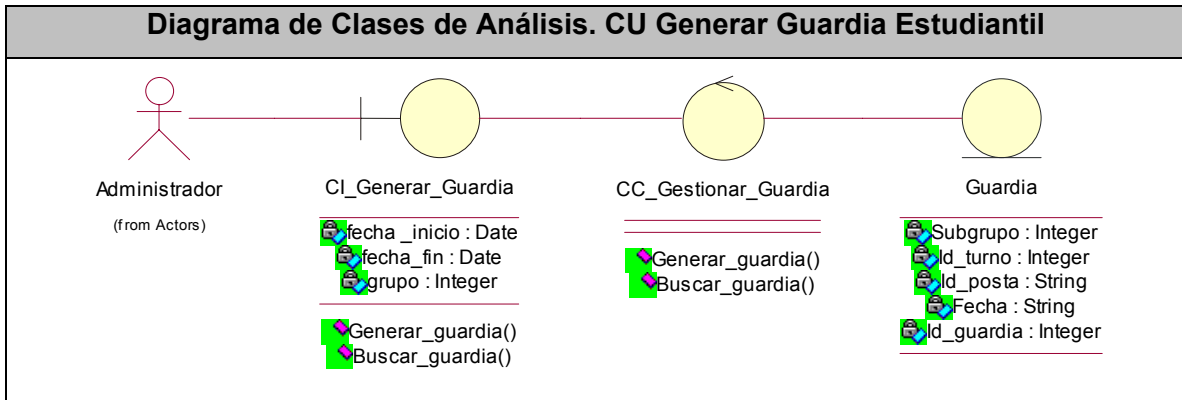


**Diagrama de Clases de Análisis. CU Generar Cuartelería**

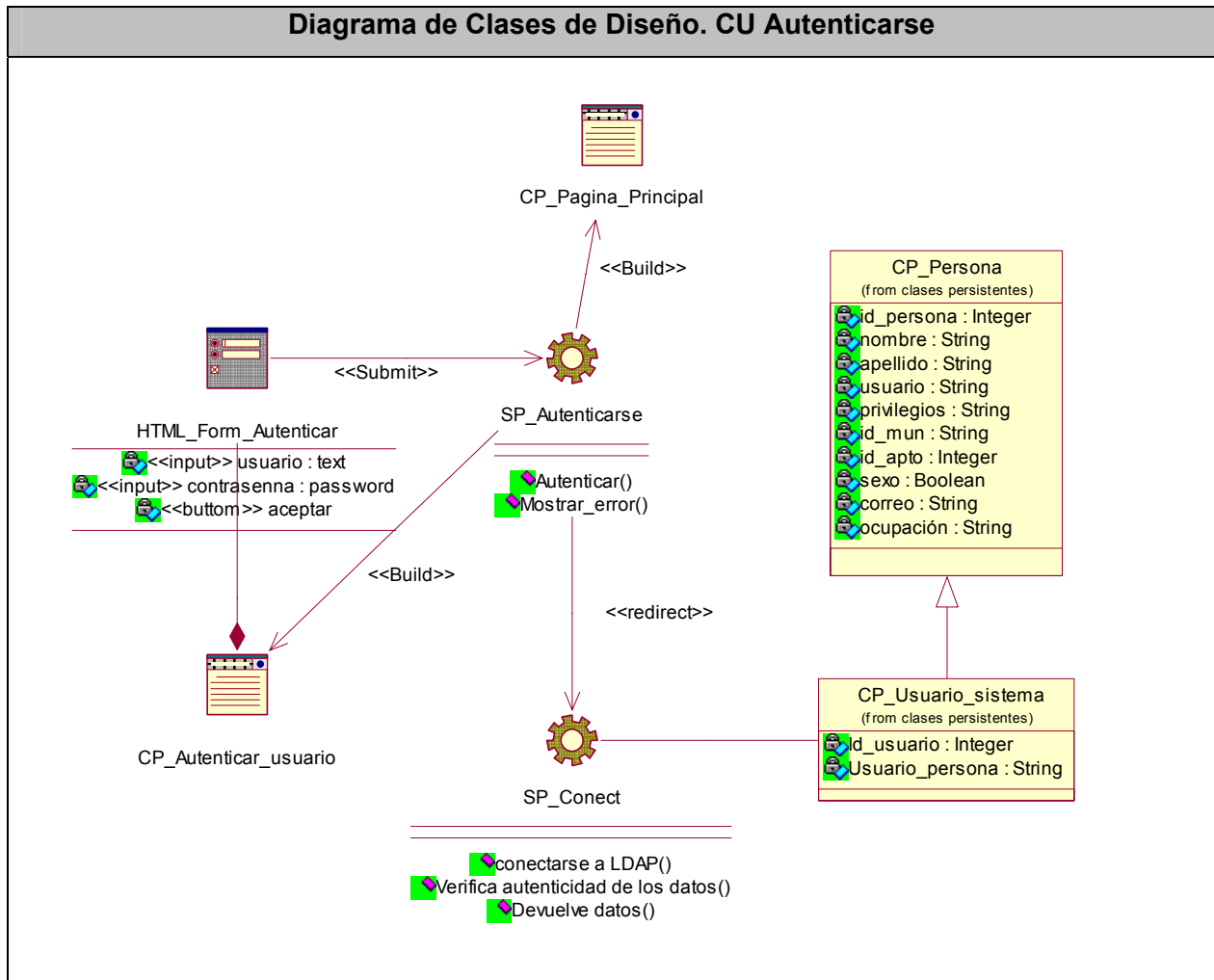


**Diagrama de Clases de Análisis. CU Ver Cuartelería**

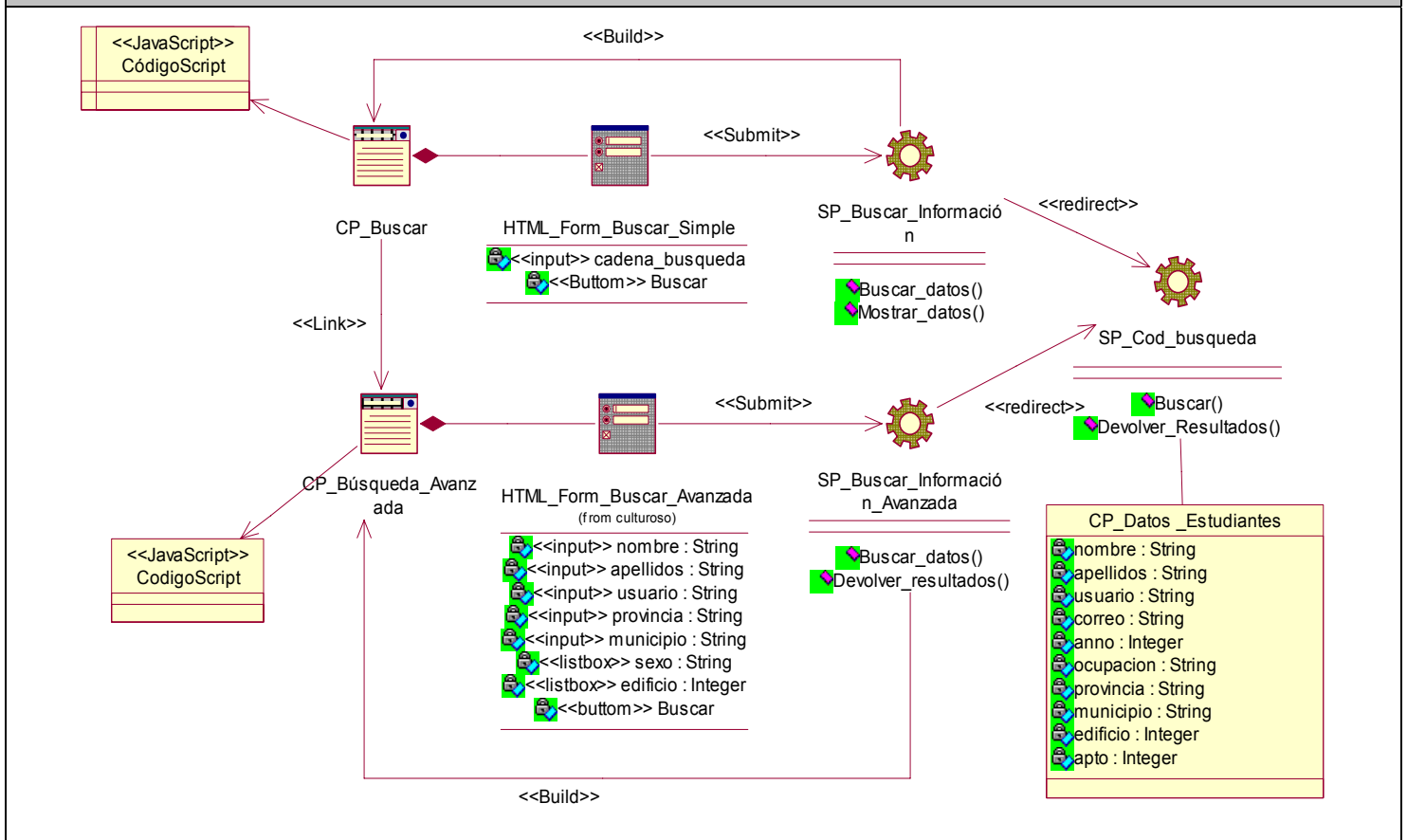




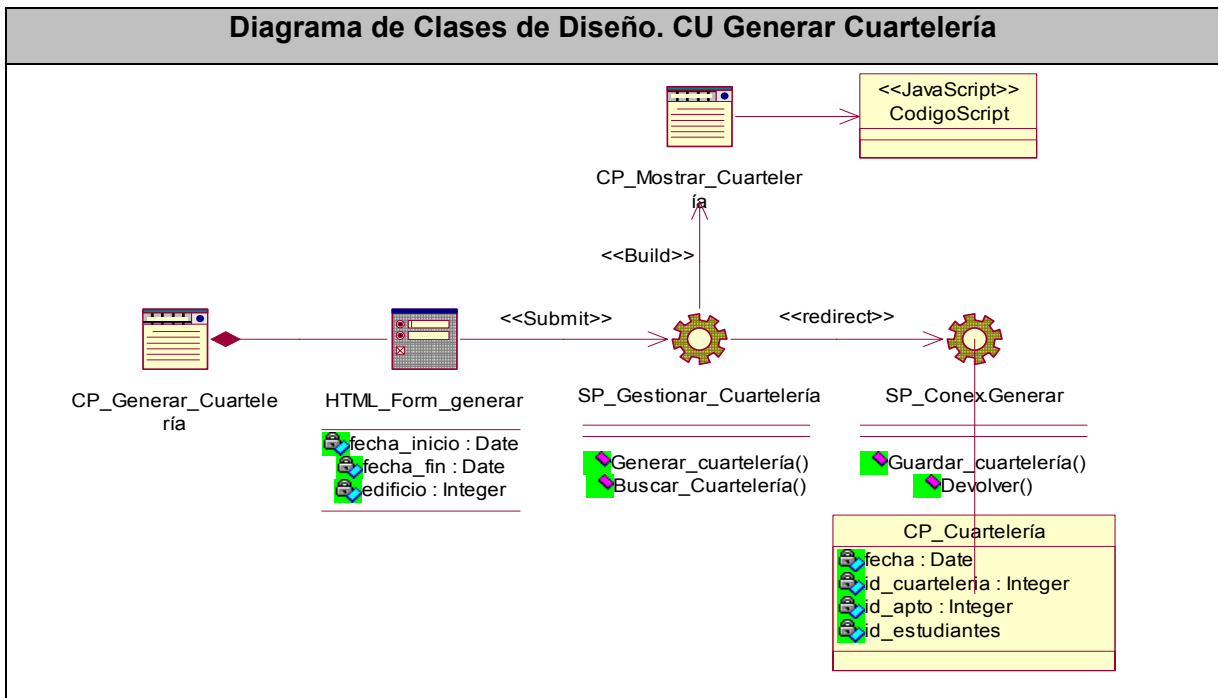
3.3 Diagramas de clases del diseño.



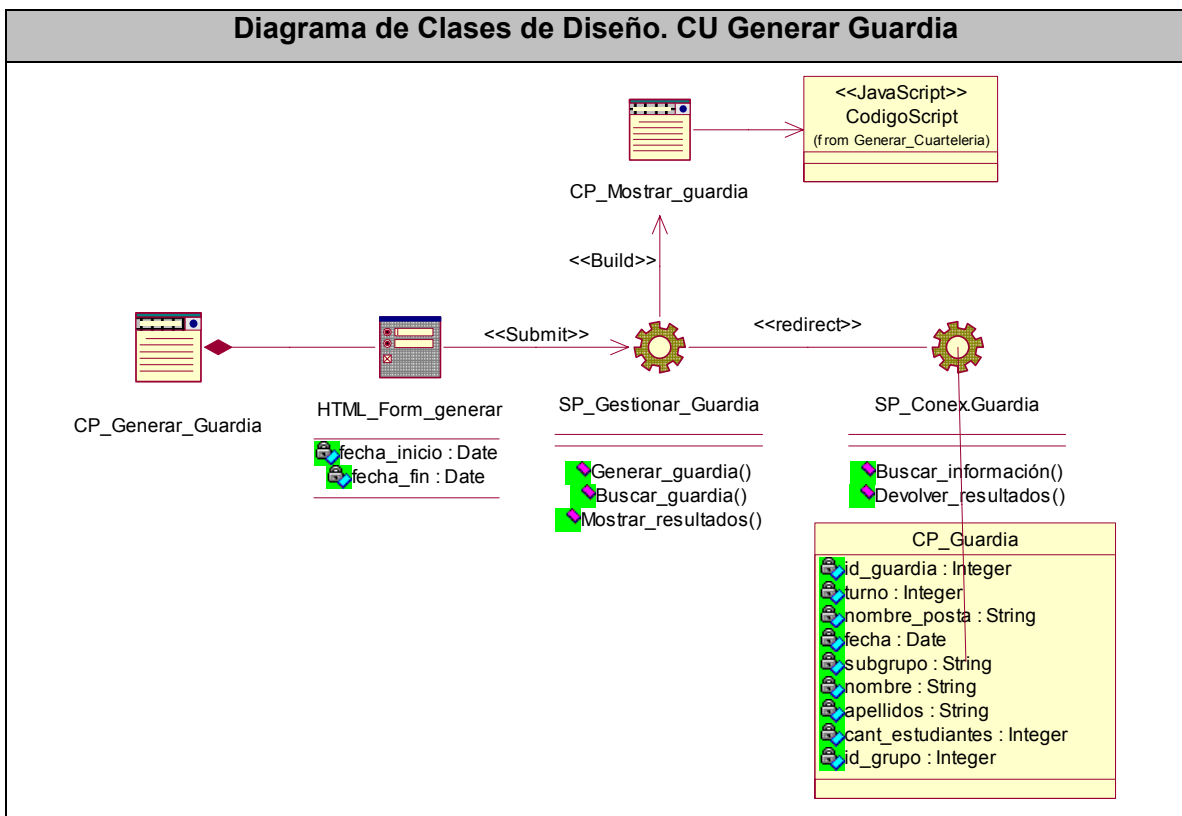
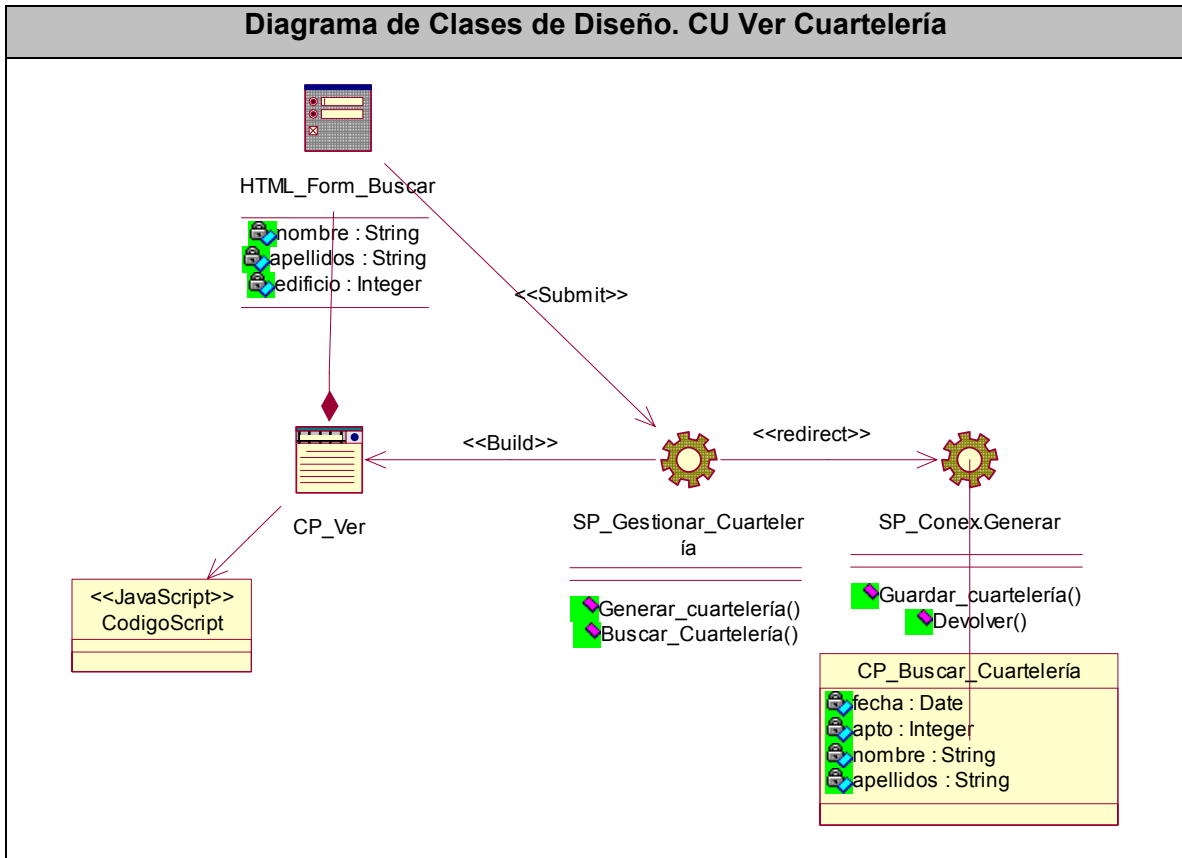
**Diagrama de Clases de Diseño. CU Buscar Información**

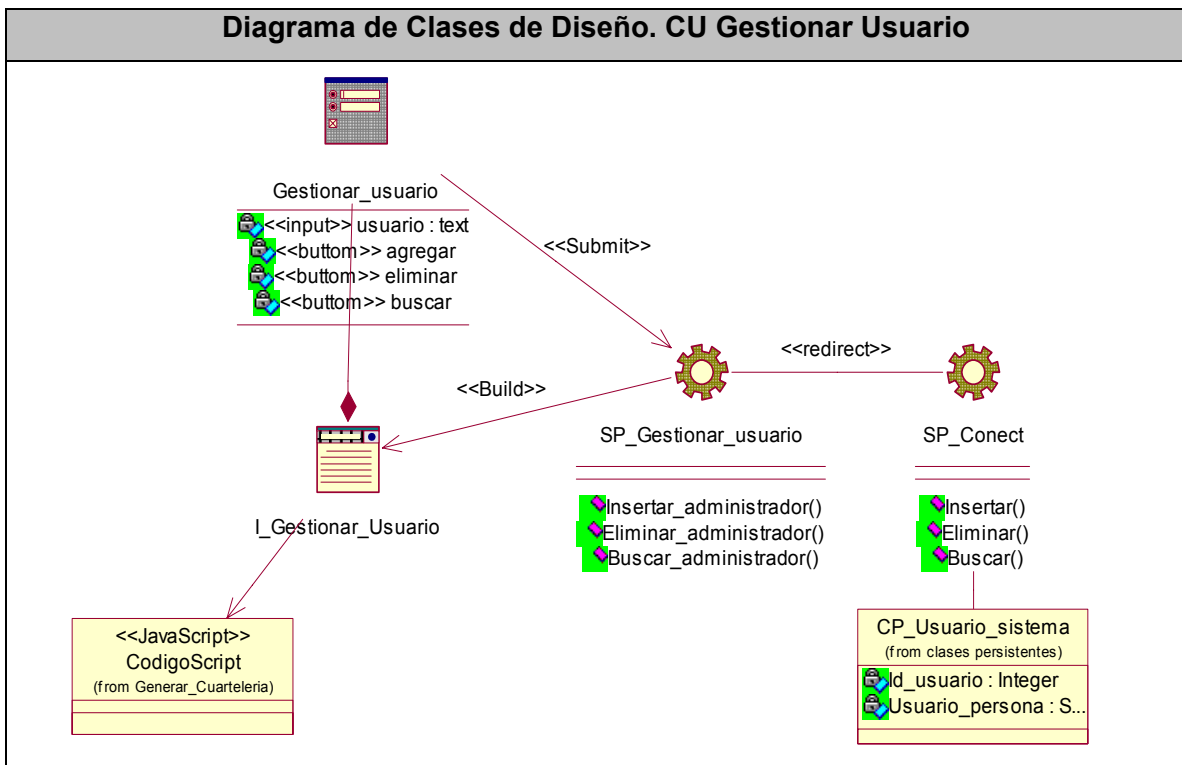
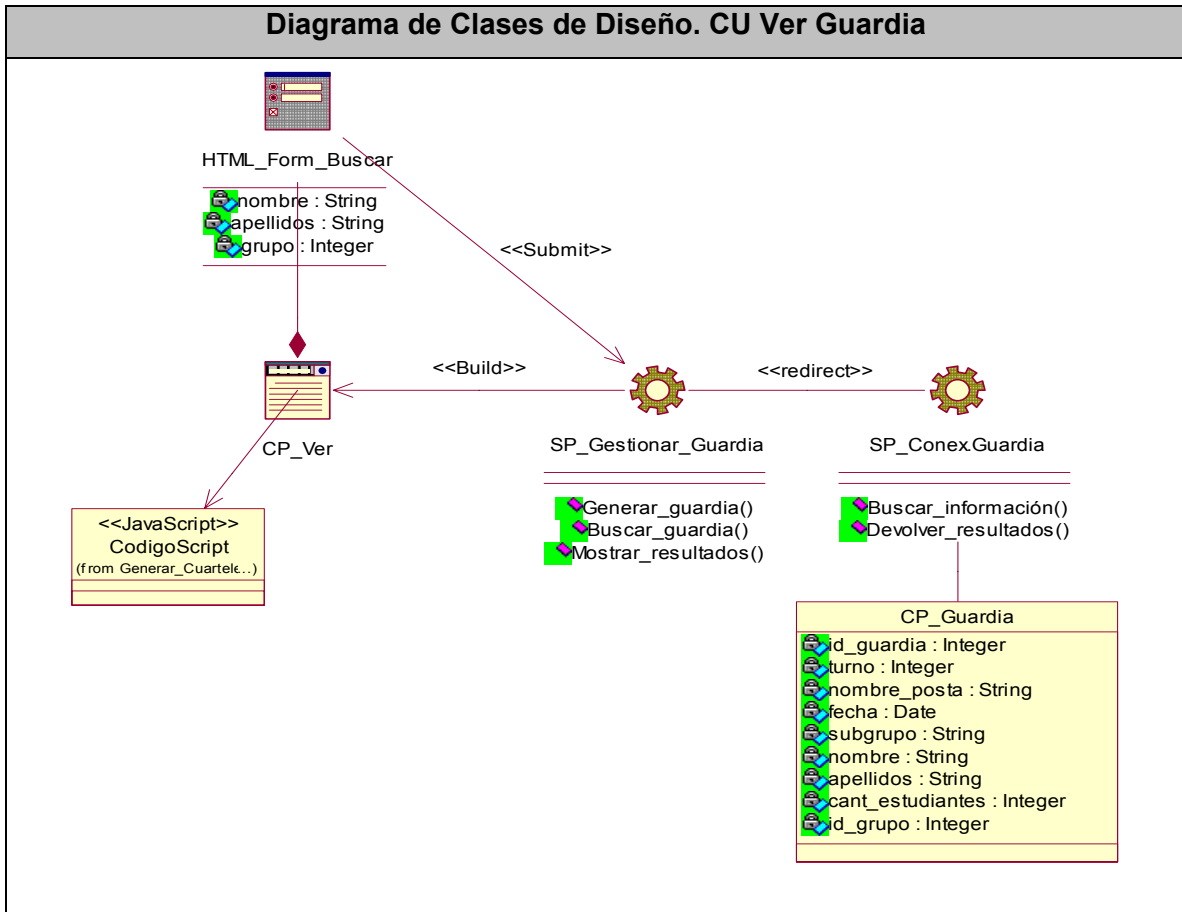


**Diagrama de Clases de Diseño. CU Generar Cuartería**



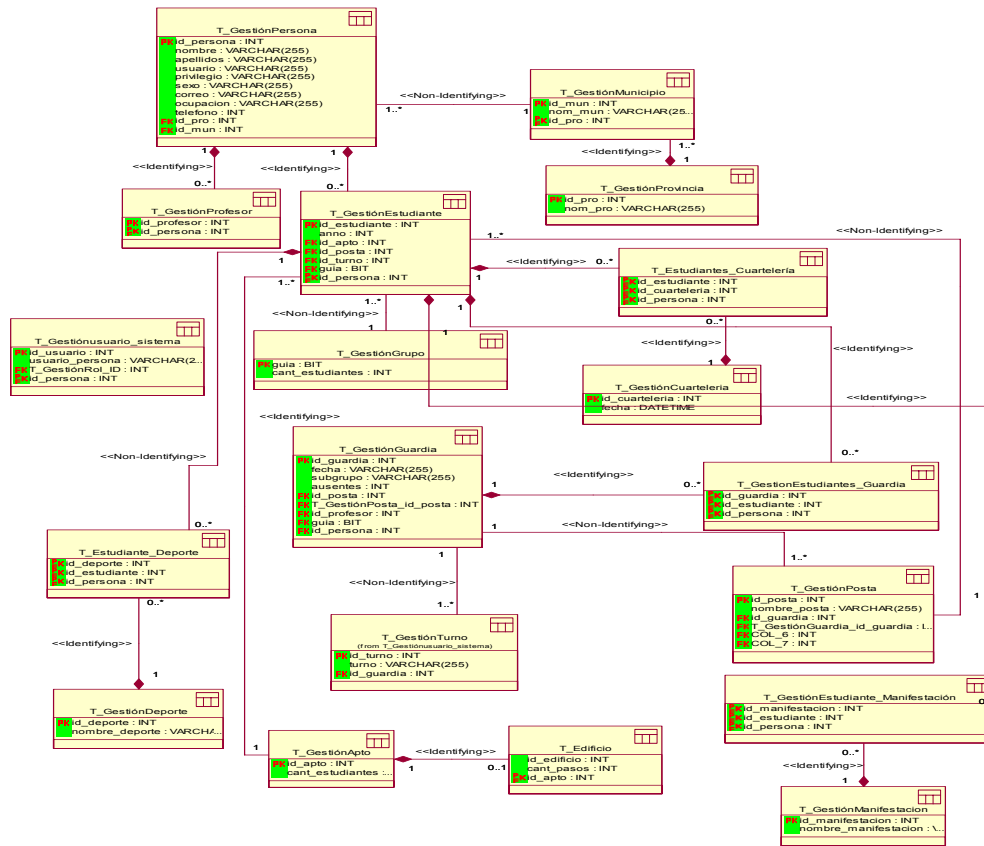








### 3.4.2 Modelo de datos



### 3.5 Conclusiones

En el presente capítulo se desarrollaron los diagramas de clases de análisis y diseño, diagramas de clases persistentes y el diseño de la base de datos del sistema.

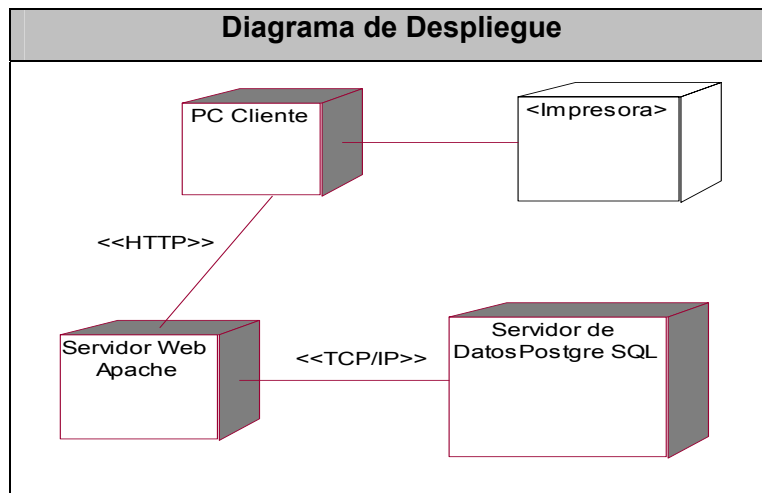
## CAPÍTULO 4 IMPLEMENTACIÓN

### 4.1 Diagrama de despliegue

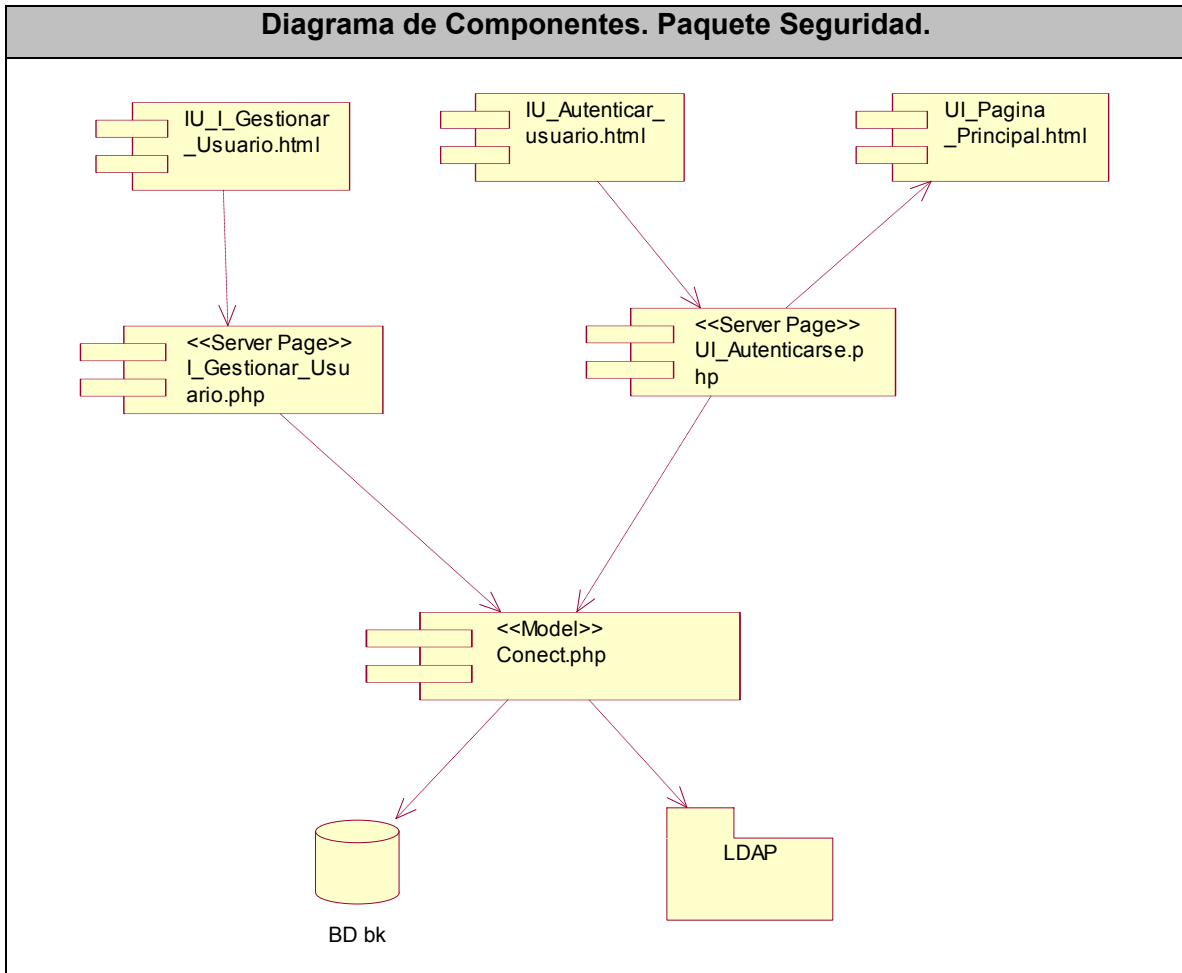
El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

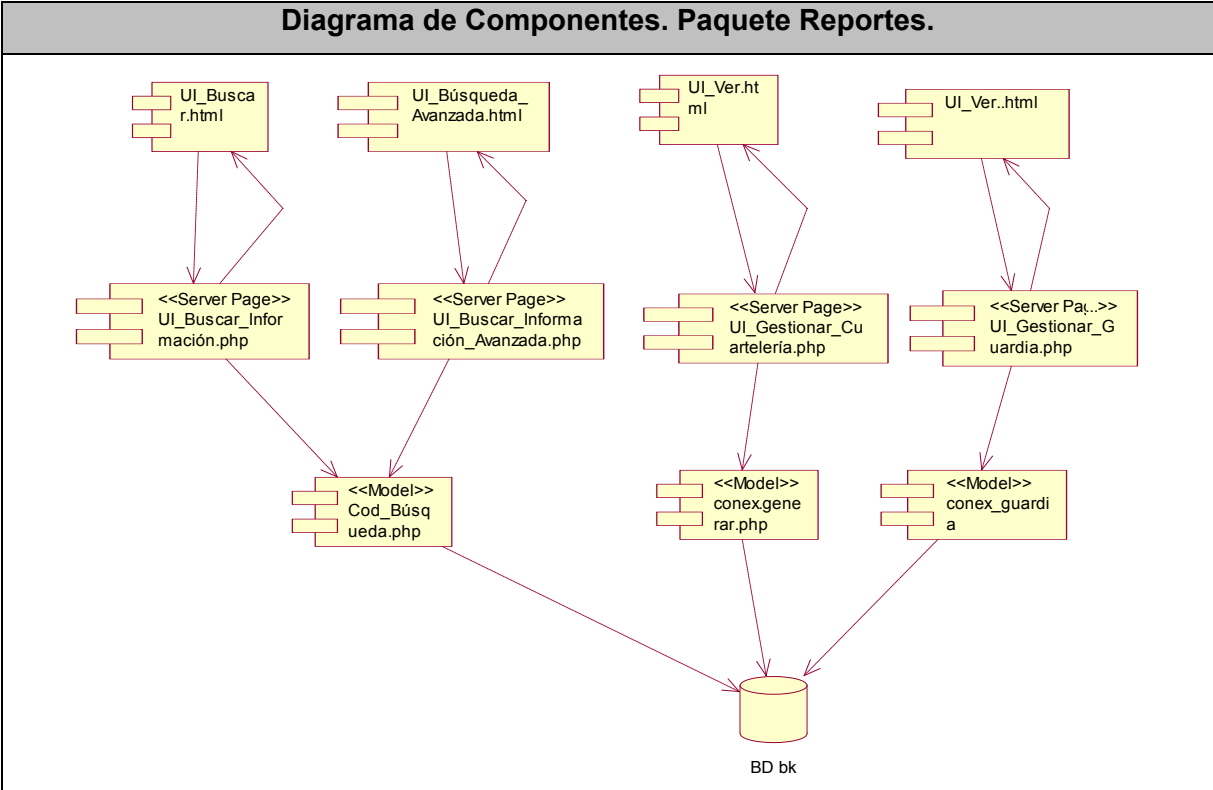
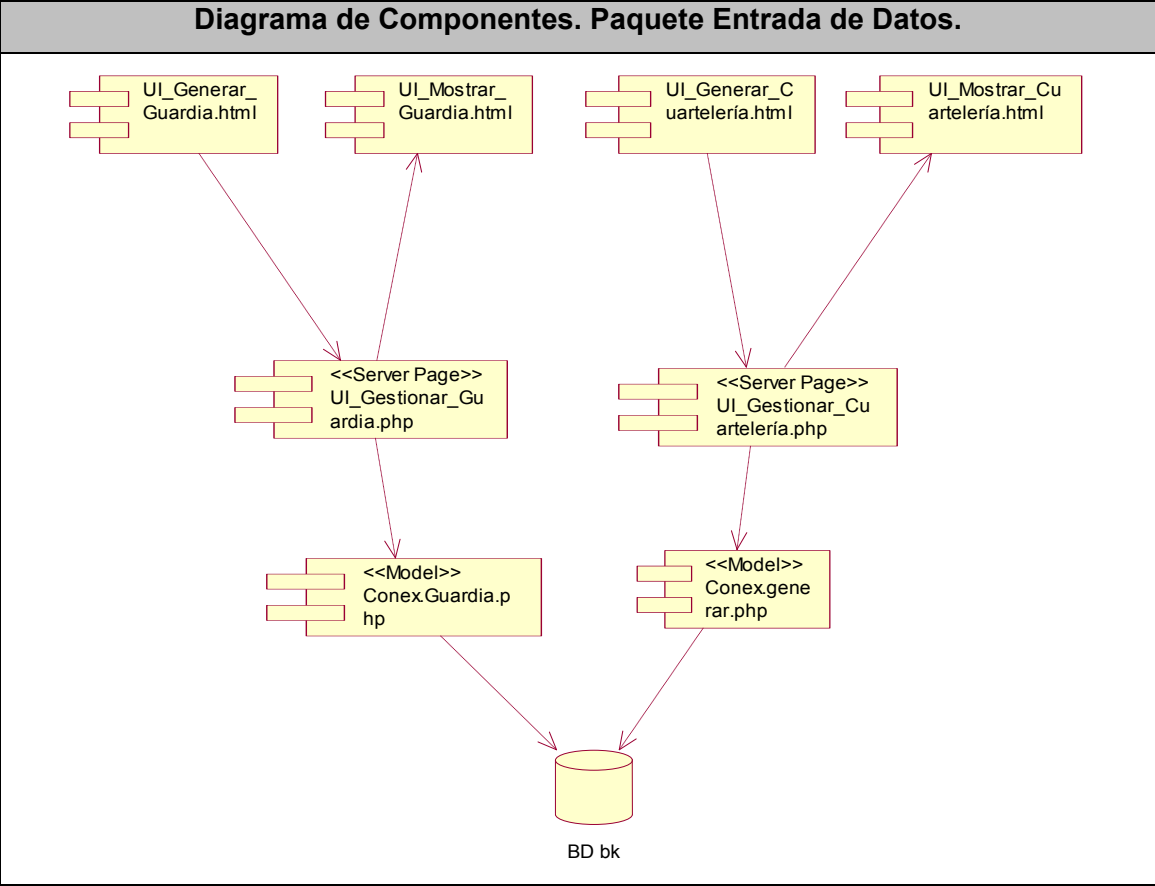
Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema.



## 4.2 Diagramas de componentes por paquetes





## CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD

### 5.1 Introducción

Para la realización de un proyecto es de suma importancia el análisis del costo y los beneficios que reportará. Como resultado de este análisis se obtiene el tiempo de desarrollo en meses, costo y la cantidad de personas que se necesitan para desarrollar el proyecto.

### 5.2 Estimación del esfuerzo

El método de estimación mediante el Análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de estos factores.

#### Cálculo de Puntos de Caso de Uso sin Ajustar

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente formula:

$$UUCP = UAW + UUCW$$

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.

#### Factor de Peso de los Actores sin ajustar (UAW)

Para conocer este valor se hacen cálculos mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo Actor	Descripción	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	5	15
----------	--	---	---	----

$$UAW = S(\text{Factor} * \text{Actores})$$

$$UAW = 15$$

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

El valor se obtiene mediante un análisis de la cantidad de Casos de Uso existentes 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.

Tipo de CU	Descripción	Peso	Cantidad	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	16	80
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	2	20
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

$$UUCW = \text{Sumatoria}(\text{Factor} * \text{CantCU})$$

$$UUCW = 100$$

$$\text{Entonces: } UUCP = UAW + UUCW = 115$$

**Cálculo de Puntos de Casos de Uso ajustados.**

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

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

### Factor de complejidad técnica (TCF).

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

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	2
T3	Eficiencia del usuario final	1	5	1
T4	Funcionamiento Interno complejo	1	4	1
T5	El código debe ser reutilizable	1	3	2
T6	Facilidad de instalación	0,5	0	0
T7	Facilidad de uso	0,5	5	1
T8	Portabilidad	2	4	4
T9	Facilidad de cambio	1	1	1
T10	Concurrencia	1	3	0
T11	Incluye objetivos especiales de seguridad	1	0	2
T12	Provee acceso directo a terceras partes.	1	3	0
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1

Sumatoria = 33,5

$$TCF = 0.6 + 0.01 * \text{Sumatoria}(\text{Peso} * \text{Valor})$$

$$TCF = 0,935$$

### Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo.

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	4	3
E2	Experiencia en la aplicación	0,5	4	1
E3	Experiencia en la orientación a objetos.	1	3	2
E4	Capacidad del analista líder.	0,5	4	4.5
E5	Motivación.	1	5	4
E6	Estabilidad de requerimientos	2	4	2
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	3	-1

Sumatoria = 23

$EF = 1.4 - 0.03 * \text{Sumatoria}(\text{Peso} * \text{Valor})$

$EF = 0,71$

Factor de conversión

$CF = 20$

$UCP = UUCP * TCF * EF$

$UCP = 76,34275$

$E = UCP * CF$

$E = 1526,855$

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	305,371
Diseño	25	763,4275
Implementación	50	1526,855
Pruebas	10	305,371
Sobrecarga (otras actividades)	5	152,6855
Total	100	3053,71

Esfuerzo Total (Horas--Hombre)	ET1	3053,71
Esfuerzo Total (Mes--Hombre)	ET2	12,7237917
Salario Promedio	SM	100
Cantidad de Hombres	CH	2

Costo Hombre--Mes	CHM	750
Costo Total	Costo	1272

Aunque el costo de realizar la aplicación es bastante, son muchos los resultados que se obtendrán. Todo software siempre tiene un costo asociado, aunque puede compensarse con los beneficios que se obtienen. Con este software se logrará automatizar la gestión de la guardia estudiantil y de la cuartería en la facultad, además cuenta con un buscador donde solo se encontrará información relacionada con los estudiantes de nuestra facultad.

### 5.3 Beneficios tangibles e intangibles

El sistema automatizado para la gestión de la información del Área de Extensión Universitaria no es un producto con fines comerciales, su objetivo principal es resolver los problemas que tienen relación con el VD Extensión, entre los que se encuentran el control y gestión de la guardia estudiantil y la cuartería.

El beneficio fundamental del sistema es contar con una aplicación Web flexible, dinámica y de interfaz agradable que permite insertar, eliminar, generar, y realizar búsquedas de una forma precisa y en un tiempo relativamente corto.

Por todo esto observamos que los beneficios inmediatos son generalmente intangibles:

- ✓ Disminución del tiempo y esfuerzo que se invierte en esta tarea que se realiza, en la actualidad de forma manual.
- ✓ Disminución de la acumulación de materiales impresos relacionados con los procesos de cuartería y guardia estudiantil.
- ✓ Fácil y rápido acceso de información actualizada.
- ✓ Fácil procesamiento de la información y obtención, dinámica, de reportes de la cuartería y la guardia estudiantil.

### 5.4 Análisis de costos y beneficios

Una vez que se halla implantado el sistema este contribuirá a aumentar la eficiencia de los procesos que se llevan a cabo en el Área de Extensión Universitaria de la facultad, pues se disminuirá el tiempo y esfuerzo que se invierte en la realización de estos procesos que en la actualidad se realizan manualmente.

Llevar a cabo el desarrollo de un proyecto cuesta, es por ello que para desarrollarlo es necesario tener presente los beneficios que reportarían su implantación y desarrollo. La tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no hay que incurrir en gastos en el pago de licencias de uso. El sistema es portable por lo que un cambio de plataforma para la implantación del mismo es viable y factible, y no hay que incurrir en muchos cambios; debido a la estructuración en capas de los procesos del negocio que se diseñaron.

Analizando el costo del proyecto y los numerosos beneficios que reporta la realización del proyecto, se puede concluir que su implementación es realmente factible.

### **5.5 Conclusiones**

En este capítulo se analizó la factibilidad durante el desarrollo del sistema, calculándose, el costo que arrojó la producción del mismo, el tiempo que consumió para su culminación, la cantidad de personas involucradas y a su vez el esfuerzo realizado por este equipo de desarrollo, haciendo finalmente todos estos cálculos tanto cualitativos como cuantitativos, prevalecer criterios favorables de que el sistema influye positivamente en el desarrollo sostenible en todos los puntos de vistas valorados.

## CONCLUSIONES

- ✚ Después de realizado el trabajo se espera que el documento haya servido para la comprensión teórica de la situación problemática existente y su solución, así como el desarrollo de las diferentes etapas de la aplicación usando la metodología RUP.
- ✚ Se alcanzó, satisfactoriamente, el objetivo propuesto: desarrollar un sistema informático robusto, flexible que controle y gestione la información del VD Extensión.
- ✚ Se dio cumplimiento también a los objetivos específicos que se trazaron al inicio:
  - ✓ Se realizó una aplicación Web de fácil acceso y con una interfaz cómoda que facilita la obtención de los datos de los estudiantes de la facultad # 2.
  - ✓ Se automatizó la gestión de la guardia y la cuartelaría en la Facultad # 2.
  - ✓ Se diseñó e implementó una base de datos capaz de almacenar de manera organizada la información que se manipula durante estos procesos.
  - ✓ Se desarrolló el análisis y diseño de los procesos seleccionados.
  - ✓ Se implementó el sistema con las características definidas en los procesos de análisis y diseño.
- ✚ En fin se han seguido los principios básicos de diseño descritos para el desarrollo del sistema y se ha logrado un alto nivel de seguridad y protección de de la información almacenada.

## RECOMENDACIONES

Se recomienda:

- ✚ 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 actividad que se está gestionando.
- ✚ Que se haga uso de los Web Services ya que los mismos brindan todos los datos de los estudiantes que se encuentran en la Base de Datos UCI.
- ✚ Que se le añadan nuevas funcionalidades al caso de uso del sistema Generar Guardia, como por ejemplo que una vez generada la guardia la primera vez, el administrador no tenga que la próxima vez entrar un período determinado sino que esto se realice de forma automática y que el solamente realice una búsqueda por fecha , por grupo e informe a los estudiantes y profesores el día que le corresponde la realización de la misma.
- ✚ Que se le añadan nuevas funcionalidades al caso de uso del sistema Generar Cuartelería, como por ejemplo que después de generada la cuartelería la primera vez, el administrador no tenga que la próxima vez entrar un período determinado sino que esto se realice de forma automática y que el solamente realice una búsqueda por fecha ,por edificio ,por paso de escalera e informe a los estudiantes cuando le corresponde.
- ✚ Que se realice continúe el estudio de esta tesis que es tema muy interesante y que siempre puede tener nuevos detalles para seguir desarrollando

## REFERENCIAS BIBLIOGRÁFICAS

- [1] *Programación Web*. [Disponible en: <<http://www.arsys.es/soporte/programacion/windows.htm>> [Fecha de consulta 10 Febrero 2007].
- [2] Valido, Y. y Moreira, Y. *SAIMM: Sistema de Apoyo Integral a la Misión Milagro*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2005.
- [3] Méndez, L. y Torres, A. *Sistema de Promoción y Gestión Comercial para la oficina de Transferencia Tecnológica de la Universidad de Cienfuegos*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2005.
- [4] Cantero, J. *Un vistazo a PHP5 [I]*, julio 2004. Disponible en: <<http://libertonia.escomposlinux.org/story/2004/7/15/115328/134>>. [Fecha de consulta 10 febrero 2007].
- [5] *Macromedia Dreamweaver MX 2004*. Getting Started. Ayuda. Macromedia, Inc. 2003. [Fecha de consulta 15 febrero 2007].
- [6] Jacobson, I.; Booch, G. y Rumbaugh, J. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley. 2000.
- [7] *OMG Unified Modeling Language Specification*. OMG, INC. 2003.
- [8] *AJAX un nuevo acercamiento a aplicaciones Web*, mayo 28 del 2005. Disponible en: <<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>> [Fecha de consulta 15 febrero 2007].
- [9] AJAX. [Disponible en: <<http://es.wikipedia.org/wiki/AJAX>> [Fecha de consulta 15 febrero 2007].
- [10] *Clases de Ingeniería del Software I*, curso 2005-2006, UCI.



[11] *Introducción a PHP*. Disponible en: <[www.ciberteca.net/webmaster/php](http://www.ciberteca.net/webmaster/php)> [Fecha de consulta 15 febrero 2007].

[12] *PostgreSQL 8.1.x*. Disponible en: <<http://www.postgresql.cl/>> [Fecha de consulta 15 febrero 2007].

[13] *Tutorial de PostgreSQL*. Disponible en: <<http://es.tldp.org/Postgresqls/web/navegable/tutorial/tutorial.html>> [Fecha de consulta 17 febrero 2007].

[14] Peralta, Mario. *Estimación del esfuerzo basada en casos de uso*. Centro de Ingeniería del Software e Ingeniería del Conocimiento, Buenos Aires, Argentina.

[15] Welicki, L. *Patrones y Antipatrones: una Introducción –Parte II*. Disponible en: <[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317.asp](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317.asp)>. [Fecha de consulta 16 febrero 2007].

## Bibliografía

- ✓ Introducción a PHP. Disponible en: <[www.ciberteca.net/webmaster/php](http://www.ciberteca.net/webmaster/php)> [Fecha de consulta 15 febrero 2007].
  
- ✓ AJAX un nuevo acercamiento a aplicaciones Web, mayo 28 del 2005. Disponible en: <<http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php>> [Fecha de consulta 15 febrero 2007].
  
- ✓ Tutorial de PostgreSQL. Disponible en: <<http://es.tldp.org/Postgresqls/web/navegable/tutorial/tutorial.html>> [Fecha de consulta 17 febrero 2007].
  
- ✓ Programación Web. [Disponible en: <<http://www.arsys.es/soporte/programacion/windows.htm>> [Fecha de consulta 10 Febrero 2007].
  
- ✓ Jacobson, I.; Booch, G. y Rumbaugh, J. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley. 2000.
  
- ✓ *Clases de Ingeniería del Software I*, curso 2005-2006, UCI.

## **GLOSARIO DE TÉRMINOS Y SIGLAS**

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el desarrollo del sistema para la gestión de información del VD Extensión. Se trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo de un sistema para la gestión de información del área de extensión universitaria. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo de un sistema para la gestión de información del VD Extensión.

- **Administrador:** es la persona que tiene privilegios para determinadas funcionalidades del sistema.
- **APACHE:** es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

**AJAX:** Asynchronous JavaScript And XML.

- **COCOMO:** Modelo para la estimación de costos de productos informáticos.
- **CUN:** Caso de uso del negocio.
- **CUS:** Caso de uso del sistema.
- **HTML:** HyperText Markup Language. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información,

normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

- HTTP: HyperText Transfer Protocol. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.
- Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.
- Macromedia Dreamweaver MX: Herramienta para el desarrollo de aplicaciones Web de Macromedia. Combina en un único entorno de desarrollo accesible y potente las reconocidas herramientas de presentación visual de Dreamweaver, las características de rápido desarrollo de aplicaciones Web de Dreamweaver UltraDev y ColdFusion Studio, y el extenso soporte de edición de código de HomeSite. Ofrece una completa solución abierta para las tecnologías Web y estándares de hoy, incluyendo la accesibilidad y servicios Web.
- MVC: Modelo Vista Controlador.
- PHP: Hypertext Preprocessor. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.
- PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.
- RUP: Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.
- Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

- 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.
- UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software

## ANEXOS

### Anexo # 1

Diagramas de actividades asociados a cada caso de uso del negocio, así como la descripción textual y el modelo de objetos.

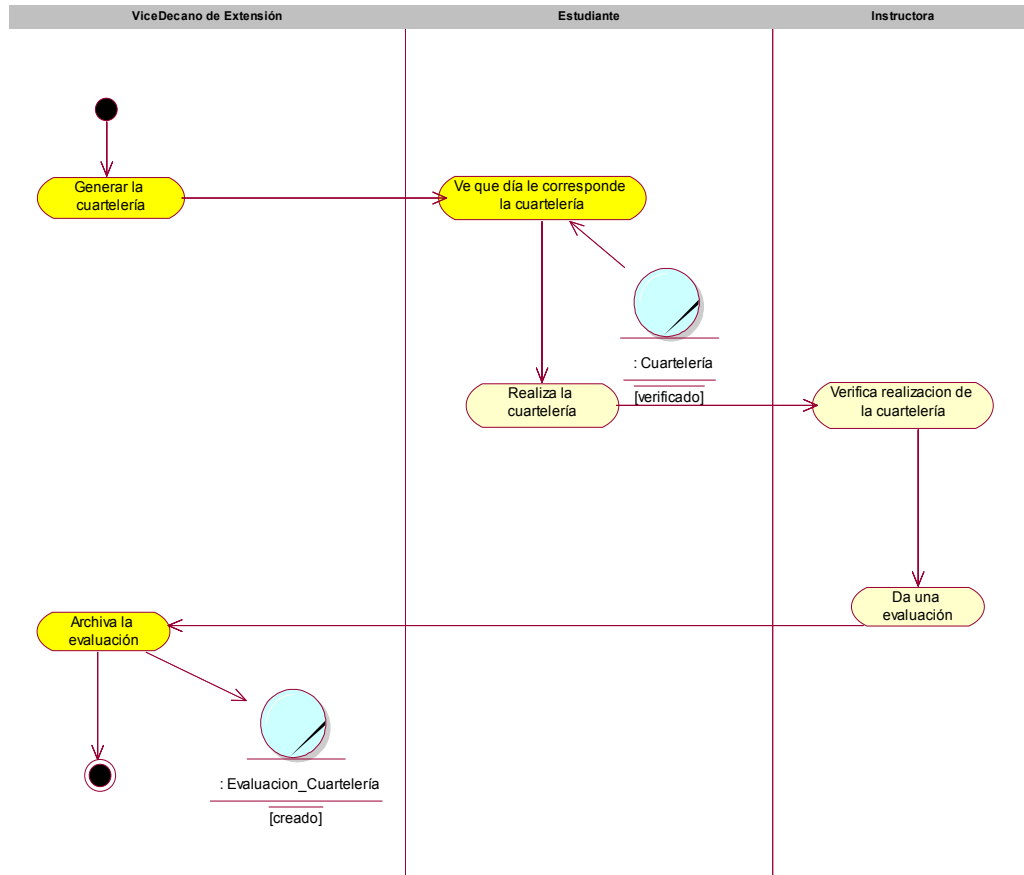
#### Caso de Uso del Negocio: Gestionar Cuartelería

##### Descripción textual

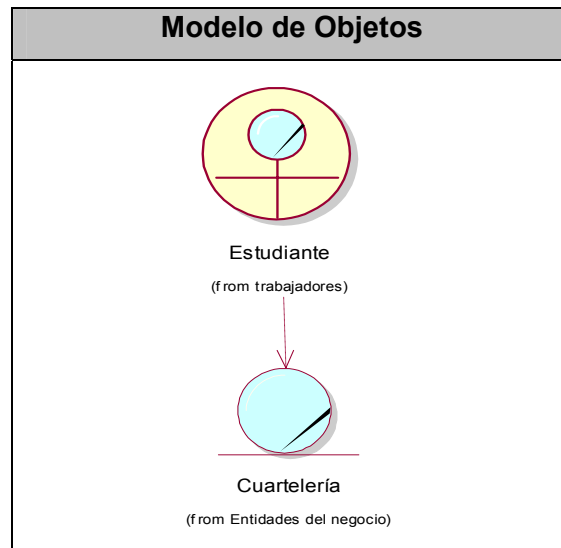
Nombre del Caso de Uso:	Gestionar Cuartelería	
Actores:	Vice Decano de Extensión (inicia)	
Propósito:	Permitir al Vice Decano de Extensión que se lleve a cabo la realización de la cuartelería.	
Resumen:	El Caso de Uso se inicia cuando el Vice Decano de Extensión gestiona la cuartelería y siguiendo una serie de procesos se da cumplimiento a la misma finalizando así el caso de uso.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del proceso de negocio	
1. El Vice Decano de Extensión gestiona la cuartelería.		
	2. El estudiante ve que día le corresponde la cuartelería.	
	3. El estudiante realiza la cuartelería.	
	4. La instructora verifica si el estudiante realizó la cuartelería.	
	5. La instructora da una evaluación y la entrega.	
6. El Vice Decano de Extensión archiva la evaluación.		
Curso alternativo de los eventos		
Acción 5	Si no realizó la cuartelería la instructora lo reporta y es llevado a comisión disciplinaria.	
Prioridad	Crítico	
Mejoras		

Otros	
-------	--

Diagrama de actividades



Modelo de objetos

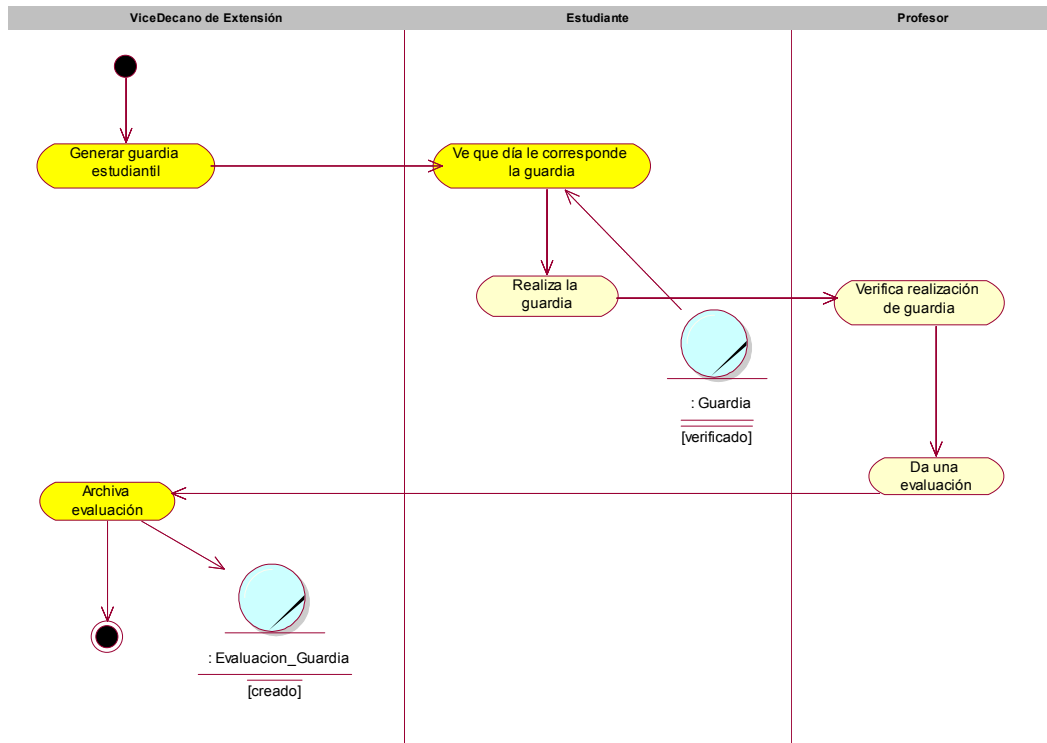


**Caso de Uso del Negocio: Gestionar Guardia Estudiantil**
 Descripción textual

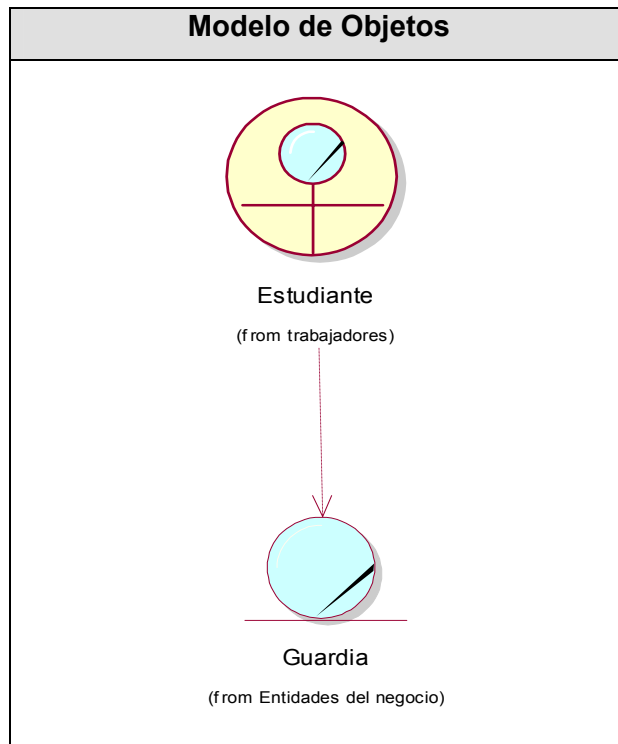
Nombre del Caso de Uso:	Gestionar Guardia Estudiantil	
Actores:	Vice Decano de Extensión (inicia)	
Propósito:	Permitir al ViceDecano de Extensión que se lleve a cabo la realización de la guardia estudiantil.	
Resumen:	El Caso de Uso se inicia cuando el Vice Decano de Extensión gestiona la guardia estudiantil y siguiendo una serie de procesos se da cumplimiento a la misma finalizando así el caso de uso.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del proceso de negocio	
1. El Vice Decano de Extensión gestiona la guardia estudiantil.		
	2. El estudiante ve que día le corresponde la guardia estudiantil.	
	3. El estudiante realiza la guardia estudiantil.	
	4. El profesor verifica la realización de la guardia estudiantil.	
	5. El profesor si realizó la guardia estudiantil da una evaluación y la entrega.	
6. El Vice Decano de Extensión archiva la evaluación.		
Curso alternativo de los eventos		
Acción 5	Si no realizó la guardia estudiantil el profesor lo reporta y es llevado a comisión disciplinaria.	
Prioridad	Crítico	
Mejoras		
Otros		



Diagrama de actividades



Modelo de objetos




**Anexo # 2**

Descripción textual de los Casos de Uso críticos del Sistema.

**Paquete de Seguridad**

 **Descripción del caso de uso Autenticar usuarios:**

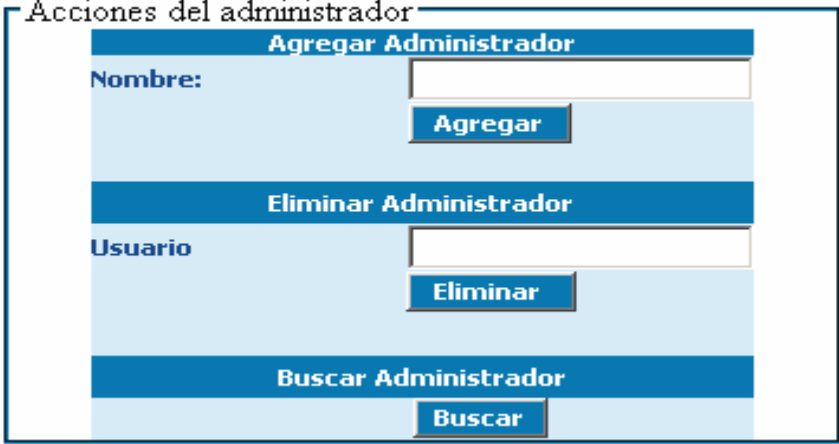
Nombre del Caso de Uso:	Autenticar usuarios	
Actores:	Usuario del Sistema (inicia)	
Propósito:	Controlar el acceso de los usuarios al sistema.	
Resumen:	El caso de uso se inicia cuando el usuario intenta introducir su usuario y contraseña para verificar si tiene o no acceso al sistema, si tiene acceso el usuario pasará inmediatamente a utilizar el sistema después de verificar el nivel de seguridad de acceso al sistema que posee el usuario, sino se le muestra un mensaje de error diciéndole usuario o contraseña incorrectos. Después el usuario puede nuevamente ingresar su usuario y contraseña. Una vez que el usuario acceda al sistema y se le realice el chequeo correspondiente termina el caso de uso.	
Referencias:	R1	
Precondiciones:		
Poscondiciones:	Se habilitan las funcionalidades según los privilegios.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El usuario intenta entrar al sistema.	2. El sistema pide que se autentique.	
3. El usuario introduce los datos solicitados.		
4. El usuario presiona el botón Aceptar.	5. El sistema verifica que los datos son correctos.	
	6. El usuario tiene permiso para entrar en el sistema	
Curso Alterno		
Acciones del Actor	Respuesta del Sistema	

Acción 5	El sistema verifica que los datos no son correctos.
Acción 6.	El sistema no puede darle al usuario permiso para entrar en el sistema y muestra un mensaje de error.
Prioridad:	Crítico
	

#### ✚ Descripción del caso de uso Gestionar usuarios:

Nombre del Caso de Uso:	Gestionar usuarios
Actores:	Administrador (inicia)
Propósito:	El administrador del sistema agrega o elimina un usuario del sistema.
Resumen:	El caso de uso se inicia cuando alguna persona autorizada decide agregar o eliminar algún usuario del sistema. Para agregar un usuario se introduce el usuario y privilegios, para eliminar se introduce el usuario que se desea eliminar, en caso de no poderse realizar este proceso se enviará un mensaje al usuario avisando que no se pudo realizar la operación terminado así este caso de uso.
Referencias:	R 1.2, R 1.2.1, R 1.2.2
Precondiciones:	Tener permiso de administración.
Poscondiciones:	Fueron modificados los datos de la tabla usuario. Fue creada o eliminada una instancia usuario.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema

1. El administrador del sistema selecciona la opción de gestión de usuario.	2. El sistema le brinda las opciones referentes a los usuarios: a) Insertar usuario. b) Eliminar usuario.
3. El administrador del sistema selecciona la opción insertar un nuevo usuario.	4. El sistema muestra un formulario solicitando la información necesaria para insertar un nuevo usuario: a) Nombre del usuario. b) Usuario. c) Privilegio
Sección insertar usuario:	
5. El administrador del sistema introduce los datos solicitados por el sistema.	
6. El administrador del sistema presiona el botón Adicionar.	7. El sistema verifica que todos los datos estén correctos.
	8. El sistema almacena la información en la base de datos. Concluyendo así el caso de uso.
Curso Alterno	
Acciones del Actor	Respuesta del Sistema
Acción 7.	El sistema verifica que los datos no son correctos, alguno no cumple con el formato adecuado. Se indica donde está el error para que sea corregido y se retorna a la Acción 4.
Acción 8.	El sistema no pudo almacenar los datos del usuario. Se muestra una notificación indicando el error. Retornar a Acción 4.
Sección eliminar usuario	

3. El administrador del sistema selecciona la opción eliminar usuario.	4. El sistema muestra el listado de los usuarios existentes en la base de datos.
5. El administrador del sistema localiza el usuario que se desea eliminar y presiona el botón Eliminar.	6. El sistema pudo eliminar correctamente al usuario de la base de datos.
Curso Alterno	
Acciones del Actor	Respuesta del Sistema
Acción 6	El sistema no pudo eliminar al usuario solicitado porque ha ocurrido algún error. Se muestra un mensaje de error. Se retorna a la Acción 4.
Prioridad:	Crítico
 <p>Acciones del administrador</p> <p><b>Agregar Administrador</b></p> <p>Nombre: <input type="text"/></p> <p><b>Agregar</b></p> <p><b>Eliminar Administrador</b></p> <p>Usuario <input type="text"/></p> <p><b>Eliminar</b></p> <p><b>Buscar Administrador</b></p> <p><b>Buscar</b></p>	

### Paquete de Entrada de Datos

#### 🚩 Descripción del caso de uso Generar Cuartelería:

Nombre del Caso de Uso:	Generar Cuartelería
Actores:	Administrador (inicia)
Propósito:	Brindar al administrador la posibilidad de generar una cuartelería habiendo seleccionado antes una fecha de inicio, una fecha fin y un edificio.
Resumen:	El caso de uso se inicia cuando el administrador necesita generar la cuartelería de un período

	determinado, el administrador selecciona una fecha de inicio, una fecha fin y el número del edificio del que desea generar la cuarterería. El sistema le muestra la cuarterería para el período de tiempo indicado. Si el administrador desea realizar algún cambio de estudiante selecciona la opción modificar, realiza los cambios y estos son guardados en la BD.
Referencias:	R 2, R 2.1,R 2.2
Precondiciones:	El administrador ha sido validado.
Poscondiciones:	Se actualiza en la base de datos la tabla que guarda la cuarterería.

## Curso Normal de los Eventos

Acciones del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción Generar Cuarterería.	2 El sistema muestra un formulario: a) fecha inicio b) fecha fin. c) edificio
3. El Administrador selecciona la fecha inicio, fecha fin y el edificio.	4. El sistema muestra la cuarterería del período de tiempo que generó.
5. El Administrador imprime la cuarterería mostrada.	
Prioridad:	Crítico

Generar Cuarterería

Fecha Inicio	Fecha Fin	Edificio
<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text"/>	<input style="width: 90%;" type="text" value="[-]"/> ▼
<input style="background-color: #0070C0; color: white; padding: 5px 10px;" type="button" value="Generar"/>		

< Jun 2007 >

Do	Lu	Ma	Mi	Ju	Vi	Sa
22	27	28	29	30	31	1
23	3	4	5	6	7	8
24	10	11	12	13	14	15
25	17	18	19	20	21	22
26	24	25	26	27	28	29
27	1	2	3	4	5	6

**✚ Descripción del caso de uso Generar Guardia Estudiantil:**

Nombre del Caso de Uso:	Generar Guardia Estudiantil	
Actores:	Administrador (inicia)	
Propósito:	Brindar al administrador la posibilidad de generar la guardia estudiantil habiendo seleccionado antes una fecha de inicio y una fecha fin.	
Resumen:	El caso de uso se inicia cuando el administrador necesita generar la guardia estudiantil de un período determinado, el administrador selecciona una fecha de inicio y una fecha fin y el sistema le muestra la guardia estudiantil para el período de tiempo indicado.	
Referencias:	R 3, R 3.1	
Precondiciones:	El administrador ha sido validado.	
Poscondiciones:	Se actualiza la base de datos del listado de la guardia estudiantil en el sistema.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El Administrador selecciona la opción Generar Guardia Estudiantil.	2. El sistema muestra un formulario: a) fecha inicio b) fecha fin.	
3. El Administrador selecciona la fecha inicio y la fecha fin para la cual desea generar la guardia.	4. El sistema guarda la guardia estudiantil generada en la BD y luego la muestra.	
5. El Administrador imprime la c mostrada.		
Prioridad:	Crítico	

Generar Guardia

Fecha Inicio	Fecha Fin	Generar
<input type="text"/>	<input type="text"/>	<input type="button" value="Generar"/>

< Jun 2007 >

	Do	Lu	Ma	Mi	Ju	Vi	Sa
22	27	28	29	30	31	1	2
23	3	4	5	6	7	8	9
24	10	11	12	13	14	15	16
25	17	18	19	20	21	22	23
26	24	25	26	27	28	29	30
27	1	2	3	4	5	6	7

Limpia Cierre

### Paquete de Reportes

#### 🚩 Descripción del caso de uso Ver la cuartería:

Nombre del Caso de Uso:	Ver la cuartería.	
Actores:	Usuario. (Inicia)	
Propósito:	Permite al usuario ver que día le corresponde la cuartería, además su evaluación si ya la realizó.	
Resumen:	El caso de uso se inicia cuando el usuario desea ver que día le corresponde la cuartería.	
Referencia:	R 4.1	
Precondiciones:	El usuario ha sido validado.	
Poscondiciones:	Se utiliza la base de datos que contiene la cuartería.	
Curso Normal de los Eventos		
Acciones del Actor	Respuesta del Sistema	
1. El usuario selecciona la opción ver la cuartería.	2. El sistema muestra un formulario: a) nombre b) apellidos c) edificio	
3. El usuario selecciona uno o todos estos campos.	4. El sistema muestra toda la información referente a la cuartería: a) fecha	



	b)nombre y apellidos del estudiante c)edificio y apto												
Prioridad:	Crítico												
<p>Cuartelería del Estudiante</p> <table border="1"> <thead> <tr> <th>Fecha Cuartelería</th> <th>Nombre</th> <th>Apellidos</th> <th>Apartamento</th> </tr> </thead> <tbody> <tr> <td>2007-06-04</td> <td>Tomas</td> <td>Rodriguez Barrios</td> <td>11-1-01</td> </tr> <tr> <td>2007-06-13</td> <td>Tomas</td> <td>Rodriguez Barrios</td> <td>11-1-01</td> </tr> </tbody> </table> <p style="text-align: center;">1</p>		Fecha Cuartelería	Nombre	Apellidos	Apartamento	2007-06-04	Tomas	Rodriguez Barrios	11-1-01	2007-06-13	Tomas	Rodriguez Barrios	11-1-01
Fecha Cuartelería	Nombre	Apellidos	Apartamento										
2007-06-04	Tomas	Rodriguez Barrios	11-1-01										
2007-06-13	Tomas	Rodriguez Barrios	11-1-01										

#### 🚩 Descripción del caso de uso Ver la Guardia Estudiantil:

Nombre del Caso de Uso:	Ver la Guardia Estudiantil
Actores:	Usuario. (Inicia)
Propósito:	Permite al usuario ver que día le corresponde la Guardia Estudiantil.
Resumen:	El caso de uso se inicia cuando el usuario desea ver que día le corresponde la guardia estudiantil.
Referencia:	R 4.2
Precondiciones:	El usuario ha sido validado.
Poscondiciones:	Se utiliza la base de datos que contiene la guardia.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona la opción ver la guardia estudiantil.	2. El sistema muestra un formulario: a)nombre b)apellidos c)grupo
3. El usuario selecciona uno o todos estos campos.	4. El sistema muestra toda la información referente a la guardia estudiantil: a) fecha

	b) nombre y apellidos del estudiante c) grupo y subgrupo d) turno y posta														
Prioridad:	Crítico														
<p>Guardia del Estudiante</p> <table border="1"> <thead> <tr> <th>Fecha</th> <th>Nombre</th> <th>Apellidos</th> <th>Grupo</th> <th>Subgrupo</th> <th>Turno</th> <th>Posta</th> </tr> </thead> <tbody> <tr> <td>2007-06-02</td> <td>Yan</td> <td>Ayata Escalona</td> <td>2101</td> <td>Primer subgrupo</td> <td>11:00pm-1:30am</td> <td>detras del 1 y el 2</td> </tr> </tbody> </table>		Fecha	Nombre	Apellidos	Grupo	Subgrupo	Turno	Posta	2007-06-02	Yan	Ayata Escalona	2101	Primer subgrupo	11:00pm-1:30am	detras del 1 y el 2
Fecha	Nombre	Apellidos	Grupo	Subgrupo	Turno	Posta									
2007-06-02	Yan	Ayata Escalona	2101	Primer subgrupo	11:00pm-1:30am	detras del 1 y el 2									

#### 🚩 Descripción del caso de uso Buscar información:

Nombre del Caso de Uso:	Buscar información.
Actores:	Usuario. (Inicia)
Propósito:	Permite al usuario buscar información referente a los estudiantes. (Directorio)
Resumen:	El caso de uso se inicia cuando el usuario desea obtener información referente a los estudiantes, para esto puede hacerlo mediante una búsqueda simple o una búsqueda avanzada.
Referencia:	R 4.3.1, R 4.3.2
Precondiciones:	El usuario ha sido validado.
Poscondiciones:	Se utiliza la base de datos para obtener los datos de los estudiantes.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona la opción Buscar.	2. El sistema muestra la opción de realizar una búsqueda simple o una búsqueda avanzada.
3. El usuario selecciona la búsqueda simple.	4. El sistema muestra la opción de buscar por un campo (nombre).

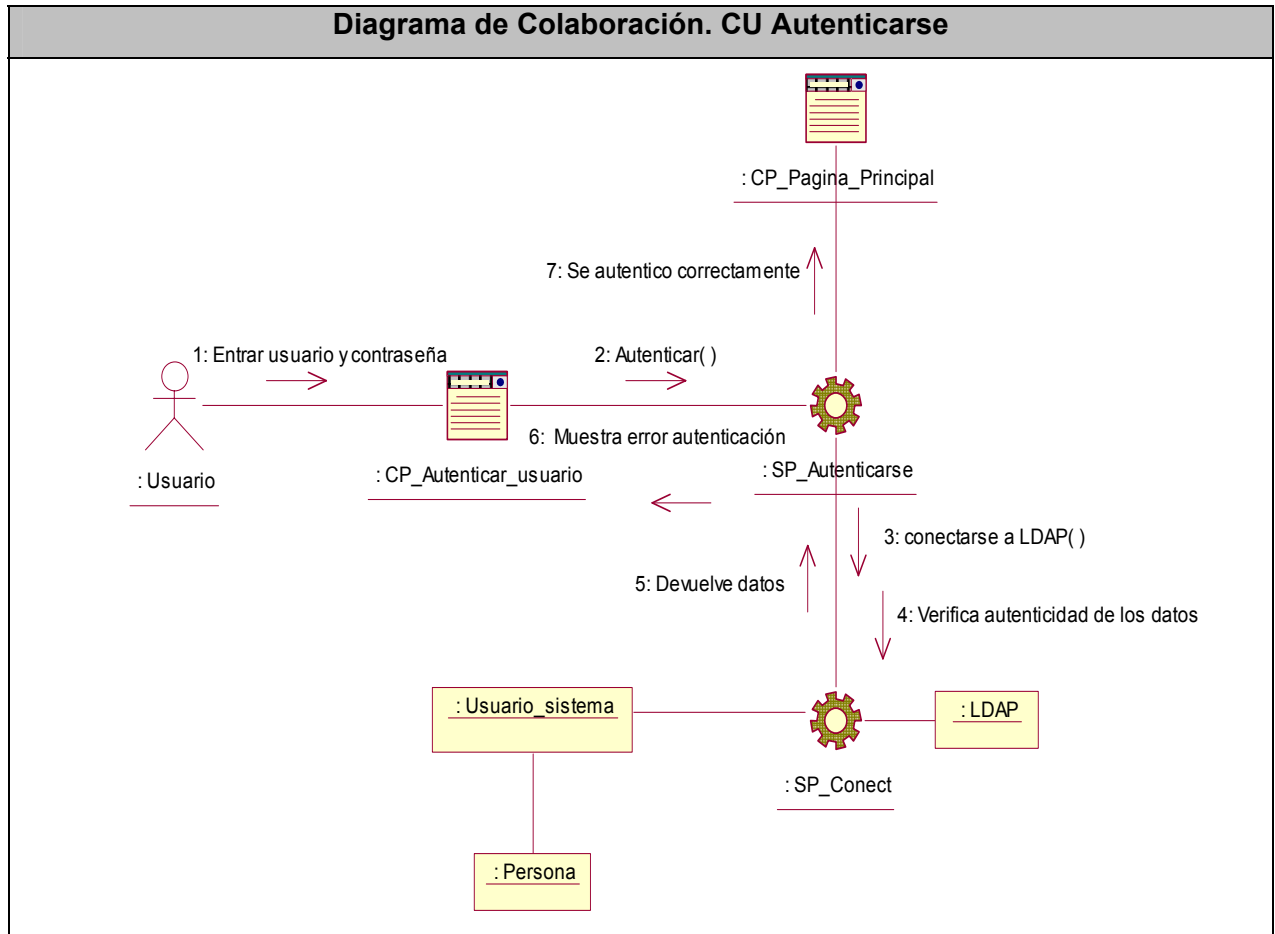
5. El usuario entra el criterio de búsqueda.	6. El sistema muestra toda la información referente al o los estudiantes que coinciden con el criterio de búsqueda entrado. a) nombre y apellido b) provincia y municipio c) sexo d) grupo e) usuario y correo
Curso Alterno	
Acciones del Actor	Respuesta del Sistema
Acción 3 El usuario selecciona la búsqueda avanzada.	7. El sistema muestra la opción de buscar por varios criterios de búsqueda (nombre, apellidos, provincia, municipios, edificio).Retornar a 5
Prioridad:	Crítico

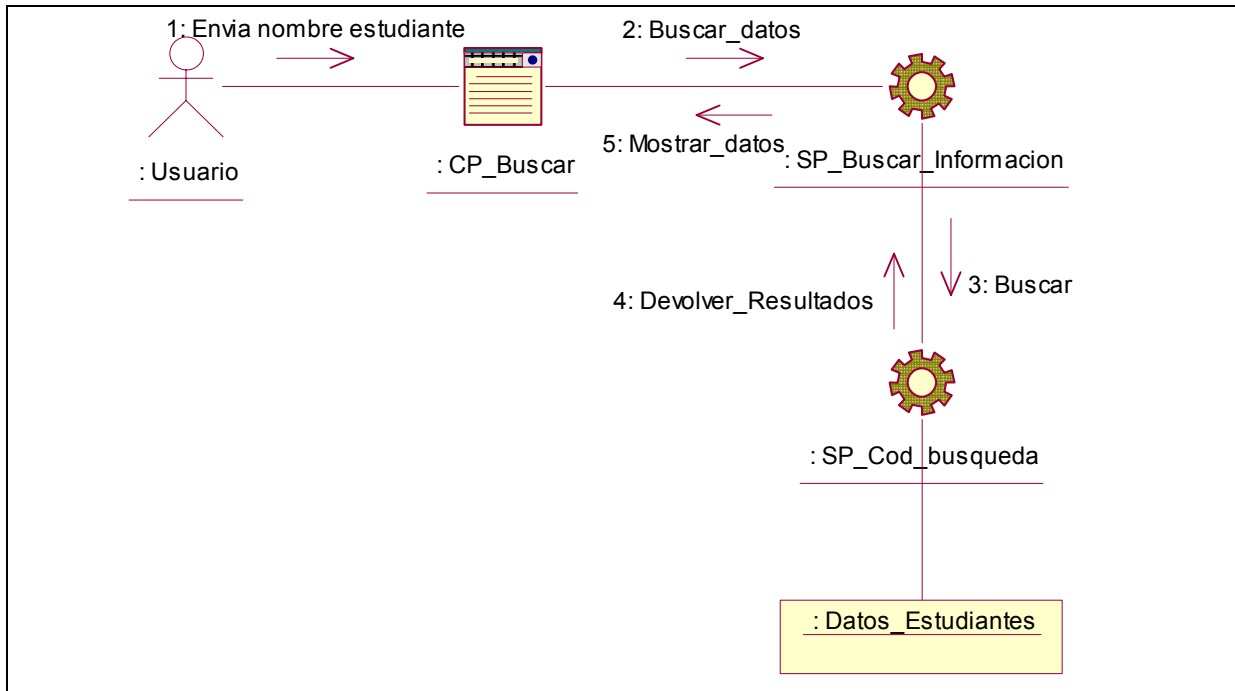
Búsqueda Avanzada

Búsqueda Avanzada	
Nombre:	<input type="text"/>
Provincia:	<input type="text" value="[-Seleccione-]"/>
Sexo:	<input type="text" value="[--Seleccione--]"/>
Edificio:	<input type="text" value="[-]"/>
Apellidos:	<input type="text"/>
Municipio:	<input type="text" value="Selecciona opción..."/>
Año:	<input type="text" value="[--Seleccione--]"/>
Usuario:	<input type="text"/>
<input type="button" value="Buscar"/> <input type="button" value="Limpiar"/>	

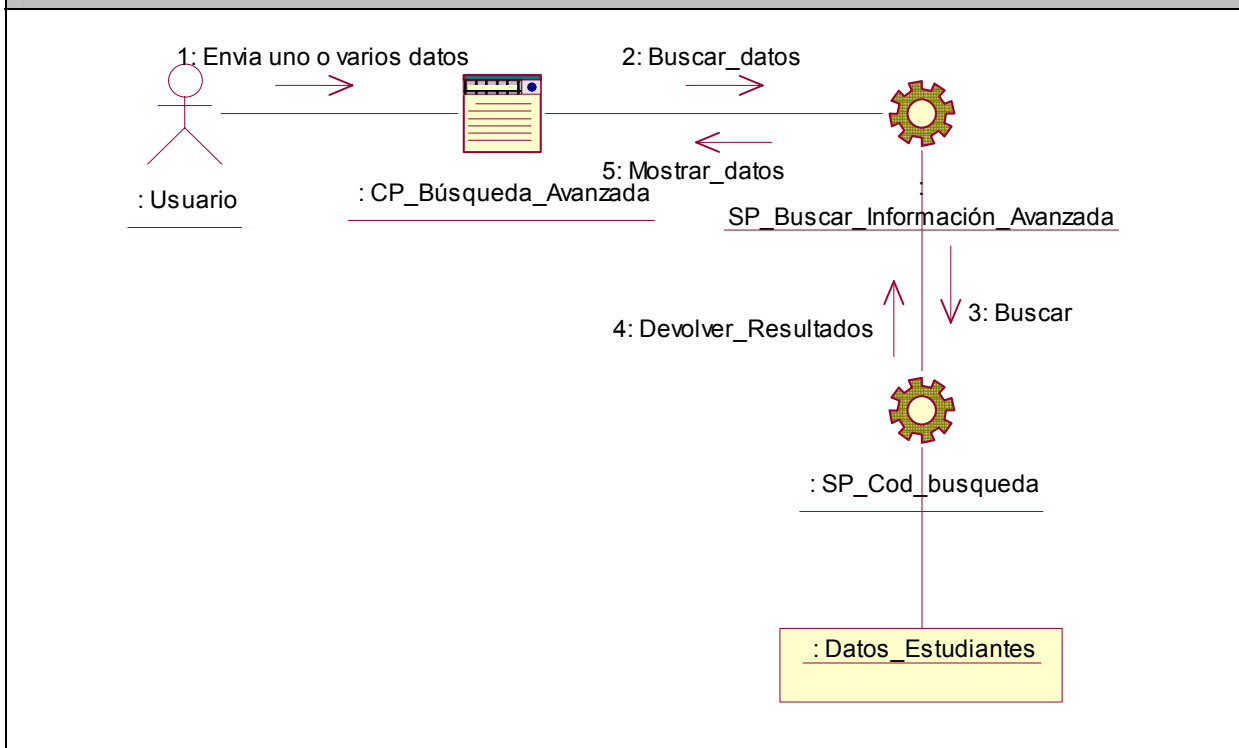
**Anexo # 3**

Diagramas de Interacción.

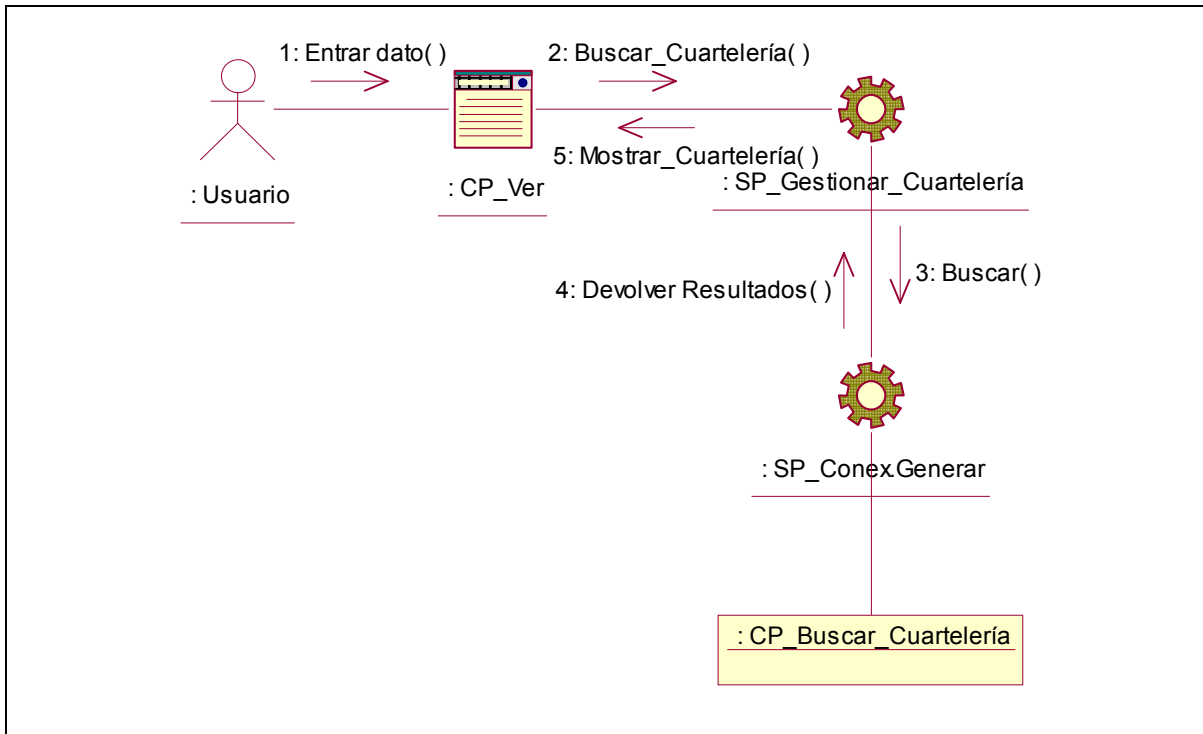
**Diagrama de Colaboración. CU Buscar Información Simple.**



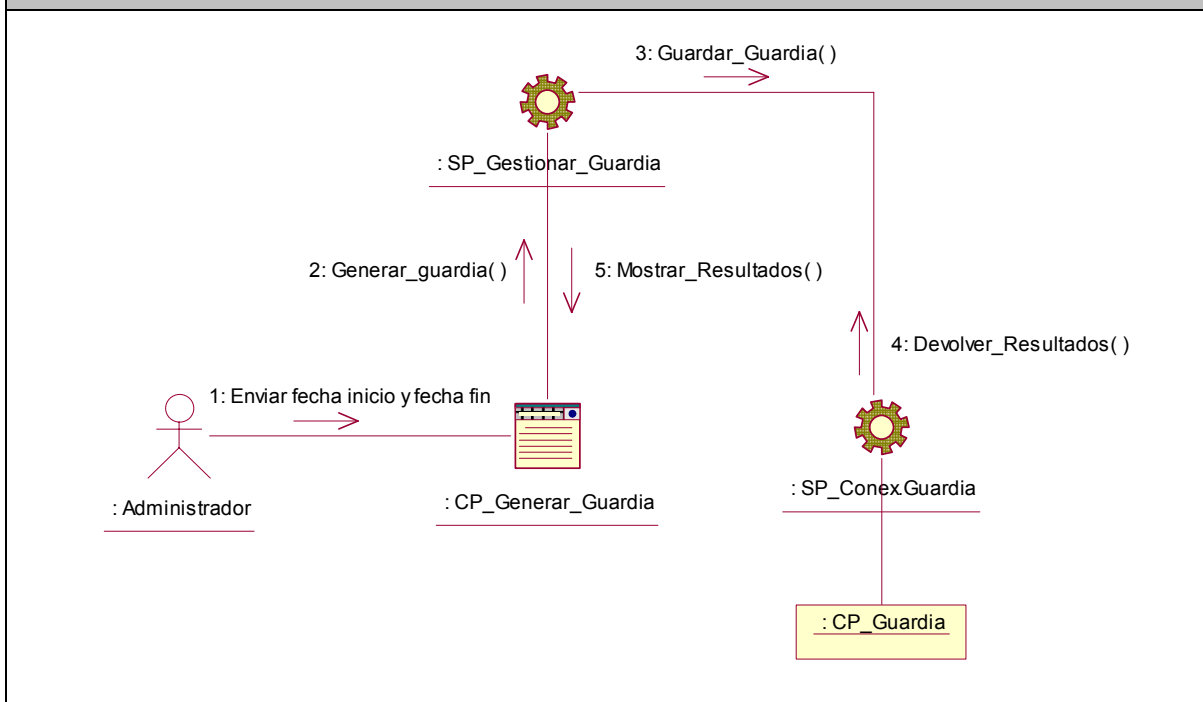
### Diagrama de Colaboración. CU Buscar Información Avanzada.



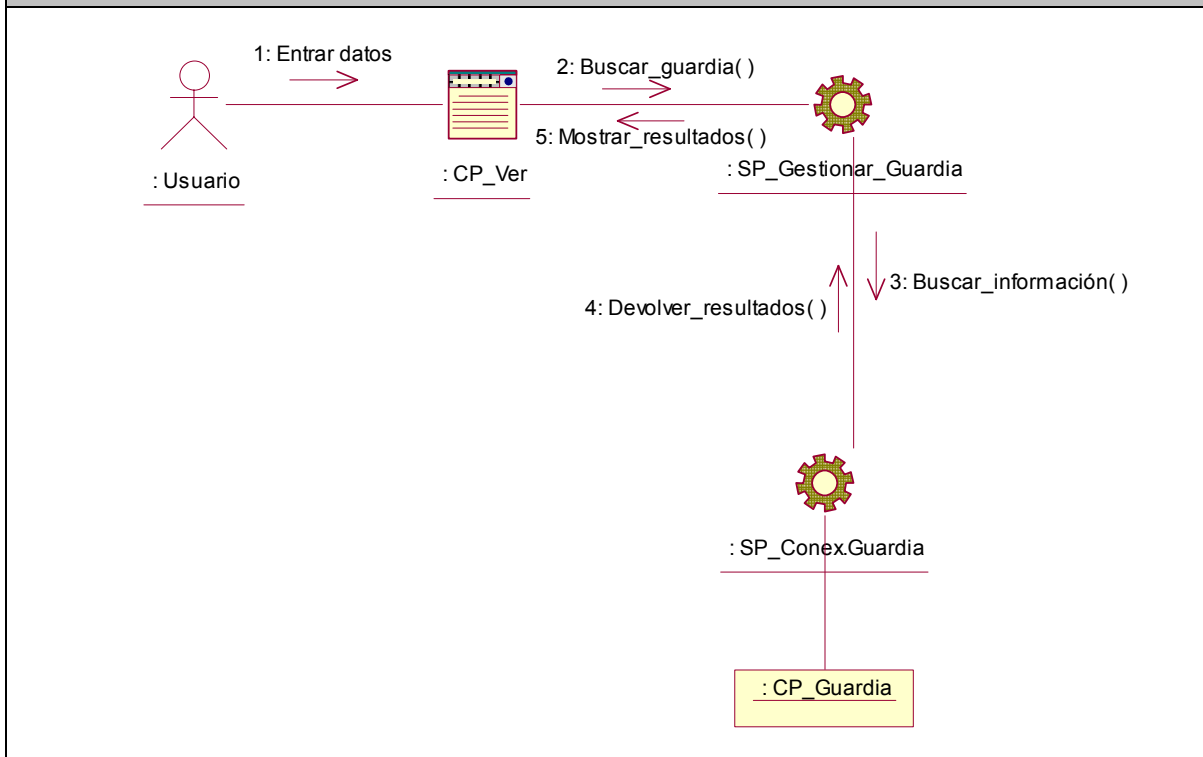
### Diagrama de Colaboración. CU Ver Cuartelería



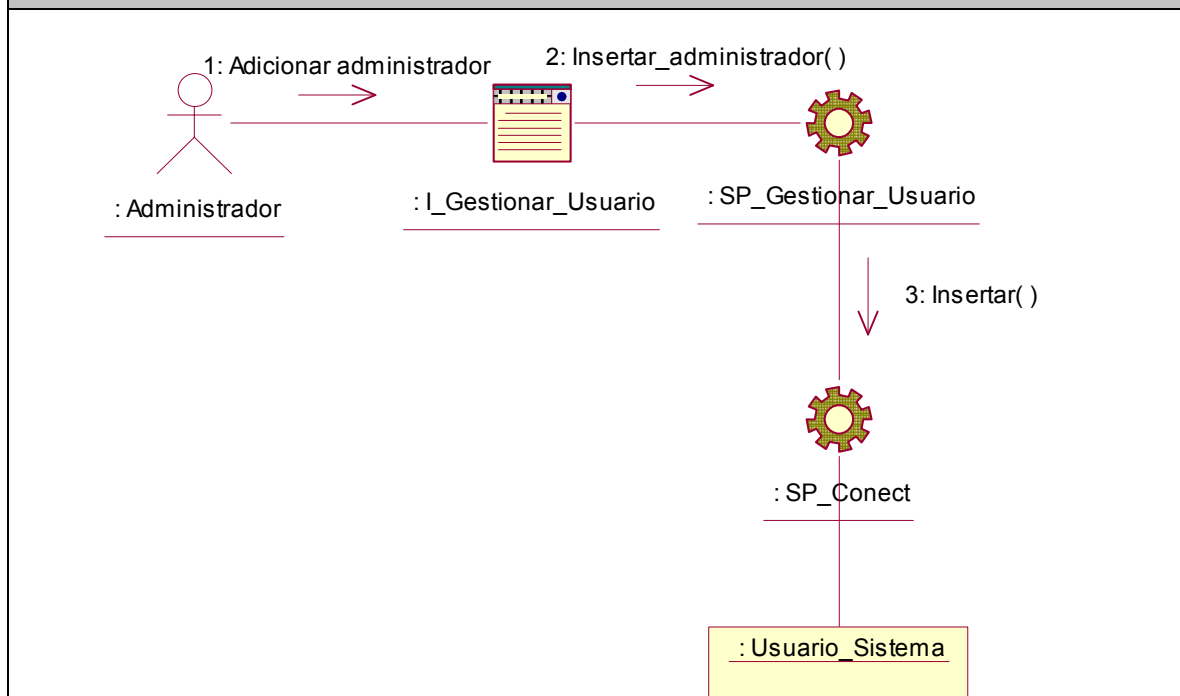
### Diagrama de Colaboración. CU Generar Guardia.



### Diagrama de Colaboración. CU Ver Guardia.



### Diagrama de Clases de Diseño. CU Gestionar Usuario(Insertar Administrador)



### Diagrama de Clases de Diseño. CU Gestionar Usuario(Eliminar Administrador)

