

**Universidad de las Ciencias Informáticas**

**Facultad 5**



**Título: Sistema de Gestión para la  
Planificación Docente en la Facultad 5.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**Autoras:** Dailé Alvarez Hernández  
Elizeth Martínez Vazquez

**Tutora:** Ing. Dania Souchay Fábrega.

**Ciudad de La Habana**

**Junio 2009**

*El esqueleto de la ciencia son los hechos, pero los músculos y los nervios son el significado que se les confiere y el alma son las ideas.*

***Ruy Pérez Tamayo***

## DATOS DE CONTACTO

### Autores

**Nombre:** Elizeth Martínez Vazquez

**Correo Electrónico:** [emvazquez@estudiantes.uci.cu](mailto:emvazquez@estudiantes.uci.cu)

**Nombre:** Dailé Álvarez Hernández

**Correo Electrónico:** [dalvarezh@estudiantes.uci.cu](mailto:dalvarezh@estudiantes.uci.cu)

### Tutor

**Nombre:** Dania Souchay Fabrega

**Institución:** Universidad de las Ciencias Informáticas (UCI)

**Título:** Ingeniería Informática.

**Categoría Docente:** Profesor Asistente.

**Correo Electrónico:** [dsouchay@uci.cu](mailto:dsouchay@uci.cu)

## DEDICATORIA

*A mis padres por ser la luz de mis ojos.  
Elizeth*

*A mi abuelita MIMA, porque se que estaría orgullosa de mí si estuviera aquí.  
Dailé*

## *Agradecimientos Generales*

*A Roberkys por su apoyo incondicional, porque sin ti no hubiera sido posible este momento.*

*A Yoisy por habernos dedicado parte de su tiempo, por su ayuda sin limite y por ser una tutora para nosotros.*

*A Lida por ser la autora intelectual de este tema.*

*A Dania nuestra tutora por aceptar guiarnos por este largo camino.*

*A Laritza que junto con Roberkys hicieron este sueño posible.*

*A todos los amigos que nos apoyaron y nos brindaron su ayuda a: Leandro, Aleida, Yurian, Gera, Yausell, Yaself, la china y Jamit.*

## *Agradecimientos Dailé*

*A mi mamá por ser lo más grande que tengo en este mundo. Por su sacrificio en todos estos años de estudio. Por estar siempre a mi lado y porque sin ti nada de esto fuera posible.*

*A mi hermano Andry por todos los momentos que pasamos juntos algunos buenos y otros no tanto pero siempre juntos.*

*A mi papá por su cariño inmenso, por hacerme reír con tu cosas y porque se que siempre me tienes presente.*

*A mis tías y a mis primos por estar siempre pendientes a mis necesidades, por su apoyo incondicional, por confiar en mí y darme ánimo para seguir adelante.*

*A la muñeca porque aunque ella no quiere que la ponga, tengo mucho que agradecerle, porque apareciste cuando pensé que todo estaba perdido y aun sin conocerme me ayudaste como nadie, porque tu amistad es una de las cosas mas lindas que siempre voy a recordar, por todos los momentos que pasamos juntas que fueron muchos y casi todos muy buenos, por malcriarme y por quererme tanto.*

*A Yoisyta por ese cariño tan grande que me tienes, por siempre estar ahí cada ves que te necesité, por cada uno de tus regaños, tus peleas porque me daban la medida de cuanto me quieres y te preocupas por mi, y por sobre todas las cosas gracias por ser tan especial conmigo.*

*A Yaijet por ser mi primera amiga en esta Universidad por el apoyo que me diste siempre, porque a pesar de que nos separamos seguiste siendo como una hermana para mí.*

*A Susej y a Yurian por cada uno de los detalles que tuvieron conmigo, por quererme y apoyarme siempre, por ser grandes amigas para mí, Gracias a ti Susej por cada una de las llamadas que me hiciste en los momentos que mas las necesite.*

*A Juan Manuel por brindarme su ayuda en todo momento, por ser tan bueno conmigo.*

*A los amigos que ya no están Yamilkita y el Luisi porque no existe el olvido entre nosotros.*

*A mis compañeros del grupo por los grandes momentos que pasamos juntos especialmente a la Peluita (Lisandra), Yaira, Yaimy, Mirtica, Pablito.*

*Al piquete (Aleida, Gera, Maco, Pepe, Luisi, Yeily ) por los momentos de fiesta .*

*A los profesores que me ofrecieron su ayuda cuando lo necesitaba.*

*A mi compañera de tesis Elizeth por el apoyo durante todo este tiempo.*

# *AGRADECIMIENTOS DE ELIZETH*

*Por su amor incondicional y ser lo más importante en mi vida, a mi mamá y mi papá.*

*Por su inmensa ternura y comprensión, a mi abuelita Digna.*

*Por aguantar mis malacrianzas, a mis hermanos Eliza y Leonardo.*

*Por sus sabios consejos, a mi tío Marcelino y mi tía Gume.*

*Por la simplicidad de su cariño y su inocencia, a todos mis sobrinos.*

*Por su apoyo y preocupación, a toda mi familia.*

*Por haber estado a mi lado estos años de universidad, a mis amigas: Yadi, Lisney, Yadira,*

*Yaimy, Lisandra y Mirt ha.*

*Por pasar junto conmigo este arduo andar, a mi compañera de tesis PAPIŦA.*

*Por estar siempre conmigo y aceptarme como soy, a los amigos de antes, los de ahora y los de siempre.*

## RESUMEN

El impulso de los sistemas de gestión en estos tiempos promete estrategias encaminadas a lograr la optimización de los procesos de negocio. Permitir un ascenso perdurable en el desarrollo de la gestión de políticas, procedimientos y procesos es la principal meta de estos sistemas.

El presente trabajo se basa en el desarrollo de un Sistema de Gestión para la Planificación Docente (SGPD) en la Facultad 5 de la Universidad de las Ciencias Informáticas, que permita la gestión de procesos que tributan a un adecuado desempeño y funcionamiento del proceso docente educativo, como lo son la guardia docente y el cuidado de exámenes. El proceso de desarrollo del sistema estuvo guiado por la metodología tradicional RUP (Proceso Unificado de Rational), propiciando un conjunto de productos intermedios en cada una de las fases de trabajo como lo son la especificación de requisitos y el modelo de diseño correspondiente a la aplicación resultante. Para el desarrollo de dicho sistema se empleó a PHP como lenguaje de programación, Apache como servidor Web y MySQL como sistema gestor de bases de datos. Además se utilizó al CMS Drupal como ambiente de desarrollo, incluyéndole algunos módulos para soportar las funcionalidades necesarias.

Como resultado final se propone un sistema que permite la planificación eficiente del cuidado de exámenes y la guardia docente logrando una mayor rapidez y organización en su ejecución. Tal eficiencia pudo ser comprobada durante la etapa de pruebas a la que fue sometido el sistema, arrojando resultados favorables a su rendimiento y aceptación del cliente/usuario final.

**Palabras Claves:** Sistema de Gestión, Guardia Docente, Cuidado de Examen, SGPD.

# TABLA DE CONTENIDOS

|  |    |
|--|----|
| INTRODUCCIÓN .....   | 1  |
| CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....                                    | 6  |
| INTRODUCCIÓN .....   | 6  |
| 1.1 ¿Qué es la gestión?.....   | 6  |
| 1.2 La Gestión de la Información.....                                      | 6  |
| 1.3 Sistemas de Información.....   | 7  |
| 1.4 La Planificación .....   | 9  |
| 1.5 Sistema de Gestión.....  | 10 |
| 1.6 Análisis de otras soluciones existentes.....                           | 11 |
| 1.7 Sistemas Gestores de Base de Datos (SBGD) .....                        | 12 |
| 1.7.1 PostgreSQL.....  | 13 |
| 1.7.2 MySQL.....   | 14 |
| 1.7.3 Oracle .....   | 16 |
| 1.7.4 Microsoft SQL Server.....  | 17 |
| 1.8 Metodologías para el desarrollo del Software .....                     | 17 |
| 1.8.1 RUP .....  | 18 |
| 1.9 Lenguajes de Modelados.....  | 19 |
| 1.9.1 UML.....   | 20 |
| 1.10 Patrones .....  | 20 |
| 1.10.1 Patrones de casos de uso .....                                      | 21 |
| 1.10.2 Patrones de diseño.....   | 22 |
| 1.10.3 Patrones arquitectónicos .....                                      | 24 |
| 1.11 Lenguajes de Programación y Tecnologías .....                         | 25 |
| 1.11.1 Lenguajes de Programación y Tecnologías del lado del Cliente .....  | 25 |
| 1.11.2 Lenguajes de Programación y tecnologías del lado del Servidor ..... | 27 |
| 1.12 IDEs de desarrollo.....   | 29 |
| 1.12.1 Dreamweaver .....   | 29 |
| 1.12.2 Zend Studio.....  | 29 |
| 1.13 Servidores Web .....  | 30 |
| 1.13.1 Internet Information Servers (IIS).....                             | 30 |
| 1.13.2 Apache .....  | 30 |
| 1.14 Sistema de Gestión de Contenidos (CMS).....                           | 31 |
| 1.14.1 ¿Qué es un CMS?.....  | 31 |
| 1.15 Propuesta de tendencias y tecnologías a utilizar:.....                | 35 |
| CONCLUSIONES DEL CAPÍTULO.....   | 36 |
| CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA.....                     | 37 |
| Introducción .....   | 37 |
| 2.1 Modelo del negocio actual.....   | 37 |
| 2.1.1 Planificación de la Guardia Docente.....                             | 37 |

|   |    |
|---|----|
| 2.1.2 Planificación del Cuidado de Exámenes .....             | 38 |
| 2.2 Actores del Negocio .....                                 | 38 |
| 2.3 Trabajadores del negocio .....                            | 38 |
| 2.4 Modelo de casos de uso del negocio .....                  | 39 |
| 2.5 Casos de uso del negocio .....                            | 40 |
| 2.5.1 Descripción de los casos de uso del negocio. ....       | 40 |
| 2.6 Definición de los requisitos funcionales. ....            | 43 |
| 2.7 Definición de los requisitos no funcionales .....         | 44 |
| 2.8 Actores del Sistema.....                                  | 45 |
| 2.8.1 Descripción de los actores del sistema.....             | 46 |
| 2.9 Diagrama de Casos de Uso del sistema .....                | 47 |
| 2.10 Casos de uso del sistema a tratar: .....                 | 47 |
| 2.10.1 Descripción de los Casos de Uso del Sistema. ....      | 48 |
| CONCLUSIONES DEL CAPÍTULO.....                                | 52 |
| <br>  |    |
| CAPÍTULO 3: CONTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....        | 53 |
| <br>  |    |
| Introducción .....  | 53 |
| 3.1 Análisis.....   | 53 |
| 3.1.1 Diagrama de clases de análisis.....                     | 53 |
| 3.1.2 Diagramas de interacción.....                           | 54 |
| 3.2 Diseño .....  | 54 |
| 3.2.1 Descripción de los paquetes de Drupal. ....             | 56 |
| 3.2.2 Diagrama de clases persistentes.....                    | 58 |
| 3.2.3 Modelo de Datos. ....                                   | 59 |
| 3.2.4 Descripción de las tablas de Modelo de Datos. ....      | 59 |
| 3.2.5 Tratamiento de Errores.....                             | 60 |
| 3.2.6 Seguridad. ....   | 60 |
| Conclusiones del capítulo .....                               | 61 |
| CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA .....                     | 62 |
| <br>  |    |
| Introducción .....  | 62 |
| 4.1 Implementación.....                                       | 62 |
| 4.1.2 Diagrama de Despliegue. ....                            | 62 |
| 4.1.3 Diagrama de Componentes. ....                           | 63 |
| 4.2 Modelo de Prueba. ....                                    | 65 |
| 4.2.1 Métodos de Prueba.....                                  | 66 |
| 4.3 Casos de pruebas de la aplicación.....                    | 66 |
| 4.3.1 Caso de Prueba “CU Autenticar Usuario”. ....            | 66 |
| 4.3.2 Casos de Prueba “CU Planificar Guardia Docente” .....   | 68 |
| 4.3.3 Caso de Prueba: “CU Planificar Cuidado de Examen” ..... | 69 |
| 4.3.4 Caso de Prueba: “CU Gestionar Grupos” .....             | 72 |
| 4.4 Resultados de la aplicación.....                          | 73 |
| Conclusiones del capítulo .....                               | 74 |
| CONCLUSIONES .....  | 75 |
| <br>  |    |
| RECOMENDACIONES .....   | 76 |
| <br>  |    |
| BIBLIOGRAFÍA .....  | 77 |

|                        |     |
|------------------------|-----|
| GLOSARIO .....         | 81  |
| ANEXOS .....           | 84  |
| ÍNDICE DE FIGURAS..... | 109 |
| ÍNDICE DE TABLAS ..... | 110 |

### Introducción

En este siglo, marcado por el acelerado ritmo del desarrollo tecnológico y la globalización del mercado a nivel mundial, los sistemas de gestión son cada vez más conscientes de la importancia del individuo en la consecución de sus metas, jugando un papel de primer orden por su aplicación a cualquier esfera de la vida social. Partiendo del hecho de que un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización [9], éste ayuda a lograr los objetivos de la organización mediante una serie de estrategias que incluyen la optimización de los procesos, el enfoque centrado en la gestión y el pensamiento disciplinado. La necesidad de enfrentar los desafíos del cambiante mercado global de hoy y el desarrollo potencial existente en las organizaciones han provocado que, tanto en los países más como en los menos desarrollados, se torne imprescindible el auge del empleo de los sistemas de gestión.

Cuba es uno de los países que se encuentra en vía de desarrollo, aún así se hace difícil pensar en la educación sin referenciar el avance de las Tecnologías de la Información y las Comunicaciones (TICs). En este sentido el país se encuentra inmerso en un proceso de crecimiento y extensión, provocando de esta manera, el desarrollo paulatino de sistemas de gestión de los procesos de negocio en las diferentes áreas de la nación. La universidad cubana ha sido un baluarte en la cooperación al desarrollo de la gestión de aplicaciones. Un ejemplo de ello lo constituye la Universidad de las Ciencias Informáticas (UCI) que ha venido contribuyendo de forma gradual a la automatización y optimización de los diferentes procesos en correspondencia con las problemáticas existentes.

En la UCI, específicamente en la Facultad 5, se llevan a cabo un conjunto de procesos que tributan al adecuado desempeño y funcionamiento del proceso docente educativo, como los relacionados con la planificación de las actividades docentes. El cuidado de exámenes y la planificación de la guardia docente son dos de los procesos que se ejecutan con más frecuencia y necesitan de gran eficiencia en su cumplimiento. Actualmente su ejecución se torna compleja; pues la planificación de locales, horarios, distribución de profesores con sus respectivas afectaciones como las preparaciones metodológicas y horarios de superación, además de toda la información necesaria para efectuar los procesos que se

manejan es muy diversa y gran parte de ella se registra de forma manual, provocando que la información no posea un orden consecuente de almacenamiento.

Durante el desarrollo de tales procesos se manejan herramientas como Microsoft Excel y Microsoft Word; sin embargo no se aprovechan las ventajas que pudiera aportar el empleo de un sistema de base de datos, permitiendo el procesamiento y recuperación de la información de forma automática. Seguidamente se resume un conjunto de inconvenientes o dificultades provocadas por la problemática explicada:

- ✓ La cantidad de información a gestionar aumenta aceleradamente y de manera dispersa, provocando dificultades en el acceso a los datos archivados como:
  - Planificación de exámenes.
  - Horario Docente.
  - Relación de profesores por asignatura, departamento y grupo.
  - Relación de profesores eximidos y/o asignados por el Dpto. para estas tareas.
- ✓ No se tiene un espacio o sitio publicado en red donde almacenar la información que pueda ser consultado sin fronteras de espacio en tiempo o lugar.
- ✓ Es un hecho el riesgo de deterioro o pérdida de los documentos físicos donde se almacena y registra información valiosa, trayendo consigo un engorroso proceso de re planificación y obtención en el tiempo requerido.

Teniendo en cuenta la problemática explicada se puede definir, mediante la siguiente interrogante, el **problema científico** a resolver en la presente investigación:

¿Cómo lograr un mejor desempeño en los procesos de planificación, cuidado de exámenes y guardia docente, en el área de formación de la Facultad 5 de la Universidad de las Ciencias Informáticas?

Como **hipótesis de investigación** se plantea que si se implementa un Sistema de Gestión de Planificación Docente se mejorará en el proceso de la planificación y consecuentemente en el control de las actividades docente-educativas de la Facultad 5.

Para dar solución al problema planteado se ha formulado como **objetivo general de la investigación** Implementar un Sistema para la Planificación Docente.

Por lo que se enmarca como **objeto de estudio** el Proceso de Control y Planificación Docente Educativo, especificándose como **campo de acción** el Proceso de planificación del cuidado de exámenes y la guardia docente en el área de Formación de la Facultad 5.

Para dar cumplimiento al objetivo general planteado se han formulado algunas **tareas de investigación** que facilitarán el desarrollo del mismo:

- ✓ Entrevistar a la planificadora docente de la Facultad 5.
- ✓ Realizar estudio del arte acerca de los sistemas de gestión y los procesos de planificación docente.
- ✓ Realizar un estudio de las herramientas y tecnologías a utilizar.
- ✓ Realizar el análisis y diseño del sistema a implementar.
- ✓ Realizar la implementación del sistema diseñado.
- ✓ Diseñar y ejecutar las pruebas de funcionalidad al Sistema de Planificación Docente de la Facultad 5.

A continuación se muestran los **métodos científicos** que se emplearán:

- ✓ **Métodos teóricos:**
  - Analítico-Sintético: A través de este método se definieron los principales elementos que caracterizan a los sistemas de gestión, así como las herramientas más utilizadas en el desarrollo de éstos, tomando como punto de partida los aspectos relacionados con la gestión de la información y la planificación.
  - Modelación: Esta se llevará a cabo mediante el análisis y diseño del proceso de planificación a través de las diferentes herramientas que se seleccionen así como la implementación ya ésta como producto final, para obtener el resultado requerido.
- ✓ **Métodos empíricos:**
  - Entrevista: Se realizaron entrevistas al vicedecano de docencia de la Facultad 5, así como a la planificadora docente con el objetivo de seleccionar los principales requisitos que debe cumplir el sistema.

El presente trabajo estará estructurado en 4 capítulos. A continuación se muestra una breve descripción de cada uno de ellos.

### **Capítulo 1- Fundamentación Teórica:**

En este capítulo se detallan importantes temas a tener en cuenta para el desarrollo del trabajo. Se especifican conceptos como la gestión de información, sistemas de información y la planificación. Se analizan además, otras soluciones existentes en busca de semejanzas con el propuesto. Se analizan las posibles herramientas y metodologías de desarrollo de software a utilizar en la construcción del sistema, especificándose sus principales características y las ventajas de su utilización. Asimismo se tienen en cuenta los diferentes lenguajes, tanto de modelado como de programación y se describen también los posibles patrones a utilizar, desde patrones de casos de uso hasta patrones de diseño y de arquitectura.

### **Capítulo 2- Presentación de la Solución Propuesta:**

Este capítulo estará dedicado a describir detalladamente el negocio actual, basado en el funcionamiento manual de los procesos de planificación de la guardia docente y el cuidado de exámenes en el área de formación de la Facultad 5. Se definen los actores de dicho negocio, los trabajadores y las entidades que manipulan, así como los casos de uso que se manejan, especificándose las relaciones que se establecen entre ellos. Se realiza además, el levantamiento de los requisitos, tanto los funcionales como los no funcionales, que sirven de entrada a la definición de los casos de uso que se implementarán en el sistema, los cuales estarán acompañados de sus respectivas descripciones textuales.

### **Capítulo 3- Construcción de la Solución Propuesta:**

En este apartado se define cómo el sistema será implementado dada las funcionalidades requeridas y las restricciones impuestas (requerimientos). Básicamente describe el análisis y el diseño del sistema por medio de diagramas y modelos según lo propone la metodología de desarrollo seleccionada. Los diagramas de clases y diagramas de interacción de cada uno de los casos de uso del sistema constituyen pilares fundamentales para este objetivo. Se expone también, el modelo de datos de la aplicación, especificándose las descripciones de sus tablas, así como las clases persistentes que le dieron origen.

### **Capítulo 4- Implementación y Prueba:**

En este último capítulo se establece la organización de las clases y objetos por medio de componentes y se especifica la distribución física de los nodos o dispositivos de hardware para proceder al despliegue del sistema. Finalmente, luego de la implementación, se diseñan y ejecutan las pruebas al sistema en aras de detectar errores o defectos relacionados con su funcionalidad. Los resultados de dichas pruebas también son mostrados en este capítulo.

### Capítulo 1. Fundamentación Teórica

#### INTRODUCCIÓN

En el presente capítulo se abordan los distintos conceptos que sirven de soporte teórico al desarrollo del sistema a realizar. Se evalúa el estado del arte acerca de los sistemas de gestión y los procesos de planificación docente en general. Se analizan las posibles herramientas y metodologías de desarrollo de software a utilizar en la construcción del sistema, especificándose sus principales características y las ventajas de su utilización. Asimismo se tienen en cuenta los diferentes lenguajes, tanto de modelado como de programación y se describen también los posibles patrones a utilizar, tanto patrones de casos de uso como patrones de diseño y de arquitectura.

#### 1.1 ¿Qué es la gestión?

La gestión es el conjunto de tareas que se realizan para desarrollar un proceso o para lograr un producto determinado a través de la dirección o administración de una empresa o de un negocio. Se plantea que existen dos niveles de la gestión: el lineal y tradicional. Otra forma de considerar la gestión es plantearla como "Una función institucional global e integradora de todas las fuerzas que conforman una organización" [\[10\]](#).

En ese sentido la gestión hace énfasis en la dirección y en el ejercicio del liderazgo. Existen diferentes tipos de gestión como la social, de proyectos, del conocimiento, gestión ambiental, etc. La gestión enfatiza la realización de diligencias conducentes al logro de un negocio o un deseo cualquiera. Involucra el conjunto de trámites que se ejecutan para resolver un asunto o concretar un proyecto. Se puede resumir la gestión como el proceso de planeación y manejo de tareas y recursos [\[10\]](#).

#### 1.2 La Gestión de la Información

En estudios realizados se aprecia que la gestión del conocimiento implica a la gestión de la información. El conocimiento permite el acercamiento a la realidad de la naturaleza del problema y sus soluciones. Los mecanismos para alcanzar el conocimiento radican en cómo manipular la información que se tiene y cómo gestionar la información que se requiere tener. La gestión de la información comienza por saber qué es lo fundamental en la investigación del conocimiento que ha de adquirir y es un soporte de transmisión de

conocimientos que se convierte en algo casi imprescindible. Es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores tomar decisiones documentadas [17].

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información, en función de los objetivos y metas de las compañías y en materia de desempeño y de calidad. Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado para tomar la decisión adecuada. Se agrega además, que la gestión de la información no es un fenómeno nuevo, fue siempre un tema de preocupación para el hombre, desde el momento mismo en que se percató de la limitación de aprovechar únicamente su experiencia con respecto a aquellos que habían logrado alcanzar el conocimiento de las experiencias colectivas y más aún, tener estrategias de búsqueda para poder formarse el conocimiento [10].

### **1.3 Sistemas de Información.**

El estudio de los Sistemas de Información se originó como una subdisciplina de las ciencias de la computación en un intento por entender y racionalizar la administración de la tecnología dentro de las organizaciones. Estos sistemas tratan el desarrollo, empleo y administración de la infraestructura de la tecnología de la información en una organización. Además de ser un conjunto organizado de elementos originados por [14]:

- ✓ Personas.
- ✓ Datos.
- ✓ Actividades o técnicas de trabajo.
- ✓ Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos [14].

Los Sistemas de Información han ido madurando hasta convertirse en un campo de estudios superiores dentro de la administración. Adicionalmente, cada día se enfatiza más como un área importante dentro de la investigación en los estudios de administración, y es enseñado en las universidades y escuelas de negocios más grandes en todo el mundo.

Durante los próximos años, según se ha evidenciado a través de estudios realizados, estos sistemas cumplirán tres objetivos básicos dentro de las organizaciones [16]:

- ✓ Automatización de procesos operativos.
- ✓ Proporcionar información relevante al proceso de toma de decisiones.
- ✓ Lograr ventajas competitivas a través de su implantación y uso.

Un Sistema de Información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información, las cuales se detallan a continuación [15]:

- ✓ Entrada de Información: es el proceso que toma los datos requeridos para procesar la información y las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos. Estos últimos se denominan interfaces automáticas.
- ✓ Almacenamiento de información: el almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.
- ✓ Procesamiento de Información: es la capacidad para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.
- ✓ Salida de Información: la salida es la capacidad para extraer la información procesada o datos de entrada al exterior. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.

Por último se aprecia que según la función a la que vayan destinados o el tipo de usuario final del mismo, los Sistemas de Información pueden clasificarse en (esta clasificación obedece a un punto de vista empresarial) [\[14\]](#):

- ✓ Sistema de Procesamiento de Transacciones (**TPS**): gestiona la información referente a las transacciones producidas en una empresa u organización.
- ✓ Sistemas de Información Gerencial (**MIS**): orientados a solucionar problemas empresariales en general.
- ✓ Sistemas de Soporte a Decisiones (**DSS**): herramientas para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones.
- ✓ Sistemas de Información Ejecutiva (**EIS**): herramientas orientadas a usuarios de nivel gerencial, que permiten monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna y externa a la misma.
- ✓ Sistemas de Automatización de Oficinas (**OAS**): aplicaciones destinadas a ayudar al trabajo diario del administrativo de una empresa u organización.
- ✓ Sistema Experto (**SE**): emula el comportamiento de un experto en un dominio concreto.
- ✓ Sistema Planificación de Recursos (**ERP**): integra la información y los procesos de una organización en un solo sistema.

### 1.4 La Planificación

El mundo de los negocios es un mundo volátil, complejo y sin reglas, pero sobre todo es un mundo incierto. Ante estas características se necesitan herramientas para minimizar el riesgo y la planificación es una de ellas.

Planificar supone reducir el riesgo de una determinada acción al permitir una anticipación a sus consecuencias. Es un proceso continuo que refleja los cambios del ambiente en torno a cada organización y busca adaptarse a ellos [\[18\]](#).

Una de las cuestiones que toda organización debe hacer es la de ordenar el trabajo que se va a desarrollar, el cual siempre debe obedecer a unos objetivos establecidos que previamente deben haber

sido estipulados en la planificación. De esta manera en [18] la planificación es considerada como el ejercicio a ejecutar con el fin de reducir el riesgo en la empresa y organizar los medios productivos en correspondencia con los objetivos buscados. Su función en las organizaciones se basa en dos propósitos principales: el protector y el afirmativo.

El propósito protector consiste en minimizar el riesgo reduciendo la incertidumbre que rodea al mundo de los negocios y definiendo las consecuencias de una acción administrativa determinada. El propósito afirmativo de la planificación consiste en elevar el nivel de éxito organizacional y en definir el curso de acción y los procedimientos requeridos para alcanzar los objetivos y metas.

El plan establece lo que hay que hacer para llegar al estado final deseado. Es algo esencial en el trabajo de un gestor ya que es el que le permitirá conseguir sus objetivos. Planificar supondrá crear una herramienta con la que conseguir objetivos, una herramienta que ayuda a obtener resultados. Utilizando el símil como recurso literario, se puede manifestar que la planificación es como la locomotora de un tren; pues arrastra las restantes actividades de manera tal que si se detiene su ejecución las demás actividades o el resto de los vagones del tren también se detendrían.

Se puede resumir la planificación como una herramienta de acción para decidir qué se va a hacer y por qué. Consiste en establecer metas y definir cómo se van a conseguir. Crear un plan tiene muchos beneficios pero, sobre todo, clarifica muchas dudas acerca del trabajo a realizar [18]:

- ✓ Define la necesidad de recursos para conseguir objetivos.
- ✓ Clarifica las actividades y las dudas respecto a objetivos buscados.
- ✓ Cuantifica los niveles de desempeño para tener éxito.
- ✓ Establece prioridades.
- ✓ Clarifica debilidades y fortalezas para conseguir objetivos.

### **1.5 Sistema de Gestión.**

Las empresas en el siglo XXI se ven en la necesidad de aumentar el rendimiento operativo en forma sistemática, para poder convertir las presiones de la competencia en ventajas comparativas. Por lo que los

sistemas de gestión pueden ayudar a centrar, organizar y sistematizar los procesos para la gestión y mejora. Algunos de los retos por los que transitan las empresas son:

- ✓ Rentabilidad
- ✓ Competitividad
- ✓ Globalización
- ✓ Velocidad de los cambios
- ✓ Capacidad de adaptación
- ✓ Crecimiento
- ✓ Tecnología

Equilibrar éstos y otros requisitos empresariales puede constituir un proceso difícil y desalentador. Es aquí donde entran en juego los sistemas de gestión antes mencionados, al permitir aprovechar y desarrollar el potencial existente en la organización. El uso de uno de estos sistemas permite renovar constantemente los objetivos, las estrategias, las operaciones y los niveles de servicio de las empresas; además de que la implementación eficaz de los mismos puede ayudar a [\[27\]](#):

- ✓ Gestionar los riesgos sociales, medioambientales y financieros.
- ✓ Mejorar la efectividad operativa.
- ✓ Reducir costos.
- ✓ Aumentar la satisfacción de clientes y partes interesadas.
- ✓ Proteger la marca y la reputación.
- ✓ Lograr mejoras continuas.
- ✓ Potenciar la innovación.
- ✓ Eliminar las barreras al comercio.
- ✓ Aportar claridad al mercado.

### **1.6 Análisis de otras soluciones existentes.**

Varios son los Sistemas de Gestión para la planificación docente encontrados en el análisis realizado. A continuación se citan algunos de ellos:

- ✓ SGD WEB- del centro de demostración TECNAUSA, el cual permite acceder a los horarios, faltas y notas, además de permitir la comunicación entre padres, profesores y alumnos desde casa, sin necesidad de visitar el centro [\[11\]](#).
- ✓ Un reconocido sistema Web, Sistema Generador de Horarios (SIGEH), que brinda servicio en la Universidad Autónoma de Baja California para la Facultad de Ciencias, facilitando la automatización de los horarios de diferentes carreras [\[12\]](#).
- ✓ Sistema de Gestión de Alumnado, Planificación Docente y Campus Virtual, que tiene como funcionalidad aportar funcionalidades que completan y cubren con totalidad la planificación docente universitaria, entre las que se encuentran la gestión de actividades docentes, asignación docente, definición de horarios, gestión de espacios, disponibilidad y situación de los recursos [\[13\]](#).

Ninguno de estos sistemas de gestión se enfoca en cumplir las mismas perspectivas y seguir los mismos objetivos que los que se persiguen en los procesos de la guardia docente y el cuidado de exámenes. Similares fueron los resultados arrojados del estudio realizado en el país sobre la existencia de algún sistema de gestión que se adecuara a los mismos requerimientos, donde no se señaló ningún caso que siguiera la estrategia marcada. Estos resultados se deben, principalmente, a que los procesos de guardia docente y cuidado de exámenes son realizados en la Universidad de las Ciencias Informáticas mediante una sucesión de requisitos propios de la universidad, como se expuso en la introducción del trabajo. En igual manera, no en todas las facultades de la UCI se realizan estos procesos bajo las mismas condiciones ni del mismo tipo. Ello propicia la necesidad de realizar un sistema de gestión para lograr una mejor planificación de la guardia docente y el cuidado de exámenes en el área de formación docente de la Facultad 5 en correspondencia con las estrategias que se siguen en la facultad.

### **1.7 Sistemas Gestores de Base de Datos (SGBD)**

Se entiende por Sistema Gestor de Base de Datos (SGBD) al conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Los principales objetivos que persiguen estos sistemas se citan a continuación:

- ✓ Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- ✓ Mejorar los mecanismos de seguridad de los datos y la privacidad. Se puede distinguir cuatro tipos de contextos para usar mecanismos de seguridad:
  - Seguridad contra accesos indebidos a los datos.
  - Seguridad contra accesos no autorizados a la base de datos.
  - Seguridad contra destrucción causada por el entorno (fuego, inundación, robo).
  - Seguridad contra fallos del propio sistema (fallos del hardware, del software).
- ✓ Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la base de datos (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos.
- ✓ Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la base de datos.
- ✓ Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas.

Los SGBD facilitan el manejo de los datos. En este sentido presentan varias funcionalidades:

- ✓ La definición de los datos.
- ✓ La manipulación de los datos.
- ✓ Garantizar la seguridad e integridad de los datos.
- ✓ La gestión de las transacciones y el acceso concurrente.

Para la medida del rendimiento existen formas o maneras de monitorear los SGBD que van encaminados a identificar los recursos de la máquina que se están consumiendo y proporcionar tiempos específicos del sistema gestor. Existe la posibilidad de realizar programas específicos que midan exactamente el conjunto de características deseadas, basados en el registro del tiempo del sistema antes y después de realizar las operaciones cuyo rendimiento se quiere medir [\[6\]](#).

### 1.7.1 PostgreSQL

PostgreSQL es un potente sistema gestor que posee una gran escalabilidad, pues es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, soportando una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de deshacer los

cambios (rollbacks), subconsultas y transacciones, haciendo su funcionamiento mucho más eficiente y tiene la capacidad de comprobar la integridad referencial, además de almacenar procedimientos en la propia base de datos. Es también, un sistema multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso. Seguidamente se describen algunas de las características fundamentales de este gestor:

- ✓ **DBMS Objeto-Relacional:** 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, optimización de consultas, herencia, y arreglos.
- ✓ **Altamente Extensible:** PostgreSQL soporta operadores funcionales, métodos de acceso y tipos de datos definidos por el usuario.
- ✓ **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- ✓ **MVCC:** MVCC, o Control de Concurrencia Multi-Versión (Multi-Versión Concurrency Control), es la tecnología que PostgreSQL usa para evitar bloqueos innecesarios, eliminando de esta manera las esperas de lecturas para acceder a la información de la base de datos por motivo de que otros usuarios se encuentren escribiendo sobre ella.
- ✓ **Cliente/Servidor:** PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Existe un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intenta conectar a PostgreSQL.

### 1.7.2 MySQL

Es un sistema de gestión de bases de datos relacional multiplataforma bajo licencia GPL (General Public License) de la GNU, creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL y de la marca. Este gestor presenta un diseño multihilo que le permite soportar una gran carga de forma muy eficiente.

Este gestor de bases de datos es, probablemente, el más utilizado en el mundo del software libre debido a su gran rapidez y facilidad de uso. Esta gran aceptación está respaldada en parte, a la infinidad de

librerías y otras herramientas existentes que permiten su empleo a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. En las aplicaciones web hay baja concurrencia en la modificación de datos y el entorno es intensivo en cuanto a la lectura de datos, lo que hace a MySQL ideal para las aplicaciones web.

En [\[20\]](#) se resume un conjunto de las principales características que distinguen a MySQL:

- ✓ Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo ya que múltiples clientes tienen acceso concurrente a una (o más) bases de datos simultáneamente.
- ✓ Interioridades y portabilidad:
  - Su implementación multihilo le permite aprovechar la potencialidad de sistemas multiprocesador.
  - Presenta gran portabilidad entre sistemas.
  - Soporta gran cantidad de tipos de datos para las columnas.
  - Dispone de API's (Interfaz de Programación de Aplicaciones) en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
  - Presenta gran rapidez en búsqueda de BD.
  - Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
  - Multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso.
- ✓ Seguridad: Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. Además presenta esquemas de autenticación basados en usuario-máquina.
- ✓ Internacionalización: Soporte para varios lenguajes y juegos de caracteres.
- ✓ Velocidad: Lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento. Por lo que el principal objetivo es su velocidad y robustez.
- ✓ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

- ✓ Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ✓ Cache de Consultas: Cachea los resultados de las consultas comunes.
- ✓ Fiabilidad, facilidad de uso, escalabilidad.

La versión MySQL 5.0 anexa nuevas características importantes que lo hacen más competitivo con manejadores como SQL Server, Sybase y Oracle. Entre sus nuevas características se tiene:

- ✓ Uso de transacciones ACID (Atomic, Consistent Isolated, Durable): Para construir aplicaciones más seguras mediante commit, rollback, crash recovery y bloqueo por registro.
- ✓ Store Procedures: Para mejorar la programación.
- ✓ Triggers: Para mejorar las reglas del negocio.
- ✓ Vistas: Para que la información sensible sea más segura.
- ✓ Information Schema: Para un fácil acceso a los metadatos.
- ✓ Transacciones Distribuidas(XA): Para soportar transacciones entre múltiples ambientes de bases de datos

### 1.7.3 Oracle

Oracle es un sistema gestor de base de datos relacional (RDBMS), por el acrónimo en inglés Relational Data Base Management System. Se considera uno de los sistemas de bases de datos más completos. Es un sistema robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias. Ayuda a administrar y almacenar grandes volúmenes de datos, estabilidad, escalabilidad y es multiplataforma [\[23\]](#).

Entre sus características se pueden encontrar:

- ✓ Versatilidad: por su cualidad de multiplataforma se pueden crear códigos en Java o en C++ utilizando dicha base de datos tanto en Windows como en Linux. Exporta los datos y los migra desde una plataforma a la otra sin causar problemas de integridad o pérdida de datos.
- ✓ Potencia: ofrece un rendimiento mucho mayor que cualquier otra plataforma de Base de Datos. Permite al administrador asignar sus propias zonas de memoria a los datos y cualidades, se controla en todo momento; tanto el crecimiento como el rendimiento de los distintos esquemas que componen una BD sobre Oracle.

- ✓ Seguridad: la seguridad de Oracle como sistema gestor de base de datos es alta, pues al no tener 100 por ciento de integración con Windows lo hace invulnerable a los defectos que posee el sistema operativo, además de poseer un sistema de seguridad muy avanzado.
- ✓ Complejidad: complejo para los administradores que no estén familiarizados con las bases de datos, su alto rendimiento es directamente proporcional a su nivel de complejidad.

### 1.7.4 Microsoft SQL Server

Microsoft SQL Server es una plataforma de base de datos y análisis de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico. SQL Server ofrece la tecnología y las funciones con las que pueden contar las organizaciones. Con avances significativos en áreas claves de la administración de datos empresariales, la productividad de los desarrolladores y la inteligencia empresarial, las ventajas de SQL Server son considerables:

- ✓ Aprovechamiento de los activos de datos.
- ✓ Aumento de la productividad.
- ✓ Reducción de la complejidad de la tecnología de la información.
- ✓ Disminución del costo total de propiedad.

Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft [\[24\]](#).

### 1.8 Metodologías para el desarrollo del Software

Cada día la producción de software busca adecuarse más a las necesidades del usuario. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo. Por lo que seleccionar la metodología más adecuada es un paso principal para obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un proyecto. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Algunas de las metodologías más utilizadas son Programación Extrema (XP) y el Proceso Unificado de Desarrollo de Software (RUP), ésta última es la que guiará el proceso de automatización del sistema en cuestión.

### 1.8.1 RUP

El Proceso Unificado de Desarrollo del Software (RUP), por sus siglas en inglés (*Rational Unified Process*), fue publicado en 1998 como resultado de varios años de experiencia. Este proceso de desarrollo de software, unido al Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada en la actualidad para el análisis, implementación y documentación de sistemas orientados a objetos [\[19\]](#).

Posee un conjunto de características que propician su recomendación para guiar el desarrollo de un proyecto software [\[19\]](#):

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

El funcionamiento básico de esta metodología está guiado por tres características específicas, pues presenta un desarrollo iterativo e incremental, centrado en la arquitectura y guiado por casos de uso.

- ✓ Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios desean, éstos son captados en el modelo del negocio y representados a través de los requerimientos, por lo que guían el proceso de desarrollo.
- ✓ Centrado en la Arquitectura: la arquitectura muestra la visión común del sistema completo, describiendo los elementos del modelo que son más importantes para su construcción.
- ✓ Iterativo e incremental: Es práctico dividir las partes en miniproyectos donde cada uno es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos a crecimiento del producto.

RUP propone dividir el proceso de desarrollo en ciclos, obteniéndose un producto final al terminar cada ciclo. Estos ciclos a su vez, son divididos en cuatro fases de desarrollo que finalizan con un hito.

- ✓ Fase de Inicio: se identifican y priorizan los riesgos más importantes además de planificar la fase de elaboración y se estima el proyecto de manera apropiada.
- ✓ Fase de Elaboración: se realizan los casos de uso más críticos que se identificaron en la fase de comienzo, teniendo como resultado una línea base de la arquitectura.
- ✓ Fase de Construcción: se crea el producto y la línea base crece hasta convertirse en el sistema completo. Al final, el producto contiene todos los casos de uso aunque puede que no esté el producto libre de errores por lo que se descubrirían y solucionarían en la próxima fase.
- ✓ Fase de Transición: cubre el período durante el cual el producto se convierte en la versión beta. Se corrigen los problemas, se incorporan algunas mejoras por lo que conlleva a actividades como la fabricación, formación del cliente y la corrección de defectos para realizar la entrega.

En el desarrollo de cada fase se construyen artefactos, considerados como los productos tangibles del proceso. Solo por mencionar algunos ejemplos se pudiera mencionar el modelo de casos de uso del sistema, modelo de diseño, modelo de despliegue, etc. Estos artefactos son generados producto de las actividades que desempeñan los roles que se han definido para el desarrollo de cada una de las fases. Un rol no es más que el papel desempeñado por una persona en un determinado momento, resaltando que un mismo individuo puede desempeñar más de un rol a lo largo del proceso [\[19\]](#).

### 1.9 Lenguajes de Modelados

En cualquier proyecto de ingeniería como en la construcción de un gran edificio, un avión, una represa hidroeléctrica, la construcción de un procesador de textos o un software de comunicaciones para Internet, requieren de etapas de modelamiento que permitan experimentar y visualizar el sistema que se construirá. Uniendo varios conceptos y teorías, se puede conceptualizar un lenguaje de modelado como una estandarización de notaciones y reglas, que permitan diagramar o graficar un sistema, o parte de él. La elección de un aceptado lenguaje de modelado es de vital importancia, pues un buen modelamiento del software influye determinadamente, en lograr una adecuada comunicación entre los desarrolladores y los clientes.

### 1.9.1 UML

El Lenguaje Unificado de Modelado (UML), por sus siglas en inglés (Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad. Es un lenguaje estándar para la especificación, visualización, construcción y documentación de artefactos de sistemas de Software. Muy bueno además, para la modelación de negocios y otros sistemas que no son Software. UML representa una colección de las mejores prácticas de ingeniería que tienen una probación exitosa en la modelación de sistemas largos y complejos.

El UML permite la modelación de Sistemas Orientados a Objetos. Utiliza, en su mayor parte, notaciones gráficas para expresar los proyectos de diseño del Software, permitiendo la comunicación en el equipo de desarrollo de manera fácil. De esta manera se pueden desarrollar e intercambiar modelos significativos gracias a la expresividad del lenguaje.

Posee una gran cantidad de propiedades que han sido las que han contribuido a hacer de UML el estándar de la industria en la actualidad [\[19\]](#):

- ✓ Es un lenguaje distribuido y adecuado a las necesidades de conectividades actuales y futuras. Ampliamente utilizado por la industria del software.
- ✓ Reemplaza a decenas de notaciones empleadas por otros lenguajes.
- ✓ Modela estructuras complejas.
- ✓ Las estructuras más importantes que soporta tienen su fundamento en la tecnología orientada a objeto, tales como objetos, clases, componentes y nodos.
- ✓ Comportamiento del sistema: casos de usos, diagramas de secuencia, de colaboración.

### 1.10 Patrones

En términos generales, un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares. A modo de resumen un patrón constituye una solución a un problema en un contexto, codifica conocimiento acumulado por la experiencia en un dominio y se dice que un sistema bien estructurado está lleno de patrones.

- ✓ Contexto: son las situaciones recurrentes a las que es posible aplicar el patrón.
- ✓ Problema: es el conjunto de metas y restricciones que se dan en ese contexto.

- ✓ Solución: es el diseño a aplicar para conseguir las metas dentro de las restricciones.

Los patrones constituyen soluciones técnicas concretas. Se utilizan en situaciones frecuentes y favorecen la reutilización de código, aunque su uso no se refleja específicamente en el código, pues se hace difícil reutilizar la implementación de un patrón.

No existe un único tipo de patrón, sino que se pueden citar tanto los patrones arquitecturales, como los de diseño o los de idioma.

- ✓ Patrones arquitecturales: son aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.
- ✓ Patrones de diseño: son aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.
- ✓ Patrones de idioma: son patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

Seguidamente se tratarán algunos tipos de patrones, específicamente los patrones de diseño incluyendo los patrones de casos de uso.

### 1.10.1 Patrones de casos de uso

Los Patrones de casos de uso son una pareja de problema / solución con un nombre, que codifica (estandariza) buenos principios y sugerencias. Teniendo en cuenta (nombre, solución, problema, explicación, ejemplo de utilización) [\[19\]](#).

Los patrones de casos de uso capturan mejores prácticas para la modelación de casos de uso de los sistemas a desarrollar. Su empleo propicia modelos mantenibles, reusables y entendibles y aumento en la productividad.

Son varios los tipos de patrones de casos de uso a tener en cuenta para la modelación de sistemas. Algunos de los más utilizados son los que siguen: Reglas del Negocio, Patrón CRUD, Sistema de Capas, Múltiples Actores, entre otros.

### 1.10.2 Patrones de diseño

Los patrones de diseño consisten en una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. De manera general un patrón de diseño identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. El uso de estos patrones ha contribuido a fomentar la flexibilidad y extensibilidad en los diseños y han demostrado ser una forma muy útil y exitosa de reutilizar diseños. Estos patrones no solo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores [\[19\]](#).

Son varios los patrones de diseño que existen y que pueden ser utilizados indistintamente en uno u otro modelo. Sin embargo, es casi imposible realizar un modelo o diseño eficiente sin hacer uso de los patrones generales de asignación de responsabilidades o patrones GRASP, por sus siglas en inglés (General Responsibility Assignment Software Patterns). Estos patrones forman un conjunto de cinco propuestas que se basan en describir los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Seguidamente se mencionan los cinco patrones que satisfacen el conjunto de los GRASP. Para la total comprensión se hace importante entender el término responsabilidad de una clase como el contrato u obligación que posee una clase en concreto.

#### **Patrones de asignación de responsabilidades (GRASP)**

- ✓ Experto en información: guía la asignación de responsabilidades relacionadas a asignar una responsabilidad al experto en información – la clase que tiene la información necesaria para la realización de la asignación. Es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo.
- ✓ Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El principio de solución de este patrón se basa en que una instancia de un objeto tiene que ser creada por el objeto que tiene la información para ello.

- ✓ Bajo Acoplamiento: guía la asignación de responsabilidades de manera que el acoplamiento permanezca bajo, lo que significa dar soporte a una escasa dependencia y a un aumento de la reutilización. La solución que brinda el patrón consiste en asignar las responsabilidades de forma tal que las clases se comuniquen con el menor número de clases posibles.
- ✓ Alta Cohesión: guía la asignación de responsabilidades relacionadas con un número relativamente pequeño o manejable. Se basa en cómo mantener la complejidad dentro de límites manejables. Para ello propone asignar a las clases las responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.
- ✓ Controlador: guía la asignación de responsabilidades relacionadas con la gestión de un evento de entrada, o sea, quién debería encargarse de atender un evento del sistema. La solución consiste en asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controlador.

### **Patrones de diseño que utiliza Drupal.**

A continuación se especifican algunos de los patrones de diseño que implementa el CMS Drupal.

- ✓ Singleton (instancia única): consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. Pensando en los módulos y temas de Drupal como objetos, entonces este sigue el patrón Singleton. En general estos objetos no encapsulan datos. Lo que diferencia un módulo Drupal de otro es el conjunto de funciones que este contiene, por lo que debe ser pensado como una clase con una única instancia.
- ✓ Decorator: Este patrón responde a la necesidad de añadir dinámicamente funcionalidades a un objeto. Esto permite no tener que crear sucesivas clases que hereden de la primera incorporando la nueva funcionalidad, sino otras que la implementan y se asocian a la primera. Drupal hace un uso extensivo de este patrón. El polimorfismo con el objeto nodo es un ejemplo claro, pero esto es solo una parte de la potencialidad del sistema de nodos de Drupal. Más interesante es el uso de funciones (hook\_nodeapi) que permite a cualquier módulo extender el comportamiento de todos los nodos. Esta característica permite una amplia variedad de comportamientos que se añadirán a los nodos sin necesidad de crear subclases.
- ✓ Observer: el patrón Observador también conocido como "spider" define una dependencia del tipo uno-a-muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, el observador se

encarga de notificar este cambio a todos los otros dependientes. El patrón Observer también es muy utilizado en Drupal. Cuando es realizada una modificación a un vocabulario del sistema de taxonomía de Drupal, la taxonomía hook (gancho) es llamada en todos los módulos que la implementan. Al implementar el hook estos se han definido como observadores del objeto vocabulario, cualquier cambio que se realice es notificado y actualizado en los dependientes.

### 1.10.3 Patrones arquitectónicos

Los patrones arquitectónicos describen los principios fundamentales de la arquitectura de un sistema de software. Identifica los subsistemas, define sus responsabilidades y establece las reglas y guías para organizar las relaciones entre ellos.

Los sistemas distribuidos pueden agrupar los siguientes estilos arquitectónicos:

- ✓ Modelo Vista-Controlador (MVC): permite que en el modelo se administre el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador). En la vista maneja la visualización de la información y en el controlador controla el flujo entre la vista y el modelo (los datos).
- ✓ Arquitecturas en Capas: básicamente define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual significa que los componentes de una capa solo pueden hacer referencia a los componentes en capas inmediatamente inferiores.
- ✓ Arquitecturas Basadas en Componentes: un componente de software es una unidad de composición con interfaces especificadas contractualmente. Que sea una unidad de composición y no de construcción significa que no es preciso confeccionarla, se puede comprar hecha o se puede producir para que otras aplicaciones de la empresa la utilicen en sus propias composiciones.
- ✓ Arquitecturas orientadas a Servicios: un componente en esta arquitectura podría considerarse como un servicio que puede ser publicado, descubierto e invocado de forma dinámica. Las Características más importantes que presenta son bajo acoplamiento entre los componentes y mayor flexibilidad ante cambios futuros.

**Patrón arquitectónico que utiliza Drupal.**

El hecho de utilizar Drupal hace que sea este sistema el que defina la arquitectura del software. Drupal propone una arquitectura por capas que guarda mucha relación con un MVC (Modelo-Vista-Controlador) ya que utiliza una arquitectura por capas por distinción entre presentación, lógica de la aplicación y gestión de la persistencia de datos.

### 1.11 Lenguajes de Programación y Tecnologías

#### 1.11.1 Lenguajes de Programación y Tecnologías del lado del Cliente

Los lenguajes de programación por parte del cliente se utilizan principalmente para añadir elementos dinámicos a la interfaz del usuario. Además de que son totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. A continuación se definirán los usados en la implementación de la aplicación web.

#### El HTML Dinámico (DHTML)

El DHTML (del inglés Dynamic HTML) es una forma que poseen las páginas de aportar su interactividad. Los navegadores *Netscape Communicator 4.0* y *Microsoft Explorer 4.0*, encargados de mostrar y manipular las páginas Web, cuentan ya con esta característica, la cual está orientada al usuario. El DHTML tiene la ventaja de ser una herramienta con la que se pueden crear efectos que requieren poco ancho de banda para su descarga desde la Web son, precisamente, estos efectos los que aumentan la funcionalidad de la página.

DHTML se puede utilizar para crear animaciones, juegos, aplicaciones, así como para introducir nuevas formas de navegar a través de los sitios Web y para crear un auténtico esqueleto de capas que haciendo uso solo del HTML sería imposible abordar. Aunque muchas de las características del DHTML se podrían duplicar con otras herramientas como Java o Flash, el DHTML ofrece la ventaja de que no requiere ningún tipo de plug-in para su utilización [\[4\]](#).

#### CSS

CSS, por sus siglas en inglés Cascading Style Sheets, en español Hojas de estilo en Cascada, son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML(Lenguaje de Marcado Extensible) pueden ser escritas dentro del código HTML de la página web, o en un archivo aparte y enlazarlas con el mismo.

Una de las características más potentes de la programación con hojas de estilo consiste en la posibilidad para definir los mencionados estilos de todo un sitio web. Esto es posible con la creación de un archivo donde se coloquen las declaraciones de estilos de la página y se enlacen todas las páginas del sitio. De este modo, todas las páginas comparten una misma declaración de estilos y, por tanto, si se realiza alguna modificación, cambiarán todas las páginas. Las Hojas de Estilo en Cascada constituyen un estándar muy amplio, con grandes especificaciones y posibilidades [\[25\]](#).

### **AJAX**

Asynchronous Javascript and XML (Extensible Markup Language), traducido como indica el título, no es más que una forma de programar aplicaciones interactivas para Web. AJAX no es una tecnología en sí misma, sino un término que se refiere al uso de un grupo de tecnologías.

El empleo de AJAX proporciona un conjunto de ventajas respaldadas por una mayor rapidez en las operaciones, se está más cerca de crear realmente "Aplicaciones Web", propicia menos carga del servidor (menos transferencia) y menos ancho de banda. Sin embargo también se pueden citar algunas inconvenientes de su uso al señalarse gran cantidad de código AJAX, lo que hace lento al navegador, no es posible las recomendaciones a vínculos (links) específicos, ni tampoco el cambio de sus estados. También existe falta de Integración con el botón de Retroceder Página del Navegador y no se emiten señales inmediatas sobre la carga de los vínculos.

Existen algunos navegadores que no permiten AJAX:

- ✓ Opera 7 y sus versiones anteriores
- ✓ Microsoft Internet Explorer para Windows versión 4.0 y sus anteriores
- ✓ Microsoft Internet Explorer para Macintosh en todas sus versiones.
- ✓ Dillo

- ✓ Navegadores basados en texto como Lynx y Links.
- ✓ Navegadores para incapacitados visuales (braille) [\[5\]](#).

### **Javascript**

Es un lenguaje de programación utilizado para crear pequeños algoritmos y funciones, encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, pues es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Es un lenguaje de programación simple y permite concretar funcionalidades con alto grado de rapidez. Su empleo es recomendable aún para personas con poca experiencia en programación, pues su sencillez permite asimilarlo y practicarlo con facilidad [\[4\]](#).

Entre las acciones típicas que posibilita realizar Javascript existen dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página como cambios de color o cualquier otro efecto. Por el otro, permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas como calculadoras, agendas, o tablas de cálculo [\[25\]](#).

### **1.11.2 Lenguajes de Programación y tecnologías del lado del Servidor**

Existe un conjunto de lenguajes concebidos para llevar a cabo la ejecución de los sistemas por parte del servidor. Cada uno de ellos explota, de manera más profunda, ciertas características que lo hacen más o menos útiles para el desarrollo de las distintas aplicaciones. La decisión de seleccionarlos también depende, por supuesto, de los requerimientos de la aplicación que se pretende construir. En el dominio de la red, ASP, PHP, PERL y JAVA forman el conjunto de los lenguajes del lado del servidor más ampliamente utilizados para el desarrollo de páginas dinámicas.

### **Perl**

Perl se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre de uso, lo que significa que es

gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

Es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de internet, de manera que el código de los scripts en Perl no se compila sino que cada vez que se desee ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Es además, extensible a partir de otros lenguajes, pues desde él es posible hacer llamadas a subprogramas escritos en otros lenguajes [\[25\]](#).

### PHP

Hypertext Preprocessor, es un lenguaje interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Una de sus características más potentes está dada por el soporte para gran cantidad de gestores de bases de datos, como por ejemplo 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. Recientemente su utilidad ha llegado hasta la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+. Es además, una de las variantes libres que existe para este propósito, lo que propicia menos costos y servidores más baratos que otras alternativas. Es muy rápido, pues permite rápidamente a los desarrolladores la generación dinámica de páginas, 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.

PHP es multiplataforma y funciona tanto para Unix como para Microsoft Windows, de manera que el código creado para Unix no sufre modificaciones al ser ejecutado en Windows y viceversa. Su sintaxis está inspirada en el lenguaje C, ligeramente modificada para adaptarlo al entorno en el que se trabaja, lo que facilita en este sentido, asimilar los recursos o contenidos necesarios para emplearlo. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages). Ofrece soluciones simples y universales para las paginaciones dinámicas de la Web mediante una fácil programación y su diseño elegante lo hace más fácil de mantener y actualizar, a diferencia del código en otros lenguajes.

Debido a su amplia distribución y su característica de código abierto, PHP está perfectamente soportado por una gran comunidad de desarrolladores, lo que propicia una rápida reparación de los fallos de funcionamiento que sean detectados. Existen otros lenguajes como Perl (Practical Extraction and Report Language), ASP y JSP (Java Server Pages) que poseen características similares al PHP aunque presentan rasgos que los marcan y lo hacen diferentes:

- ✓ Velocidad de ejecución: la velocidad es mayor en PHP, seguidos por PERL y JSP.
- ✓ Disponibilidad de recursos: actualmente los más utilizados en la Internet son el PHP y el JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos.
- ✓ Familiaridad con el lenguaje: en las universidades los lenguajes más utilizados por los programadores son el ASP y el PHP [\[4\]](#).

### 1.12 IDEs de desarrollo

Un entorno de desarrollo integrado, (IDE, del inglés: Integrated Development Environment) es un entorno de programación empaquetado como un programa, que brinda una serie de herramientas y facilidades a los programadores y pueden centrarse en uno o a varios lenguajes.

#### 1.12.1 Dreamweaver

Es un editor HTML profesional para diseñar, codificar y desarrollar sitios, páginas y aplicaciones Web. Tanto si desea controlar manualmente el código HTML como si prefiere trabajar en un entorno de edición visual, Dreamweaver le proporciona útiles herramientas que mejorarán su experiencia de creación Web. Las funciones de edición visual de Dreamweaver permiten crear páginas de forma rápida, sin escribir a penas líneas de código. No obstante, si se prefiere crear el código manualmente, Dreamweaver también incluye numerosas herramientas y funciones relacionadas con la codificación. Además, ayuda a crear aplicaciones Web dinámicas basadas en bases de datos empleando lenguajes de servidor como ASP, ASP.NET, ColdFusion Markup Language (CFML), JSP y PHP.

#### 1.12.2 Zend Studio

Zen Studio es un Ambiente de Desarrollo Integrado disponible para diseñadores profesionales que abarca todos los componentes de desarrollo necesarios durante el ciclo de vida de las aplicaciones PHP. Está desarrollado por los impulsores de la tecnología de servidores PHP y posee un completamiento de código

de gran nivel. Está escrito completamente en java, por lo que resulta un poco más lento que otros IDEs pero a la vez esto garantiza que pueda correr en varios sistemas operativos. Zend Studio posee una ayuda abundante además de una referencia completa de funciones predefinidas.

### 1.13 Servidores Web

Básicamente, un servidor web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP (del inglés: Hyper Text Transfer Protocol). Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP).

#### 1.13.1 Internet Information Servers (IIS)

IIS es la solución de Microsoft a las necesidades de las empresas y usuarios de enviar y recibir la información, no sólo de sus clientes sino también en el entorno de la empresa. IIS engloba una serie de herramientas administrativas que le permitirán controlar sitios Web, FTP, SMTP (correo saliente) y servicios de noticias. Dispone también del soporte necesario para crear páginas dinámicas (ASP), lenguaje de aplicaciones para Internet bastante extendido y que permite la conexión y acceso a bases de datos consiguiendo aplicaciones Web dinámicas y escalables y dotando a los sitios Web de una mayor complejidad y rendimiento [\[26\]](#).

#### 1.13.2 Apache

El proyecto del servidor del HTTP de Apache es un esfuerzo de desarrollar y de mantener un servidor del HTTP de código abierto para los sistemas operativos modernos incluyendo UNIX y Windows NT. La meta de este proyecto es proporcionar un servidor seguro, eficiente y extensible que provea de servicios del HTTP en los estándares actuales del HTTP.

Apache ha sido el Servidor Web más popular en Internet desde abril de 1996. La encuesta sobre el servidor Web de noviembre de 2005 netcraft arrojó que más del 70% de los sitios Web en Internet están utilizando Apache [\[27\]](#).

Apache presenta algunas características basadas en su alta capacidad de configuración y en la presencia de bases de datos de autenticación. Posee además, negociado de contenido y tiene amplia aceptación en la red, aunque fue criticado por la falta de una interfaz gráfica que ayude en su configuración [\[27\]](#).

### 1.14 Sistema de Gestión de Contenidos (CMS)

#### 1.14.1 ¿Qué es un CMS?

Los sistemas de gestión de contenidos (CMS), por sus siglas en inglés Content Management Systems, es un software que se utiliza principalmente para facilitar la gestión de webs, ya sea en Internet o en una Intranet, y por eso también son conocidos como gestores de contenido web (Web Content Management o WCM). [\[1\]](#)

Los CMS son de gran utilidad para usuarios que no tengan vastos conocimientos técnicos, ya que aportan herramientas para que los creadores se centren en el contenido, ya sea para definir la estructura, formato de las páginas, aspecto visual, uso de patrones o como un sistema modular que permite incluir funciones no previstas.

Por su gran flexibilidad y escalabilidad es que justifican su utilización en cualquier tipo Web. Algunos de los elementos que hacen que sea necesaria la utilización de los CMS son:

- ✓ Inclusión de nuevas funcionalidades en la web.
- ✓ Mantenimiento de gran cantidad de páginas.
- ✓ Reutilización de objetos o componentes.
- ✓ Páginas interactivas.
- ✓ Cambios del aspecto de la web.
- ✓ Consistencia de la web.
- ✓ Control de acceso.

Los CMS según el tipo de licencia se pueden clasificar en CMS propietarios y de código abierto. Estos últimos posibilitan la ventaja que son prácticamente gratis aunque tienen la desventaja de poseer una escasa documentación. A continuación se mencionarán algunos de los CMS de código abierto:

### **Mambo**

Mambo es un sistema de portales CMS basado en el lenguaje de programación PHP y base de datos de código abierto. Basa todo su aspecto en themes.

Dentro de las características principales se encuentran:

- ✓ Base de datos movida por los estándares PHP/MySQL.
- ✓ Módulo de seguridad multinivel para usuarios/administradores.
- ✓ Sección de temas que pueden ser enviados por los usuarios registrados.
- ✓ Plantillas totalmente configurables incluyendo menú central y bloques a izquierda y derecha.
- ✓ Soporte de subida de imágenes para incorporar una propia librería y para usar en cualquier parte del sitio web.
- ✓ Foros dinámicos y encuestas con vista de resultados.

Presenta el inconveniente de escasa ayuda en línea.

### **Joomla**

Joomla es construida por PHP bajo la licencia de GPL y con la utilización de una base de datos MySQL. Algunas de las características que presenta están dadas por:

- ✓ Completa y fácil administración por Web.
- ✓ Creación y administración rápida de una comunidad en línea (on-line).
- ✓ Creación de la Web por inserción de módulos y componentes independientes.
- ✓ Creación y actualización dinámica de secciones, subsecciones y contenidos (públicos y privados).
- ✓ Plantillas (templates) para modificar el diseño gráfico de la Web de forma automática en pocos minutos.
- ✓ Estadísticas de acceso a los contenidos.

### **PHP Nuke.**

PHP-Nuke es un sistema automatizado basado en lenguaje PHP con el que se puede crear páginas Web de contenido fácilmente actualizable, y que permite la participación de todos aquellos que se registren como usuarios o no (la tipología de usuario la define el Superadministrador). Se mantiene 100% interactiva usando una base de datos.

“El sistema es liberado como software libre bajo la licencia GNU/GPL. Hasta la versión 7.5 se podía descargar gratuitamente desde la Web oficial de PHP-Nuke; la versión 7.5 es la primera que requirió un pago de 10 dólares. La última versión, la 8.0 requiere un pago de 12 dólares para ser bajada desde dicha Web. Esto está permitido según la licencia GNU/GPL (con la condición de que el código fuente también sea entregado), y el comprador tiene a su vez el derecho de distribuir el código fuente gratuitamente. El requisito del pago es aplicable solo a la última versión del programa; cuando una nueva versión sale, la anterior se libera y se puede descargar sin costo.” [\[2\]](#)

### **Drupal**

Drupal es un sistema de gestión de contenidos para una web, es modular y muy configurable. Es además usado y mantenido por una gran comunidad de usuarios, lo que lo hace flexible y adaptable. Presenta una gran cantidad de módulos adicionales y permite que los usuarios se puedan registrar e iniciar sesión, así como integrarse con un servidor LDAP.

Ofrece la posibilidad de brindar permisos basados en roles proporcionándole seguridad al sistema. También se pueden crear plantillas con HTML y/o con PHP. En investigaciones acerca del funcionamiento y la usabilidad de este CMS se encontró lo siguiente: “Actualmente es muy usado en la UCI en el desarrollo de los sitios Web de las Comunidades de Investigación y Desarrollo, donde el 100% de ellos se han realizado sobre Drupal, así como para la realización del Portal de las Misiones de la República Bolivariana de Venezuela como parte de los proyectos de exportación que en ella se realizan. Lo que lo atribuye como el CMS más óptimo para ser utilizado en la realización de sistemas Web.

Dentro de las características que presenta están [\[22\]](#):

- ✓ Ayuda en línea: un robusto sistema de ayuda online y páginas de ayuda para los módulos del núcleo, tanto para usuarios como para administradores.
- ✓ Búsqueda: todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.

- ✓ Código abierto: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- ✓ Módulos: La comunidad de Drupal ha desarrollado muchos módulos que proporcionan funcionalidades como páginas de categorías, autenticación mediante jabber, mensajes privados, bookmarks.
- ✓ Personalización: un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo las preferencias definidas por el usuario.
- ✓ URLs amigables: Drupal usa el mod\_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.
- ✓ Gestión de usuarios: permisos basados en roles. Los administradores de Drupal no tienen que establecer permisos para cada usuario, en su lugar, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles [\[21\]](#).
- ✓ Gestión de contenidos: el contenido creado en Drupal es, funcionalmente, un objeto de contenido (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios -o no- sobre cada objeto. El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP. Sindicación del contenido Drupal exporta el contenido en formato RDF/RSS para ser utilizado por otros sitios web. Esto permite que cualquiera con un 'Agregador de Noticias', tal como *NetNewsWire* o *Radio UserLand* visualice el contenido publicado en la web desde el escritorio [\[21\]](#).
- ✓ Características de comunidad: Drupal proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro. Drupal incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables. Incorpora además, foros de discusión para crear sitios comunitarios vivos y dinámicos [\[21\]](#).
- ✓ Rendimiento y escalabilidad: Dentro de ello encontramos el Control de Congestión en el cual Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable

y ajustable. Posee además, un mecanismo de cache que elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor [\[21\]](#).

Drupal fue seleccionado dentro de los CMS de tipo genérico como el mejor en el año 2007. Por ser uno de los CMS más completos actualmente y ajustarse a la aplicación que se va a desarrollar, es que se decide la utilización del mismo.

### **1.15 Propuesta de tendencias y tecnologías a utilizar:**

Después de haber mostrado las tecnologías y herramientas que hoy enriquecen y abarcan a la informática, se llegó a la conclusión que se utilizará como metodología para el desarrollo del software a RUP, debido a que presenta una forma muy disciplinada de asignar tareas y responsabilidades, además de implementar las mejores prácticas de ingeniería. Para la modelación se propone UML por ser un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, basado en la modelación de sistemas con tecnología orientada a objetos. Dicha modelación puede ser desarrollada en la herramienta CASE Rational Rose, la cual soporta de forma completa todas sus especificaciones.

Como lenguaje de programación por el lado del cliente se utilizará JavaScript, un conocido lenguaje basado en su sencillez y pensado para hacer las cosas con rapidez. Como lenguaje del lado del servidor se utiliza el lenguaje PHP por su sencillez, rapidez e interacción, además porque soporta una gran cantidad de gestores de bases de datos. En cuanto al almacenamiento de datos se utilizará el gestor de base de datos MySQL ya que brinda una rápida velocidad, seguridad así como una implementación multihilo. Se utilizará como servidor web al Apache debido a su alta capacidad de configuración. Se utilizará el CMS Drupal el cual hace posible manejar el contenido y darle en cualquier momento, un diseño distinto al sitio sin tener que darle formato al contenido nuevamente. El sistema al estar implementado utilizando este CMS la arquitectura y los patrones usados se heredan del mismo. Aunque Drupal no se puede enmarcar en los paradigmas de la programación orientada a objetos (POO), éste utiliza algunos conceptos como son: objetos, abstracción, encapsulamiento, polimorfismo, herencia y el diseño de patrones (singleton, decorator, observer). En cuanto a la arquitectura se puede hablar de "modelo-vista-controlador" ya que el CMS hace una excelente separación entre la persistencia de datos y los formularios, la lógica y la vista que emerge como la respuesta HTML al navegador.

### Conclusiones del capítulo

- ✓ Se realizó un estado del arte acerca de los sistemas de gestión y los procesos de planificación docente en general, analizándose las principales soluciones existentes en esta área. Ello permitió concluir que no existe un sistema de gestión que satisfaga las necesidades planteadas en este trabajo.
- ✓ Se estudiaron las metodologías, lenguajes y tecnologías existentes para el desarrollo de sistemas de gestión. Lo que propició la siguiente selección para el desarrollo del sistema:
  - RUP como metodología de desarrollo a utilizar conjuntamente con UML como lenguaje de modelado.
  - JavaScript como lenguaje de programación por el lado del cliente.
  - PHP como lenguaje de programación por el lado del servidor.
  - MySQL como gestor de base de datos.
  - CMS Drupal para la construcción del sistema.
  - Apache versión 2.2.6 como servidor Web

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

### Introducción

En el presente capítulo se realizará la descripción de la propuesta para dar solución a la problemática existente. Para ello se formularán todas las características que tendrá el sistema, partiendo del estudio de los procesos actuales del negocio donde se van a identificar los actores y trabajadores que se benefician o interactúan con el sistema, así como el rol y la responsabilidad que tiene cada uno. Se identifican los requerimientos funcionales y no funcionales, a partir de los cuales se definen los casos de uso del sistema acompañados de sus descripciones.

### 2.1 Modelo del negocio actual

El negocio que requiere la solución informática a desarrollar en este trabajo comprende el desempeño de procesos relacionados con la planificación docente del área de formación de la Facultad 5. En este sentido la planificadora de la facultad se encarga de la ejecución de un conjunto de actividades relacionadas con la planificación de los profesores que les corresponde realizar la guardia docente en cada una de las sesiones concebidas (mañana y tarde de cada día laborable) y la planificación de los profesores para ejecutar el cuidado de los exámenes previstos en cada una de las asignaturas. Básicamente el desarrollo del negocio involucra dos procesos fundamentales: Planificación de la guardia docente y Planificación del cuidado de exámenes. Seguidamente se especifica con más detalles la secuencia de actividades que se realizan en cada uno de ellos.

#### 2.1.1 Planificación de la Guardia Docente

La planificadora solicita a los jefes de departamento un listado con los profesores que ponen a disposición para realizar la guardia docente. Seguidamente agrega dichos profesores en un modelo que se tiene confeccionado en Microsoft Word y se le asignan los días y sesiones correspondientes a ejecutar la guardia docente. Posteriormente este modelo es mostrado a los profesores con el objetivo de validar su conformidad. En caso de algún inconveniente por parte del profesor, la planificadora ejecuta cambios en el modelo (la planificación) de manera iterativa, hasta lograr la conformidad por parte del profesor en función de sus necesidades.

### 2.1.2 Planificación del Cuidado de Exámenes

Para llevar a cabo la planificación del cuidado de examen la planificadora confecciona un listado con todos los profesores disponibles para esta actividad en un documento Word. Estos profesores son todos los que comprenden el claustro de la Facultad 5, exceptuando los profesores adjuntos, aquellos que los jefes de departamento decidan no disponerlos por motivo de priorizar otras actividades, así como los profesores de otras áreas. Seguidamente la planificadora va asignando el día a los profesores que van ejecutando el cuidado y los va marcando, de manera tal que un profesor no cuide más exámenes que otro y el proceso resulte lo más equitativo posible. También se tienen en cuenta las afectaciones de cada profesor, pues si a un profesor le corresponde cuidar examen por el listado y tiene preparación metodológica ese día o tiene que impartir clases, entonces se deja como pendiente para el próximo cuidado.

### 2.2 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externo con los que el negocio interactúa. Seguidamente se especifican los actores que intervienen en el negocio tratado.

Tabla 1: Actores del Negocio

| Actores del Negocio                 | Descripción  |
|-------------------------------------|--|
| Directivo (Vicedecano de Formación) | Persona que obtiene o se beneficia con los resultados de una eficiente planificación de los procesos de planificación docente y cuidado de exámenes. |

### 2.3 Trabajadores del negocio

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

Tabla 2: Trabajadores del Negocio

| Trabajadores del negocio | Descripción  |
|--------------------------|--|
| Planificadora            | Encargada de planificar el proceso de guardia docente y cuidado de exámenes.   |
| Jefe de Departamento     | Encargado de seleccionar a los profesores de su departamento. Además de definir los candidatos a realizar la guardia docente y a participar en los cuidados de exámenes. Ejecuta acciones correctivas con los incumplidores. |
| Profesor                 | Ejecuta el cuidado de exámenes que le ha sido planificado.   |
| Profesor GD              | Ejecuta la guardia docente emitiendo un reporte con los resultados del día a la planificadora.   |

### 2.4 Modelo de casos de uso del negocio.

Describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes. Representa además, las funciones que el negocio pretende realizar y su objetivo básico es detallar cómo el negocio es utilizado por sus clientes y socios.

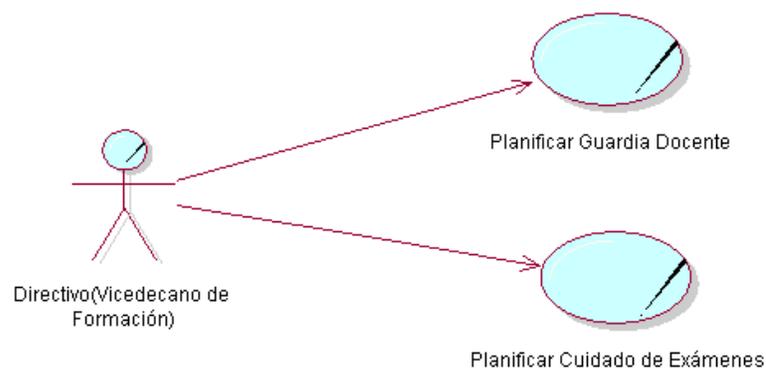


Figura 1: Modelo de Casos de uso del Negocio

### 2.5 Casos de uso del negocio

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Luego de la descripción del negocio a tratar, realizada anteriormente, se definen dos casos de uso.

- ✓ Planificar Guardia Docente.
- ✓ Planificar Cuidado de Exámenes.

Seguidamente se especifican las descripciones textuales que los acompañan.

#### 2.5.1 Descripción de los casos de uso del negocio.

Tabla 3: Descripción del caso de uso Planificar Guardia Docente

|                                    |   |
|------------------------------------|---|
| <b>Nombre del Caso de Uso</b>      | Planificar Guardia Docente  |
| <b>Actores</b>                     | Directivos (Vicedecano Docente)   |
| <b>Propósito</b>                   | Planificar la guardia docente donde se especifique el profesor a realizarla, además de la semana y la sesión en que la realiza.   |
| <b>Resumen</b>                     | El caso de uso se inicia cuando la planificadora le solicita a los jefes de departamento el listado de los profesores disponibles para realizar la guardia docente, los cuales incluye en la planificación que se realizará como resultado final del caso de uso. |
| <b>Curso Normal de los eventos</b> |   |
| <b>Acción del Actor</b>            | <b>Respuesta del proceso de negocio</b>   |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

|  |   |
|--|---|
| <p>1. El vicedecano solicita la planificación de la guardia docente.</p> | <p>1.1 La planificadora solicita a los jefes de departamento los profesores disponibles para realizar la guardia docente.</p> <p>1.2 Los jefes de departamentos envían los nombres a la planificadora.</p> <p>1.3 La planificadora distribuye los nombres recibidos según los horarios establecidos (semana, día y sesión), formalizando así el modelo de planificación.</p> <p>1.4 La planificadora envía a cada profesor la planificación.</p> <p>1.5 El profesor ejecuta la guardia docente en el horario planificado y emite el reporte a la planificadora.</p> |
| <b>Curso Alternativo de los eventos</b>                                  |   |
|  | <p>1.5 El profesor informa a la planificadora sus inconvenientes con el horario planificado, emitiendo así su propuesta.</p> <p>La planificadora atiende la propuesta realizada por el profesor.</p> <p>En caso de ser aceptada dirigirse al paso 1.3.</p> <p>En caso de no ser aceptada dirigirse al paso 1.4.</p>   |
| <b>Prioridad</b>   | Crítico   |

**Tabla 4: Descripción del caso de uso Planificar Cuidado de Exámenes**

|                               |                                       |
|-------------------------------|---------------------------------------|
| <b>Nombre del Caso de Uso</b> | Planificar Cuidado de Exámenes        |
| <b>Actores</b>                | Directivos (Vicedecano de Formación ) |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

|   |   |   |
|---|---|---|
| <b>Propósito</b>  | Planificar el cuidado de exámenes para llevar a cabo una planificación eficiente.   |   |
| <b>Resumen</b>  | El caso de uso inicia una vez que el vicedecano de formación solicita la planificación del cuidado de exámenes. De esta manera la planificadora solicita a los jefes de departamento el listado de los profesores disponibles para cuidar exámenes y a partir de ella conforma la planificación, informándole finalmente al profesor implicado. |   |
| <b>Curso Normal de los eventos</b>                                |   |   |
| <b>Acción del Actor</b>   | <b>Respuesta del proceso de negocio</b>   |   |
| 1. El vicedecano solicita la planificación del cuidado de examen. | 1.1   | La planificadora solicita a los jefes de departamento los profesores que no cuidarán examen porque tienen mucha carga en otras actividades, los adjuntos y los de otras áreas.                |
|   | 1.2   | Los jefes de departamentos envían los nombres a la planificadora.   |
|   | 1.3   | La planificadora, a partir del listado obtenido de los jefes de departamento, conforma un listado con todos los nombres de los profesores de la Facultad 5, disponibles para cuidar exámenes. |
|   | 1.4   | La planificadora, según el orden de la lista, le planifica a cada profesor un examen a cuidar a medida que éstos se vayan efectuando.   |
|   | 1.5   | La planificadora le informa al profesor la planificación correspondiente.   |
|   | 1.6   | La planificadora pasa por los locales en el momento del examen para chequear el cumplimiento de la planificación y actualiza  |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

---

|   |   |
|---|---|
|   | el registro de cuidados para mantener un control sobre el desempeño de cada profesor.   |
| <b>Curso Alternativo de los eventos</b> |   |
|   | 1.5 El profesor le informa a la planificadora su inconveniente con la planificación.<br>La planificadora le informa la nueva planificación o le sugiere que cambie con otro profesor.                 |
|   | 1.6 La planificadora detecta que algún profesor no está cumpliendo con la planificación.<br>La planificadora les informa a los Jefes de Departamento y queda pendiente para la próxima planificación. |
| <b>Prioridad</b>                        | Crítico   |

### 2.6 Definición de los requisitos funcionales.

Los requisitos funcionales no son más que las capacidades o funciones que el sistema debe cumplir. Para lograr el cumplimiento de los objetivos presentados, las funcionalidades que el sistema debe cumplir son:

**RF 1** Autenticar Usuario.

**RF 2** Gestionar Roles.

**RF 2.1** Modificar Roles.

**RF 2.2** Eliminar Roles.

**RF 3** Planificar Cuidado de Exámenes.

**RF 4** Crear Plantilla en la Guardia.

**RF 5** Planificar Guardia Docente.

**RF 6** Listado de exámenes.

**RF 7** Mostrar Plantilla en la Guardia Docente.

**RF 8** Gestionar Grupos.

**RF 8.1** Eliminar Grupos.

**RF 8.2** Modificar Grupos.

**RF 8.3** Crear Grupos.

**RF 9** Mostrar Cuidado de Exámenes.

**RF 10** Mostrar Guardia Docente.

**RF 11** Mostrar Grupos.

**RF 12** Gestionar Actividades.

### **2.7 Definición de los requisitos no funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Permitiendo que puedan marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. A continuación se dan a conocer los requisitos no funcionales del sistema a automatizar:

#### **RNF 1: Apariencia o interfaz externa**

1.1 Debe distinguir a la UCI y a la Facultad 5.

1.2 Debe ser interactivo con el usuario.

1.3 Las funcionalidades deben estar visibles para lograr una mayor navegabilidad.

#### **RNF 2: Usabilidad**

2.1 El sistema podrá ser usado por cualquier persona que acceda a él y requiriéndose conocimientos básicos de computación y trabajo en Web.

#### **RNF 3: Rendimiento**

3.1 El sistema deberá ser capaz de gestionar toda la información referente a la planificación docente de la guardia y el cuidado de exámenes y cargar las páginas dinámicas lo más rápido posible para dar respuesta a las solicitudes de los usuarios.

3.2 Debe ser eficiente a la hora de gestionar las solicitudes, permitiendo alcanzar los resultados deseados sin hacer un uso extensivo de la navegación por el sitio.

3.3 Debe estar disponible las 24 horas del día.

3.4 Soportará a varios usuarios conectados simultáneamente a la base de datos central en cualquier momento dado.

### **RNF 4: Soporte**

- 4.1 Debe ser lo más extensible posible. Puede ser usado en otras áreas docentes.
- 4.2 Fácil para el mantenimiento, de configuración sencilla y factible para los clientes.

### **RNF 5: Portabilidad**

- 5.1 Su implementación sobre PHP propicia que sea una aplicación Multiplataforma. Deberá correr no sólo sobre Windows sino también sobre Linux o Unix para que se pueda llevar a cabo esta acción sin necesidad de efectuar cambios significativos.

### **RNF 6: Seguridad**

- 6.1 La seguridad está bien definida. Se ha establecido un sistema de roles para usuarios diferentes como la planificadora, los jefes de departamento y los profesores. Cada uno de ellos tendrá acceso solo a ejecutar las acciones que le correspondan.

### **RNF 7: Confiabilidad**

- 7.1 Disponible en todo momento debido a su importancia para la planificación docente.

### **RNF 8: Software**

- 8.1 Lenguaje interpretado PHP 5.2.5.
- 8.2 Servidor de aplicaciones Apache versión 2.2.6.
- 8.3 Sistema Gestor de Base de Datos MySQL 5.0.

### **RNF 9: Hardware**

- 9.1 Computadora Pentium III o superior. Requerimiento mínimo 600Mhz.
- 9.2 RAM mínima 128 MB.

## **2.8 Actores del Sistema**

Los actores del sistema representan el rol que desempeña 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. Estos fueron trabajadores del negocio que interactúan con el sistema, en este caso los actores que interactúan con el sistema se definen a continuación:

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

---

### 2.8.1 Descripción de los actores del sistema.

Tabla 5: Actores del Sistema

| Actores del Sistema  | Descripción   |
|----------------------|---|
| Planificadora        | Encargada de supervisar el proceso del sistema a automatizar y de reportar los nombres de los profesores que incumplan con las tareas asignadas. En la guardia docente, es la responsable de velar porque los profesores hayan llenado las plantillas de la guardia correctamente, así como de ver quienes incumplieron con la tarea. |
| Jefe de Departamento | Responsable de seleccionar entre los usuarios que entran al sitio, los que pertenezcan a su departamento. Además de dar a conocer los nombres de los profesores que tienen que realizar la guardia docente y el cuidado de exámenes.  |
| Profesor GD          | Encargado después de realizar la guardia, de llenar correctamente la plantilla para subir la información con la calidad requerida.  |
| Usuario del Sistema  | Es quien intenta ingresar al sistema. Si el proceso de autenticación es correcto, recibe permisos en dependencia del rol que desempeña. El ingreso al sistema es mediante el usuario y contraseña del dominio UCI.  |
| Servidor de Dominio  | Encargado de realizar la validación de la contraseña del usuario perteneciente al dominio UCI.  |
| Profesor             | Encargado de ejecutar el cuidado de exámenes que le ha sido planificado.  |
| Administrador        | Encargado de darle los roles a los usuarios del sistema que realizan la tarea de planificadora y de jefe de departamento.   |

### 2.9 Diagrama de Casos de Uso del sistema

Un diagrama de casos de uso del sistema no es más que la representación gráfica de parte o el total de actores y casos de uso del sistema incluyendo las interacciones entre ellos.

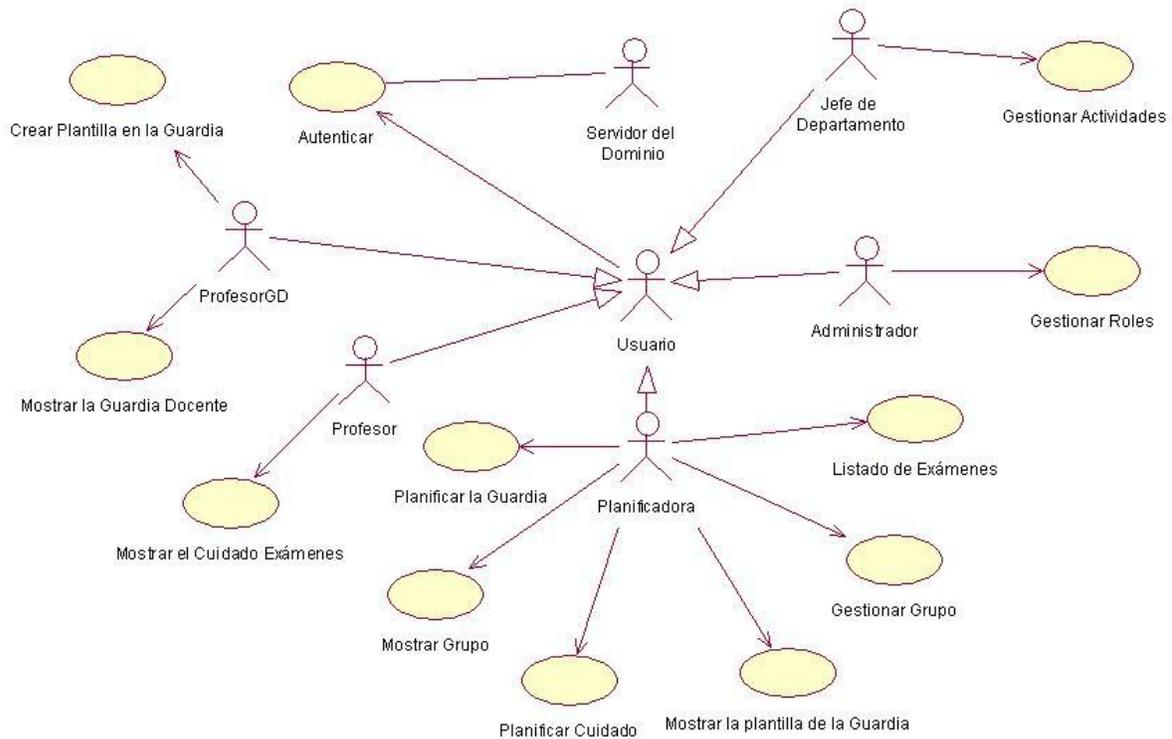


Figura 2: Diagrama de Casos de Uso del Sistema

### 2.10 Casos de uso del sistema a tratar:

- ✓ Autenticar Usuario.
- ✓ Gestionar Roles.
- ✓ Planificar Cuidado.
- ✓ Crear Plantilla en la Guardia.
- ✓ Planificar Guardia.
- ✓ Gestionar Actividades.

- ✓ Gestionar Grupos.
- ✓ Mostrar Cuidado de Exámenes.
- ✓ Mostrar Guardia Docente.
- ✓ Mostrar Plantilla de la Guardia.
- ✓ Mostrar Grupos.
- ✓ Listado de exámenes.

### 2.10.1 Descripción de los Casos de Uso del Sistema.

A continuación se muestran dos de las descripciones de los principales casos de uso del sistema. Las restantes se pueden apreciar en el Anexo 1.

**Tabla 6: Descripción del caso de uso del sistema Planificar Cuidado de Exámenes**

|                                    |   |
|------------------------------------|---|
| <b>Nombre del Caso de Uso</b>      | Planificar Cuidado de Examen  |
| <b>Actores</b>                     | Planificadora (inicia)  |
| <b>Propósito</b>                   | Establecer una planificación del los profesores que cuidarán cada examen concebido.   |
| <b>Resumen</b>                     | El CUS se inicia cuando la planificadora selecciona la opción Planificar cuidado de examen. Posteriormente el sistema define los profesores a cuidar el examen planificado, teniendo en cuenta los datos sobre la planificación de dicho examen y la disponibilidad del profesor, finalizando así el CUS. |
| <b>Referencias</b>                 | RF 3  |
| <b>Precondiciones</b>              | La base de datos debe tener almacenado el listado de los profesores y la cantidad de cuidados asignados a cada profesor.  |
| <b>Poscondiciones</b>              | Planificación de los profesores para un cuidado de examen realizada.  |
| <b>Curso normal de los eventos</b> |   |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

| Acción del actor   | Respuesta del sistema   |
|--|---|
| 1. La planificadora selecciona la acción Planificar cuidado de examen.   | 1.1 El sistema muestra una página con los datos o parámetros (año, asignatura, turno, fecha, cantidad de profesores) correspondientes con la planificación del examen candidato a planificar su cuidado, los cuales deben ser completados por parte de la planificadora para la posterior generación del cuidado. |
| 2. La planificadora llena todas las opciones requeridas y presiona el botón “Siguiente”.   | 2.1 El sistema muestra la planificación del cuidado de examen, especificando los profesores que van a ejecutar el cuidado a partir de la cantidad de profesores definida por la planificadora en la acción anterior. Se muestra también las opciones de grupo y local para ser asignadas a cada profesor.         |
| 3. La planificadora hace corresponder un grupo y un local a cada uno de los profesores designados y presiona el botón “Guardar”. | 3.1 El sistema procede a almacenar en la base de datos la planificación guardada por la planificadora.  |
| Curso alterno de los eventos   |   |
| Acción del actor   | Respuesta del sistema   |
|  | 3.1 Si el local asignado a un profesor se corresponde con Docente 5 o Docente 6 el sistema muestra una validación señalando que “Ha seleccionado un local no válido” y brinda la posibilidad de volverlo a asignar.   |
| Prioridad  | Crítico   |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

**Tabla 7: Descripción del caso de uso del sistema Planificar Guardia Docente**

|   |   |
|---|---|
| <b>Nombre del Caso de Uso</b>   | Planificar Guardia Docente  |
| <b>Actores</b>  | Planificadora (inicia)  |
| <b>Propósito</b>  | Establecer una planificación de los profesores que realizarán la guardia docente en cada día y sesión concebida.  |
| <b>Resumen</b>  | El CUS se inicia cuando la planificadora selecciona la opción Planificar guardia docente. Posteriormente el sistema define los profesores candidatos a ejecutar la guardia docente, especificando además parámetros como el día, la sesión y la paridad de la semana que la planificadora asignará a cada profesor, finalizando así el CUS. |
| <b>Referencias</b>  | RF 5  |
| <b>Precondiciones</b>   | La base de datos debe tener almacenado el listado de los profesores candidatos a ejecutar la guardia docente.   |
| <b>Poscondiciones</b>   | Planificación de los profesores a realizar la guardia docente en cada sesión realizada.   |
| <b>Curso normal de los eventos</b>  |   |
| <b>Acción del actor</b>   | <b>Respuesta del sistema</b>  |
| 1. La planificadora selecciona la acción Planificar guardia docente.                                | 1.1 El sistema muestra los nombres de todos los profesores candidatos a realizar la guardia docente, así como las posibilidades de días, sesiones y paridad de la semana a asignar a cada profesor por parte de la planificadora.   |
| 2. La planificadora asigna un día, una sesión y una paridad a cada uno de los profesores definidos. | 2.1 El sistema almacena en la base de datos la planificación final definida por la planificadora, la cual permanece en la pantalla.   |

## CAPÍTULO 2: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

---

|  |         |
|--|---------|
| Seguidamente presiona el botón "Salvar". |         |
| <b>Prioridad</b>                         | Crítico |

### Conclusiones del capítulo

- ✓ Se describió el negocio en cuestión, especificando sus dos procesos fundamentales: Planificar Guardia Docente y Planificar Cuidado de Examen, así como sus actores y trabajadores involucrados.
- ✓ Se modeló el diagrama de casos de uso del negocio acompañado de las descripciones textuales de cada uno de los casos de uso.
- ✓ Se definieron los requisitos no funcionales y los funcionales que sirvieron como entrada para la definición de los casos de uso del sistema involucrados en el modelo de sistema mostrado y acompañado además de las descripciones textuales de cada uno de ellos.

### CAPÍTULO 3: CONTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

#### Introducción

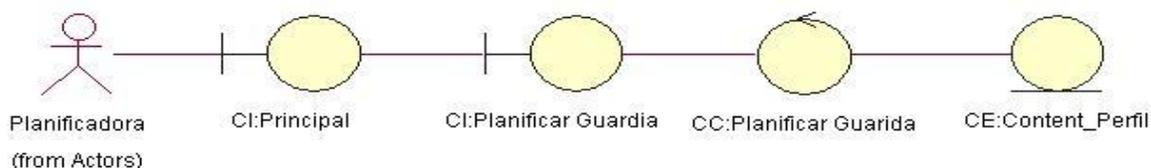
En el presente capítulo, después de haber definido las funcionalidades requeridas, se procederá a la construcción de la solución propuesta. Se especificarán elementos relacionados con el análisis y el diseño del sistema a construir, formalizando en este sentido los diagramas de clases del análisis y los diagramas de clases del diseño, haciendo uso de los estereotipos Web, para cada uno de los casos de uso de la aplicación propuesta.

#### 3.1 Análisis.

El objetivo fundamental del análisis está dado en convertir los requisitos funcionales en un diseño de clases, observando las relaciones e interacciones que existen entre ellos, obteniéndose una visión del sistema. Es representado por las relaciones entre los actores del sistema y las clases de análisis, las que se clasifican en clase de interfaz, controladora y de entidad.

##### 3.1.1 Diagrama de clases de análisis.

A continuación se muestra el diagrama de clase de análisis del caso de uso Planificar Guardia Docente, encontrándose los restantes en el Anexo 2.



**Figura 3 Diagrama de clases del análisis del caso de uso Planificar Guardia Docente**

### 3.1.2 Diagramas de interacción.

Los diagramas de interacción muestran cómo los objetos se comunican entre ellos mediante la transferencia de mensajes. Son diagramas de interacción los de secuencia, los cuales destacan la ordenación temporal de los mensajes y los de colaboración, permitiendo representar enlaces y mensajes entre objetos. Ver los diagramas de colaboración de cada uno de los casos de uso que aparecen en el Anexo 3.

### 3.2 Diseño

El diseño debe de ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Por esta razón es que tiene como propósito adquirir una comprensión de los aspectos relacionados con los requisitos no funcionales. Además de crear entradas apropiadas y un punto de partida para la implementación, para que de una forma más manejable esta pueda ser llevada a cabo por diferentes equipos de desarrollo.

Debido a que la aplicación fue desarrollada utilizando el CMS Drupal el diagrama de clases del diseño incluye los paquetes para la configuración y su debido funcionamiento. Es por ello que en lo adelante se utilizará la terminología paquetes.

Drupal contiene una única página servidor, la cual genera el contenido de la página final. Las páginas pueden o no contener formularios. Ello depende del módulo en cuestión y del propósito que tenga el mismo. A continuación se presenta el diagrama de clases de Drupal y la estructura interna del módulo implementado: SGPD.

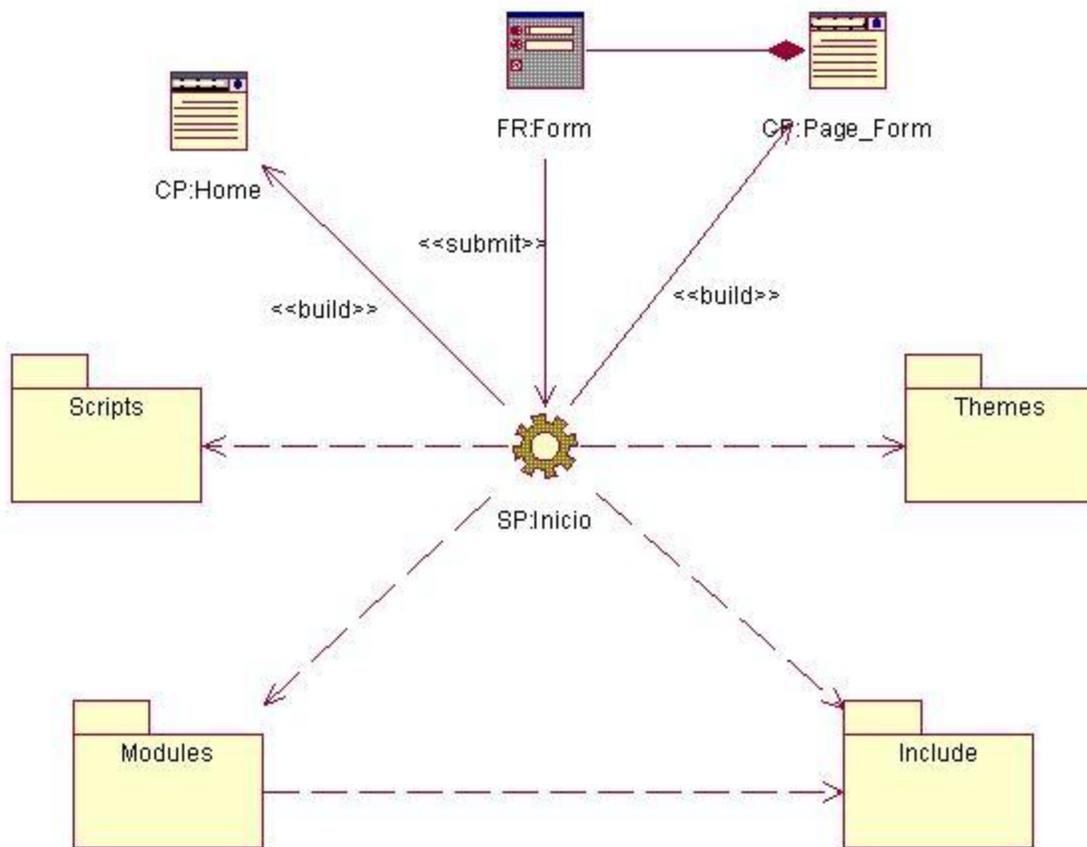


Figura 4: Diagrama de clases del diseño de Drupal

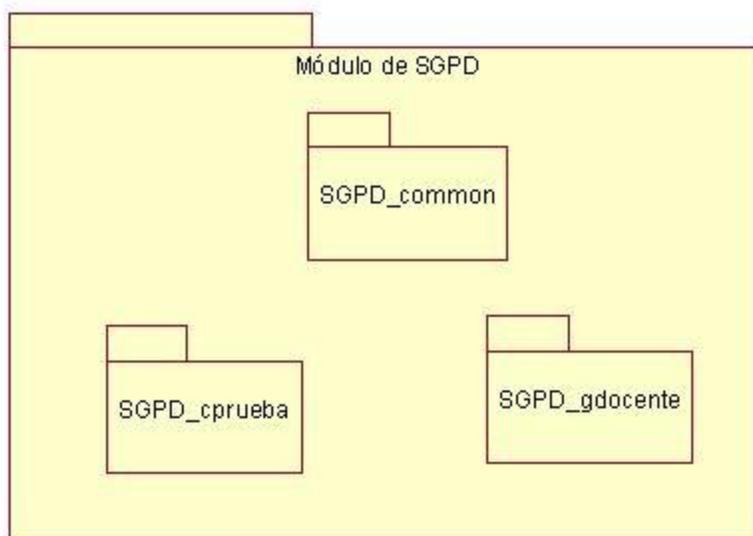


Figura 5: Estructura Interna del Módulo SGPD

### 3.2.1 Descripción de los paquetes de Drupal.

Tabla 8: Paquetes de Drupal

| Paquetes | Descripciones  |
|----------|--|
| Themes   | Contiene las distintas plantillas que permiten configurar el diseño de toda la capa de presentación del sistema, cuando se desea incluir un nuevo diseño, se copia la plantilla en esta carpeta.   |
| Includes | Contiene un conjunto de ficheros indispensables para el funcionamiento de Drupal. Incluye todo un conjunto de implementaciones de clases, los cuales son usados por los módulos. Encontrándose también en él las clases de acceso a datos. |
|          | Contiene todos los módulos que permiten extender las distintas funcionalidades del CMS, cuando se desee incorporar un nuevo módulo se copiaría en esta carpeta.  |

### CAPÍTULO 3: CONTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

---

|         |   |
|---------|---|
| Modules | Algunos de los módulos que se incluyeron para el desarrollo de la aplicación fueron: <ul style="list-style-type: none"><li>• CCK:Permite la creación de tipos de contenido personalizado.</li><li>• Views:Facilita obtener vistas de los contenidos existentes.</li><li>• Administration: Permite la creación de usuarios y roles.</li><li>• SGPD:Tiene los módulos del cuidado de examen y la guarida docente.</li></ul> |
| Scripts | Contiene un conjunto de ficheros indispensables para el funcionamiento del sistema, principalmente orientados al aspecto visual, CSS y JavaScript.  |

3.2.2 Diagrama de clases persistentes.

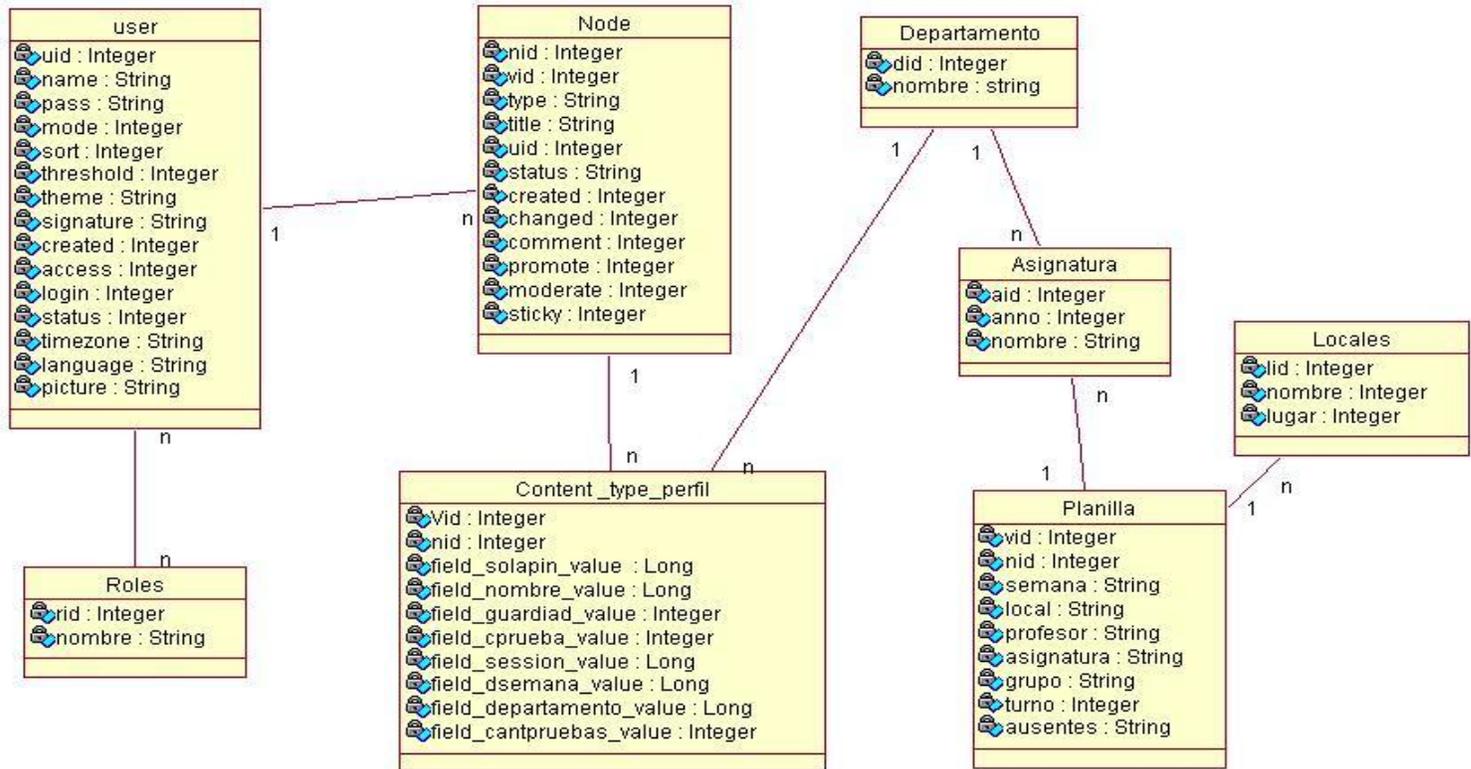


Figura 6: Diagrama de Clases Persistentes

### 3.2.3 Modelo de Datos.

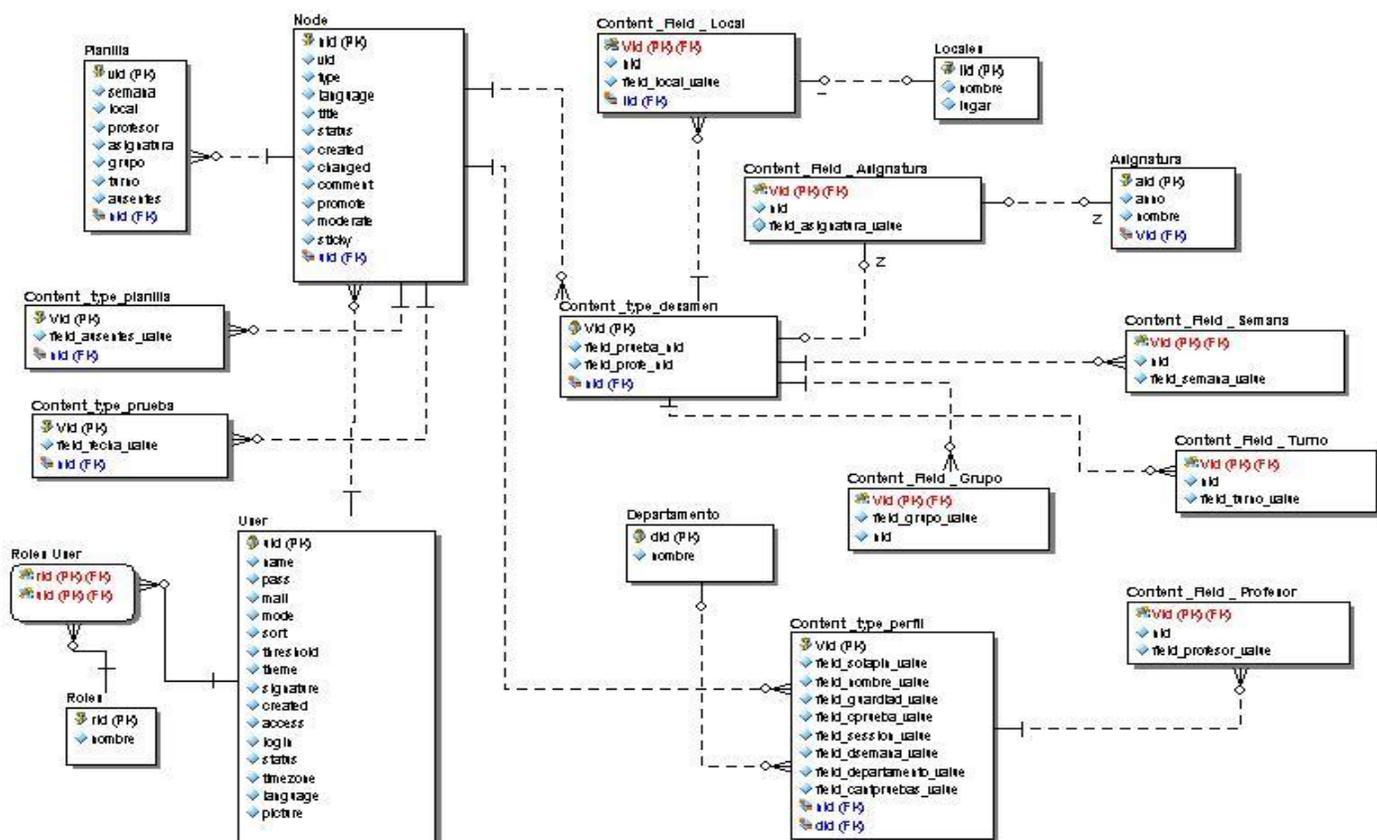


Figura 7: Modelo de Datos

### 3.2.4 Descripción de las tablas de Modelo de Datos.

A continuación se muestra la descripción de la tabla de Cuidado de Examen encontrándose las restantes en Anexos 4.

Tabla 9: Modelo de datos (Asignatura)

|  |             |                    |
|--|-------------|--------------------|
| <b>Nombre:</b> Asignatura  |             |                    |
| <b>Descripción :</b> En esta tabla se guarda el nombre de todas las asignaturas por año. |             |                    |
| <b>Atributo</b>  | <b>Tipo</b> | <b>Descripción</b> |

## CAPÍTULO 3: CONTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

---

|        |             |                                       |
|--------|-------------|---------------------------------------|
| aid    | Int (11)    | Identificador de la tabla Asignatura. |
| anno   | Int(1)      | Año al que pertenece a la asignatura. |
| nombre | Varchar(30) | Nombre de la asignatura               |

### 3.2.5 Tratamiento de Errores.

El tratamiento de errores es un paso fundamental para lograr un buen funcionamiento del sistema. Los usuarios están expuestos a hacer errores en la inclusión y alteración de los datos y para ello se debe tener en cuenta todas las posibles fallas que pueda tener el usuario.

El tratamiento de errores en esta aplicación se realiza con el sistema de captura de errores de Drupal, permitiendo que una vez que ocurre una excepción el cliente es redireccionado a una página de error con el mensaje correspondiente.

### 3.2.6 Seguridad.

En el mundo de hoy en donde mantener la información, bajo la seguridad e integridad requerida, es de una importancia vital, se hace necesaria la existencia de mecanismos que permitan una protección en cuanto a los datos del sistema.

A partir de los principios de poder lograr la integridad, confiabilidad y disponibilidad, de la idea antes expuesta, se tuvo en cuenta para el acceso a la aplicación, una previa autenticación en el sistema donde los únicos que pueden accederlo sean los profesores de la Facultad 5. De esta manera cada profesor según el privilegio o los permisos asignados, podrá acceder únicamente a las tareas o acciones que le corresponde ejecutar. Similarmente ocurre con los roles de Planificadora y Jefe de Departamento. El encargado de asignar los permisos o asignar los roles a cada usuario es el administrador del sistema.

### Conclusiones del capítulo

- ✓ Se desarrolló el análisis y diseño del sistema, permitiendo el avance de la propuesta de solución. En este sentido se construyeron los diagramas de clases de análisis y del diseño para cada caso de uso del sistema, así como los diagramas de interacción correspondientes a cada caso de uso en cada flujo.
- ✓ Se diseñó además la base de datos, conjuntamente con las descripciones de sus tablas correspondientes. De esta manera se traza el camino para la implementación y para las pruebas del sistema posteriormente.

### CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

#### Introducción

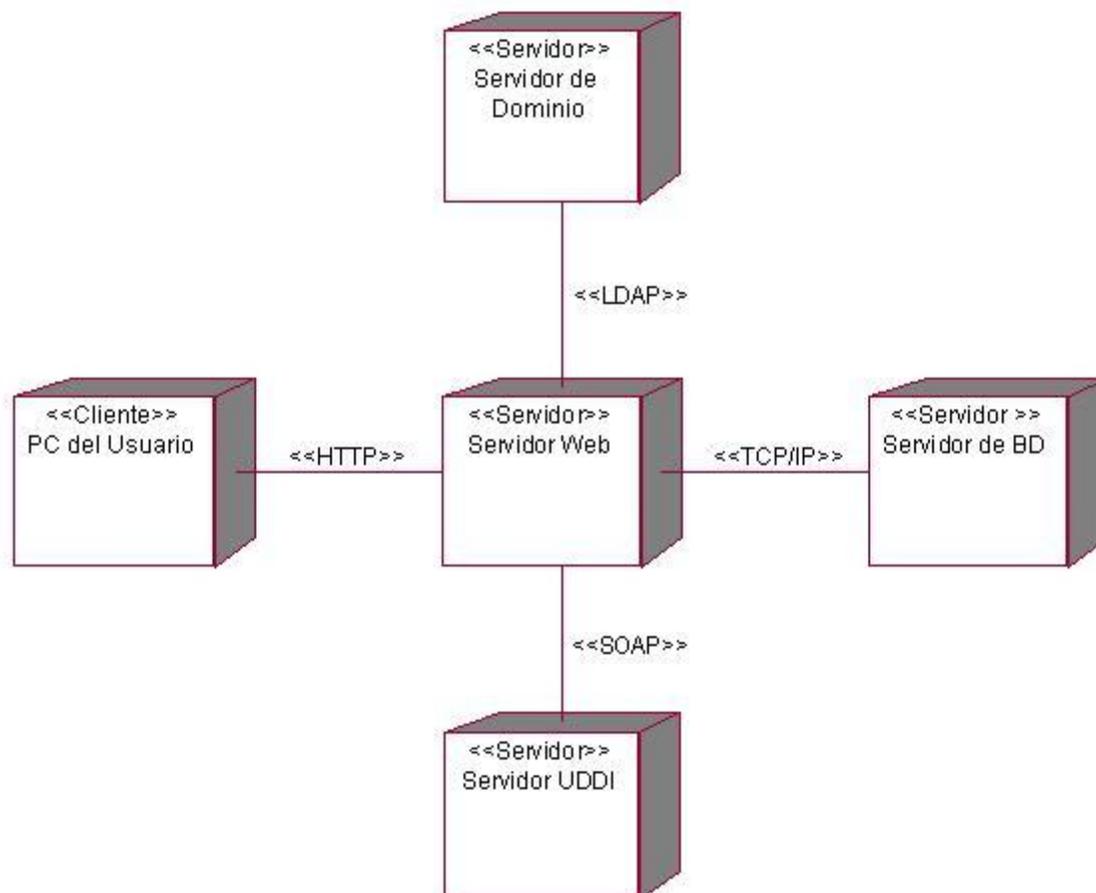
En el presente capítulo se desarrollarán dos flujos de trabajo, el de implementación y posteriormente se le dará paso a las pruebas. Se presentará el modelo de implementación, mediante el diagrama de componentes, obtenido a partir del modelo del diseño. Se diseñará además, el modelo de despliegue de la aplicación mostrando la distribución física de sus nodos. Finalmente se realizará diseñarán y ejecutarán los casos de prueba para chequear la funcionalidad de la aplicación en aras de solucionar los problemas detectados.

#### 4.1 Implementación.

Luego de obtener el resultado del diseño con el modelo obtenido, se procede a ejecutar el flujo de trabajo de implementación. En este sentido se realiza la implementación en términos de componentes, mostrándose su organización de acuerdo a nodos específicos en el modelo de despliegue. Se establece una organización del sistema en términos de subsistemas de implementación que organizados en capas.

##### 4.1.2 Diagrama de Despliegue.

El diagrama de despliegue al igual que el de componentes va a conformar lo que se conoce como un modelo de implementación, ya que describe los componentes a construir, así como la dependencia y organización entre nodos físicos en los que funcionará la aplicación. A continuación se presenta el diagrama de despliegue de la aplicación:



**Figura 8: Diagrama de Despliegue**

#### 4.1.3 Diagrama de Componentes.

El diagrama de componentes se representa como un grafo de componentes de software. Estos componentes pueden ser de código fuente, de bibliotecas, entre otros y unidos además, por medio de relaciones de dependencia, como son compilación y ejecución, logrando que se muestren las interfaces que estos soportan. A continuación se presenta el diagrama de componentes de la aplicación:

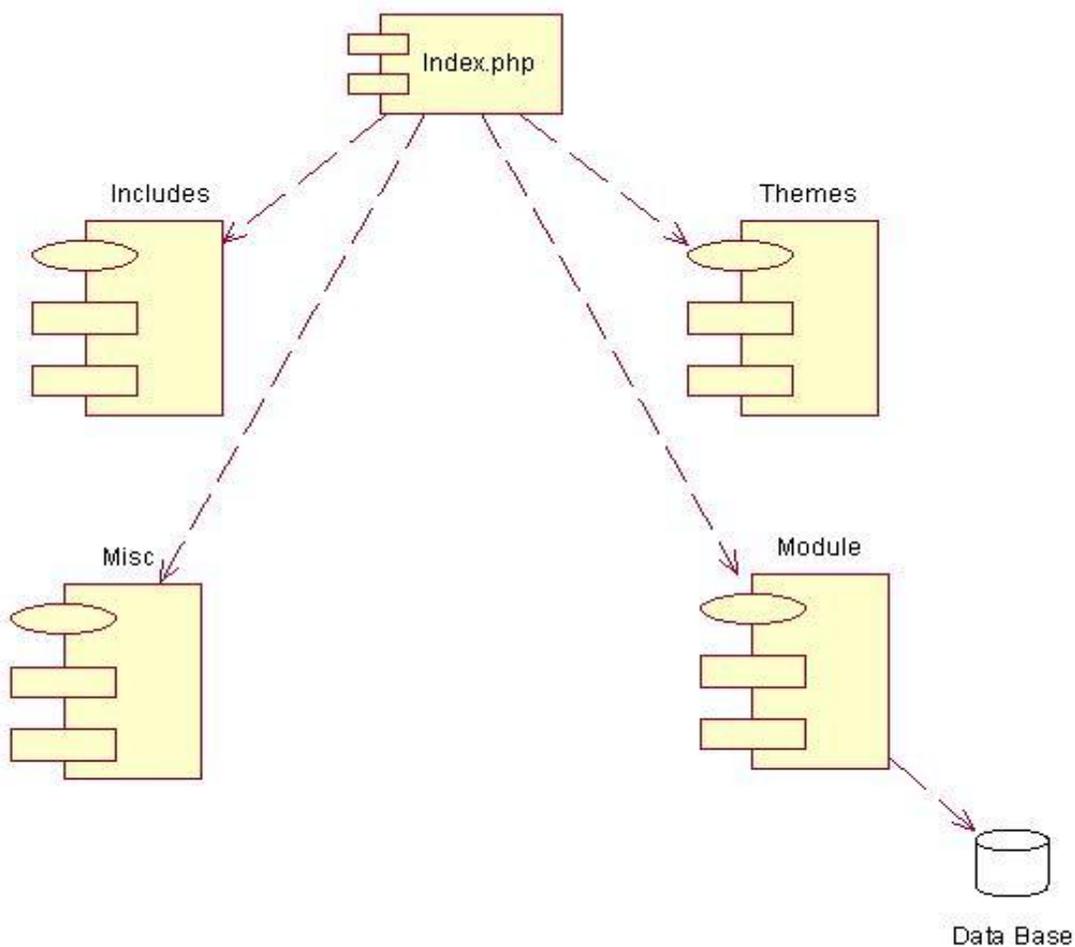


Figura 9: Diagrama de Componentes

### Descripciones del Diagrama de Componentes

Tabla 10: Descripción del Diagrama de Componentes

| Carpetas | Descripción   |
|----------|---|
| Includes | Contiene un conjunto de ficheros indispensables para el funcionamiento del sistema. |

|           |   |
|-----------|---|
| Misc      | Incluye elementos referentes al diseño y funcionamiento de la aplicación (imágenes, ficheros, entre otros)  |
| Modules   | Encierra todos los módulos, que permiten las distintas funcionalidades del CMS, cuando se desee incorporar un nuevo módulo sólo se tiene que copiar en dicha carpeta. |
| Themes    | Incluye las distintas plantillas, cuando se desea implantar un nuevo diseño, se copia la plantilla en dicha carpeta.  |
| Data Base | Bases de datos de la aplicación, donde se almacena toda la información.   |
| Index.php | Es el punto de inicio de la aplicación, a partir de esta entrada se solicitan los diferentes módulos del CMS.   |

### 4.2 Modelo de Prueba.

El desarrollo del software implica una serie de acciones en las que las posibilidades de que aparezcan errores pueden darse desde el primer momento del proceso, éste debe estar acompañado de una actividad que garantice la calidad, las pruebas.

Las pruebas es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados. Los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente [19]. Siendo un elemento que representa una revisión final de las especificaciones del diseño y de las codificaciones, de manera que encontrar un error que no había sido detectado con anterioridad es un objetivo fundamental. De ahí que el objetivo de la prueba sea: “diseñar pruebas que saquen a la luz diferentes clases de errores con la menor cantidad de tiempo y espacio” [19].

Es importante resaltar que las pruebas no pueden asegurar la ausencia de defectos, solo demuestran que existen defectos en el software. Existen dos tipos de métodos para llevar a cabo la realización de las pruebas, los cuales son especificados a continuación.

### 4.2.1 Métodos de Prueba.

Dos de los más conocidos métodos de prueba que se le realizan a un sistema se pueden encontrar: los métodos de caja blanca y los de caja negra, permitiendo que se tracen pruebas en la menor cantidad de tiempo.

La prueba de caja blanca del software comprueba los caminos lógicos del software proponiendo casos de prueba donde se ejerciten conjuntos específicos de condiciones y/o bucles [19]. Estas requieren del conocimiento de la estructura interna de las aplicaciones pudiendo examinar el estado del programa en varios puntos con el objetivo de verificar si el estado real coincide con el esperado.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. De esta manera los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene [19]. Las cuales no consideran la estructura interna del programa ni validan funciones ocultas.

Las pruebas desarrolladas en el sistema en cuestión son las de caja negra, llevadas a cabo sobre la interfaz. Pretendiendo demostrar que las funcionalidades del software son operativas y que las entradas se aceptan de forma adecuada.

### 4.3 Casos de pruebas de la aplicación

En este epígrafe se muestra el diseño de los casos de pruebas a partir de algunos de los casos de uso más significativos del sistema.

#### 4.3.1 Caso de Prueba “CU Autenticar Usuario”.

##### Descripción General

El CUS se inicia cuando un usuario intenta acceder al sistema y en dependencia del rol que desempeñe se le cargará su perfil correspondiente, terminando así el caso de uso. A este Caso de Uso se le realizó la siguiente prueba: Autenticar usuario

**Descripción de la Funcionalidad:** Permitir que cada usuario acceda a la información que le corresponda, en dependencia del rol que desempeñe.

**Flujo Central:**

- ✓ El usuario accede a la aplicación.
- ✓ El usuario selecciona la acción autenticar.
- ✓ El sistema muestra la página correspondiente.
- ✓ El usuario llena todas las opciones requeridas para la autenticación.
  - Usuario
  - Contraseña
- ✓ El usuario presiona el Botón “Log in”.
- ✓ El sistema verifica los datos.
- ✓ En caso correcto, el sistema muestra la interfaz correspondiente al rol.

**Condiciones de Ejecución:** El usuario que desea autenticarse debe estar previamente registrado en la base de datos de la UCI (LDAB).

**Tabla 11: Descripción del Caso de Prueba para el CU Autenticar Usuario**

| Clases Válidas  | Clases Inválidas | Resultado Esperado   | Resultado de la Prueba | Observaciones |
|---|------------------|--|------------------------|---------------|
| <b>Usuario:</b> ghernandez<br><b>Contraseña:</b> ghernandez |                  | El sistema muestra la interfaz correspondiente al rol determinado. |                        |               |
| <b>Usuario:</b> dalvarezh<br><b>Contraseña:</b> dalvarezh   |                  | El sistema muestra la interfaz correspondiente                     |                        |               |

|  |   |   |  |  |
|--|---|---|--|--|
|  |   | al Administrador.   |  |  |
|  | <b>Usuario:</b> lola<br><b>Contraseña:</b> lola | El sistema muestra un mensaje diciendo que el usuario no es válido. |  |  |

### 4.3.2 Casos de Prueba “CU Planificar Guardia Docente”

#### Descripción General

El CUS se inicia cuando la planificadora selecciona la opción Planificar guardia docente. Posteriormente el sistema define los profesores candidatos a ejecutar la guardia docente, especificando además parámetros como el día, la sesión y la paridad de la semana que la planificadora asignará a cada profesor, finalizando así el CUS. A este Caso de Uso se le realizaron las siguientes pruebas: Planificar guardia docente.

#### Descripción de la Funcionalidad:

Esta funcionalidad permite establecer una planificación de los profesores que realizarán la guardia docente en cada día y sesión concebida.

#### Flujo Central:

- ✓ La Planificadora accede a la aplicación.
- ✓ El sistema muestra la interfaz principal de la aplicación.
- ✓ La Planificadora se autentica y el sistema muestra la interfaz de trabajo para el rol correspondiente (Planificadora).
- ✓ La Planificadora selecciona en el menú de la izquierda la opción “Planificar Guardia Docente”.
- ✓ El sistema muestra la interfaz
- ✓ La Planificadora debe introducir los siguientes datos para cada profesor:
  - Día.
  - Sesión.

- Paridad.
- ✓ La Planificadora presiona el botón “Save”.
- ✓ El sistema inserta los datos en la BD.

**Condiciones de Ejecución:** La base de datos debe tener almacenado el listado de los profesores candidatos a ejecutar la guardia docente.

**Tabla 12: Descripción del Casos de Prueba para el CU Planificar Guardia Docente**

| Clases Válidas  | Clases Inválidas   | Resultado Esperado                                      | Resultado de la Prueba | Observaciones |
|---|--|---|------------------------|---------------|
| <b>Día:</b> lunes<br><b>Sesión:</b> Tarde<br><b>Paridad :</b> Impar |  | El sistema guarda los datos correctamente en el sistema |                        |               |
|   | <b>Día :</b> Pepe<br><b>Sesión :</b> Tarde<br><b>Paridad :</b> Par | El sistema no permite escribir nada en los campos.      |                        |               |
|   | <b>Día :</b> Martes<br><b>Sesión :</b> 23<br><b>Paridad :</b> Par  | El sistema no permite insertar números en el sistema    |                        |               |

### 4.3.3 Caso de Prueba: “CU Planificar Cuidado de Exámenes”

#### Descripción General

El CUS se inicia cuando la planificadora selecciona la opción Planificar cuidado de examen. Posteriormente el sistema define los profesores a cuidar el examen planificado, teniendo en cuenta los datos sobre la planificación de dicho examen y la disponibilidad del profesor, finalizando así el CUS. A este Caso de Uso se le realizaron las siguientes pruebas: Planificar Cuidado de Exámenes.

#### Descripción de la Funcionalidad:

Esta funcionalidad permite establecer una planificación del los profesores que cuidarán cada examen concebido.

### Flujo Central:

- ✓ La Planificadora accede a la aplicación.
- ✓ El sistema muestra la interfaz principal de la aplicación.
- ✓ La Planificadora se autentica y el sistema muestra la interfaz de trabajo para el rol correspondiente (Planificadora).
- ✓ La Planificadora selecciona en el menú de la izquierda la opción “Planificar Cuidado de Exámenes”.
- ✓ El sistema muestra la interfaz
- ✓ La Planificadora debe selecciona los siguientes datos:
  - Año
  - Asignatura
  - Turno
  - Fecha
  - Cantidad de Profesores
  - Turno
  - Grupo

**Condiciones de Ejecución:** La base de datos debe tener almacenado el listado de los profesores y la cantidad de cuidados asignados a cada profesor.

Tabla 13: Descripción del Casos de Prueba para el CU Planificar Cuidado de Exámenes

| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado de la Prueba | Observaciones |
|----------------|------------------|--------------------|------------------------|---------------|
|                |                  |                    |                        |               |

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

|   |   |  |  |  |
|---|---|--|--|--|
| <p><b>Año:</b> Primero<br/> <b>Asignatura:</b> Matemática I<br/> <b>Turno:</b> 2do<br/> <b>Fecha:</b> 25 de jun 2009<br/> <b>Cantidad de profesores:</b> 10<br/> <b>Local :</b> Aula 301<br/> <b>Grupo :</b> 5106</p> |   | <p>El sistema guarda los datos correctamente en el sistema</p>                               |  |  |
|   | <p><b>Año:</b> yo<br/> <b>Asignatura:</b> Matemática I<br/> <b>Turno:</b> 2do<br/> <b>Fecha:</b> 25 de jun 2009<br/> <b>Cantidad de profesores:</b> 10<br/> <b>Local :</b> Aula 301<br/> <b>Grupo :</b> 5106</p>        | <p>El sistema no permite escribir nada en los campos.</p>                                    |  |  |
|   | <p><b>Año:</b> Primero<br/> <b>Asignatura:</b> Matemática I<br/> <b>Turno:</b> 2do<br/> <b>Fecha:</b> 25 de jun 2000<br/> <b>Cantidad de profesores:</b> 10<br/> <b>Local :</b> Aula 301<br/> <b>Grupo :</b> 5106</p>   | <p>El sistema no permite seleccionar una fecha inferior a la actual.</p>                     |  |  |
|   | <p><b>Año:</b> Segundo<br/> <b>Asignatura:</b> Matemática III<br/> <b>Turno:</b> 7mo<br/> <b>Fecha:</b> 25 de jun 2009<br/> <b>Cantidad de profesores:</b> 10<br/> <b>Local :</b> Aula 301<br/> <b>Grupo :</b> 5106</p> | <p>El sistema no permite entrar que se introduzca ningún datos que no sean los definidos</p> |  |  |
|   | <p><b>Año:</b> Primero</p>  | <p>El sistema no</p>   |  |  |

|  |  |  |  |  |
|--|--|--|--|--|
|  | <b>Asignatura:</b> Matemática I<br><b>Turno:</b> 2do<br><b>Fecha:</b> 25 de jun 2009<br><b>Cantidad de profesores:</b> 0<br><b>Local :</b> Aula 301<br><b>Grupo :</b> 5106 | permita que se pase el cuidado de sin ninguna cantidad de profesores |  |  |
|--|--|--|--|--|

#### 4.3.4 Caso de Prueba: “CU Gestionar Grupos”

##### Descripción General

El CUS se inicia cuando la planificadora decide adicionar un grupo de la Facultad 5. A este Caso de Uso se le realizaron las siguientes pruebas: Gestionar Grupo

##### Descripción de la Funcionalidad:

Esta funcionalidad permite permitir conocer los grupos que están en la Facultad 5.

##### Flujo Central:

- ✓ La Planificadora accede a la aplicación.
- ✓ El sistema muestra la interfaz principal de la aplicación.
- ✓ La Planificadora se autentica y el sistema muestra la interfaz de trabajo para el rol correspondiente (Planificadora).
- ✓ La Planificadora selecciona en el menú de la izquierda la opción “Gestionar Grupo”.
- ✓ El sistema muestra la interfaz
- ✓ La Planificadora debe seleccionar los siguientes datos:
  - Grupo

**Tabla 14: Descripción del Casos de Prueba para el CU Gestionar Grupos**

| Clases Válidas | Clases Inválidas | Resultado Esperado | Resultado | Observa- |
|----------------|------------------|--------------------|-----------|----------|
|----------------|------------------|--------------------|-----------|----------|

|                     |                     |   | de la Prueba | ciones |
|---------------------|---------------------|---|--------------|--------|
| <b>Grupo : 5306</b> |                     | El sistema guarda los datos correctamente en el sistema           |              |        |
|                     | <b>Grupo : lola</b> | El sistema no permite la entrada de letras en el cuadro de texto. |              |        |

#### **4.4 Resultados de la aplicación**

En los resultados que se obtuvieron de las pruebas que se le realizaron a la aplicación no se encontraron dificultades en ninguno de los requerimientos propuestos.

### Conclusiones del capítulo

- ✓ Se finalizó la etapa de implementación con la realización de los diagramas de componentes y de despliegue correspondientes a la aplicación.
- ✓ Se diseñaron y ejecutaron los casos de prueba para chequear errores de funcionalidad en la aplicación.
- ✓ Se mostraron los principales resultados obtenidos en la ejecución de las pruebas, resaltando el buen funcionamiento de la aplicación.

### CONCLUSIONES

- ✓ Se realizó un estudio del arte acerca de los sistemas de gestión y los procesos de planificación docente, así como de las posibles metodologías, lenguajes y tecnologías a utilizar que permitió la selección de la metodología RUP conjuntamente con el lenguaje UML, JavaScript y PHP como lenguajes de programación, MySQL como gestor de base de datos, Apache en su versión 2.2.6 como servidor Web y CMS para el desarrollo del sistema a realizar.
- ✓ Se obtuvo un Sistema de Gestión para la Planificación Docente en la Facultad 5 “SGPD” que propicia una mayor eficiencia y rapidez en el desarrollo de los procesos de Planificación del Cuidado de Exámenes y Planificación de la Guardia Docente.
- ✓ La aplicación realizada fue sometida a un proceso de pruebas de funcionalidad que arrojó resultados favorables al buen funcionamiento de la aplicación y a la satisfacción del cliente.

## RECOMENDACIONES

- ✓ Poner en práctica la aplicación realizada en el área de Formación de la Facultad 5.
- ✓ Añadir nuevas funcionalidades al sistema favoreciendo la automatización de los demás procesos comprendidos en el área de Formación de la Facultad 5.
- ✓ Mantener la vigilancia tecnológica sobre el gestor de contenidos CMS Drupal para promover su uso en el sistema.
- ✓ Integrar esta aplicación con el sistema de generación de horario que fue desarrollado en otro trabajo de diploma este año.

### BIBLIOGRAFÍA

[1] **Cuerda, García, Xavier y Alfonso, Minguillón, Juliá.** “Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto.” 2004. [Online]. [Citado el: 8 de febrero de 2009.]

Disponible en: <http://mosaic.uoc.edu/articulos/cms1204.html>

[2] **Arceo Claro, Alfonso y Sánchez Pantoja, Maidolys.** “Portal Digital de la Facultad Regional de la UCI en Granma”. La Habana : s.n., 2007. Tesis de diploma.

[3] **Mendoza, María.** Metodologías De Desarrollo De Software. [Fecha de consulta 20 enero 2009]. [Online]

Disponible en:

[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

[4] **Álvarez, Miguel Ángel.** Índice del manual de lenguajes del Web. [Online]. [Fecha de consulta 10 febrero 2009].

Disponible En: <<http://www.desarrolloweb.com/articulos/709.php>

[5]. **Barrios, Mario.** Qué es AJAX. [Online]. [Fecha de consulta 3 febrero 2009].

Disponible en: <http://www.marciobarrios.com/ajax>

[6]. Sistema Gestor de base de datos SGBD. [Online]. [Fecha de consulta 15 enero 2009].

Disponible

En:<[http://www.error500.net/garbagecollector/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_dato.html](http://www.error500.net/garbagecollector/bases_de_datos/sistema_gestor_de_base_de_dato.html)>

PostgreSQL 8.1.x. [Online]. [Fecha de consulta 20 enero 2009].

Disponible en: <<http://www.postgresql.cl/>>

[8]. Tutorial de PostgreSQL. [Online]. [Fecha de Consulta 22 enero 2009].

Disponible en: <<http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>

[9] **Armendariz, Juan Manuel**. Una Herramienta para Gestionar sus Abonados. [Online]. [Fecha de Consulta 11 enero de 2009].

Disponible en: [www.jarmendariz.com.ar](http://www.jarmendariz.com.ar)

[10] **Romero, Yibetza**. Gestión de la tecnología y de la comunicación. [Online]. [Fecha de consulta 20 enero 2009].

Disponible en: <http://www.monografias.com/trabajos53/tecnologia-comunicacion/tecnologia-comunicacion.shtml>

[11] Sistema de Gestión Docente del centro de demostración **TECNAUSA**. [Online]. [Fecha de consulta 2 febrero 2009].

Disponible En: <http://www.tecnausa.com/?alias=ense.presentacion>

[12] Sistema Generador de Horarios. [Online]. [Fecha de consulta 4 febrero 2009].

Disponible En: [http://lcc.ens.uabc.mx/sh2007\\_1/index.jsp](http://lcc.ens.uabc.mx/sh2007_1/index.jsp)

[13] Sistema de Gestión de Alumnado, Planificación Docente y Campus Virtual Integrado. [Online]. [Fecha de consulta 21 enero 2009].

Disponible En: [http://www.ual.es/Universidad/epropias/privado/planes\\_estudio/agora\\_modulos.htm](http://www.ual.es/Universidad/epropias/privado/planes_estudio/agora_modulos.htm)

[14] **González García, Olga Lizeth**. Importancia y manejo de gestión y organización de archivos y documentación en la empresa. [Online]. [Fecha de consulta 5 febrero 2009].

Disponible en: <http://www.monografias.com/trabajos59/gestion-organizacion-archivos/gestion-organizacion-archivos2.shtml>

[15] **Mendoza Pacheco, Henry Jesús**. Introducción a los Sistemas de Información. [Online]. [Fecha de consulta 21 enero 2009].

Disponible en: <http://www.monografias.com/trabajos36/sistemas-informacion/sistemas-informacion.shtml>

[16] **Mperalta**. Sistema de Información. [Online]. [Fecha de consulta 21 enero 2009].

Disponible en: <http://www.monografias.com/trabajos7/sisinf/sisinf.shtml>

[17] **Bartle, Phil**. Información para la Gestión. [Online]. [Fecha de consulta 19 enero 2009].

Disponible en: <http://www.scn.org/mpfc/modules/mon-miss.htm>

[18] **López, Jesús**. ¿Planificación? [Online]. [Fecha de consulta 19 enero 2009].

Disponible en: <http://www.apuntesgestion.com/2008/10/07/planificacion>

[19] Clases de Ingeniería del Software I, curso 2005-2006, UCI.

[20].MySQL 5.0 Reference Manual. [Online]. [Fecha de consulta 18 enero 2009].

Disponible en:< <http://dev.mysql.com/doc/refman/5.0/es/index.html> >

[21] **De la Torre Quintana, Lana y Cruz González, Michel**. “Análisis y Diseño del Sistema de Gestión de Inquietudes de la FEU en la UCI”. Ciudad de la Habana , 2008. Tesis de diploma.

[22] **Leandro Baños, Mara Julia; Santiesteban Quintana, Leyanis y Torres Lopéz, Surayne**. “Sistema de Gestión de la Ubicación Laboral de los Egresados a la UCI”. Ciudad de la Habana , 2008. Tesis de diploma.

[23] **Serrano Hernández, Arniel y Rodriguez Hernández, Armando Alejandro**. Software de Migración de ADABAS a Oracle. Ciudad de la Habana , 2008. Tesis de diploma.

[24] Dumler, Michelle. **Microsoft SQL Server 2005**. [Online][ Fecha de consulta 2 febrero 2009].

disponible en: [www.sqlgurus.org/SQLGURUS/SS2005sodaIntroGLarriera0511.pdf](http://www.sqlgurus.org/SQLGURUS/SS2005sodaIntroGLarriera0511.pdf)

[25] **Nuñez García, Dunel.** Automatización de la gestión y reproducción de las tesis en la UCI. Ciudad de la Habana , 2008. Tesis de diploma.

[26] CIBERAULA. Internet Information Servers 2006[Online][fecha de consulta 2 de febrero de 2009]  
Disponible en: [www.ciberaula.com/curso/iis/que\\_es/](http://www.ciberaula.com/curso/iis/que_es/)

[27] Servidor HTTP Apache.[Online][Fecha de consulta 19 de enero de 2009]  
Disponible en [http://es.wikipedia.org/wiki/Apache\\_http\\_server](http://es.wikipedia.org/wiki/Apache_http_server)

[28] BSI de México. ¿Qué son los Sistemas de Gestión? [Online][Fecha de consulta 19 de enero de 2009]

## GLOSARIO

### A

**Apache:** Es el servidor que se encarga de resolver las peticiones de los clientes de páginas Web.

### B

**BD:** Base de datos.

### C

**CC:** Prefijo para las clases controladoras.

**CE:** Prefijo para las clases entidades.

**CI:** Prefijo para las clases interfaces.

**CMS:** Significa sistema de administración de contenido (en inglés Content Management System), un sistema de este tipo funciona para la creación y administración de contenido, actualmente se usa para denominar a los sistemas que sirven para administrar el contenido de páginas Web.

**CP:** Prefijo para las páginas clientes o que corren del lado del cliente.

**CU:** Una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado.

**Cúmulo:** Montón, agrupaciones.

**CUN:** Caso de uso del negocio.

**CUS:** Caso de uso del sistema.

### D

**Diagrama de Despliegue:** Es un modelo de objetos que representa las relaciones entre el hardware y los nodos físicos.

### H

**HTML:** Lenguaje de Marcas de Hipertexto, es un lenguaje para elaborar páginas Web.

**HTTP:** Protocolo de Transferencia de Hipertexto, es el método más común de intercambio de información en Internet, el método mediante el cual se transfieren las páginas Web a un ordenador.

## I

**IDE:** Entorno integrado de desarrollo.

**Interfaz:** Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.

## L

**Link:** O enlace es un navegador web de código abierto en modo texto —y gráfico a partir de su versión 2 en modo terminal—.

## M

**MySQL:** Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

## P

**Patrón:** Es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

**Perl** - lenguaje de programación de alto nivel que hereda de diversos lenguajes, muy utilizado para el desarrollo de Webs dinámicas.

**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.

**Plug-in:** O complemento como también se le conoce es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúa por medio de la API.

## R

**RUP:** Proceso de desarrollo de software, metodología utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

## S

**SGBD:** Sistemas gestores de base de datos, tipo de software que se utilizan para servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

**SGPD:** Sistema de Gestión de Planificación Docente.

**SP:** Prefijo para las páginas servidoras o que corren del lado del servidor.

## T

**TCP/IP:** Es un conjunto de protocolos de red que permiten la transmisión de datos entre las redes de computadoras.

## U

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

**UML:** “Unified Modelling Language”, lenguaje de modelado gráfico que permite especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

## ANEXOS

## Anexos 1. Descripción de los casos de uso del sistema

## Anexo 1.1 Descripción del caso de uso del sistema Autenticar Usuario

|   |   |  |
|---|---|--|
| <b>Nombre del Caso de Uso</b>   | <b>Autenticar usuario</b>   |  |
| <b>Actores</b>  | Usuarios del sistema (inicia)   |  |
| <b>Propósito</b>  | Permitir que cada usuario acceda a la información que le corresponda, en dependencia del rol que desempeñe.   |  |
| <b>Resumen</b>  | El CUS se inicia cuando un usuario intenta acceder al sistema y en dependencia del rol que desempeñe se le cargará su perfil correspondiente, terminando así el caso de uso.  |  |
| <b>Referencias</b>  | RF 1  |  |
| <b>Precondiciones</b>   | El usuario que desea autenticarse debe estar previamente registrado en la base de datos de la UCI (LDAB).   |  |
| <b>Poscondiciones</b>   | Se cargue o se muestre el perfil del usuario autenticado en el sistema.   |  |
| <b>Curso normal de los eventos</b>  |   |  |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
| 1. El usuario introduce los datos (usuario y contraseña) requeridos para autenticarse y oprime el botón "Log in". | 1.1 El sistema verifica que exista el usuario en la base de datos. Para ello verifica los valores de usuario y contraseña que han sido colocados por el usuario.<br>1.2 Si los valores son válidos, el sistema muestra el perfil correspondiente al usuario autenticado en dependencia de su rol. |  |
| <b>Curso alterno de los eventos</b>   |   |  |
| <b>Acciones del actor</b>   | <b>Respuesta del Sistema</b>  |  |
|   | 1.2 Si el sistema no encuentra el usuario en la base de datos muestra un mensaje informando que el usuario o la contraseña no se han reconocido,  |  |

|                  |  |
|------------------|--|
|                  | manteniendo la página activa para que el usuario vuelva a intentar su autenticación. |
| <b>Prioridad</b> | Crítico  |

#### Anexo 1.2 Descripción del caso de uso del sistema Gestionar Actividades

|  |  |  |
|--|--|--|
| <b>Nombre del Caso de Uso</b>  | <b>Gestionar Actividades.</b>  |  |
| <b>Actores</b>   | Jefe de Departamento(inicia)   |  |
| <b>Propósito</b>   | Poder conocer el listado de los profesores que tienen que realizar la Guardia Docente y los que tienen que Cuidar Prueba.  |  |
| <b>Resumen</b>   | El CUS se inicia cuando el jefe de departamento selecciona gestionar actividades y dado un profesor marca Guardia Docente y/o Cuidado Examen salvando la información y así termina el CUS. |  |
| <b>Referencias</b>   |  |  |
| <b>Precondiciones</b>  | Deben estar al menos un profesor autenticado en el sistema y que pertenezca al departamento.   |  |
| <b>Poscondiciones</b>  | Los profesores quedan asignados a la Guardia Docente y/o Cuidado de Examen.  |  |
| <b>Curso normal de los eventos</b>   |  |  |
| <b>Acciones del Actor</b>  | <b>Respuesta del Sistema</b>   |  |
| 1. El jefe de departamento se dirige a la página de <b>Gestionar Actividades</b> .   | 1.1 El sistema muestra el listado de los profesores del departamento brindándole la opción de Cuidar Examen y/o Guardia Docente.   |  |
| 2. El jefe de departamento marca para cada uno de los profesores Cuidar Examen y/o Guardia Docente y oprime el botón Save. | 2.1 El sistema salva los datos de los profesores seleccionados.<br>2.2 El sistema emite un mensaje informando que los datos fueron correctamente guardado y termina el CUS.                |  |
| <b>Curso alterno de los eventos</b>  |  |  |

| Acciones del Actor |  | Respuesta del Sistema |
|--------------------|--|-----------------------|
|                    |  |                       |
| <b>Prioridad</b>   |  |                       |

**Anexo 1.3 Descripción del caso de uso del sistema Crear Plantilla en la Guardia**

|   |  |   |
|---|--|---|
| <b>Nombre del Caso de Uso</b>   | <b>Crear Plantilla en la Guardia.</b>  |   |
| <b>Actores</b>  | Profesor GD(inicia)  |   |
| <b>Propósito</b>  | Poder llevar un control de la guardia donde se conozca una serie de datos que ello conlleva.   |   |
| <b>Resumen</b>  | El CUS se inicia cuando los profesores concluyen su guardia y van al sistema a llenar la plantilla de la guardia y suben la información, finalizando de esta forma el CUS. |   |
| <b>Referencias</b>  |  |   |
| <b>Precondiciones</b>   | Se deben tener los datos de la plantilla.  |   |
| <b>Poscondiciones</b>   | Que la planificadora pueda ver la plantilla llenada.   |   |
| <b>Curso normal de los eventos</b>  |  |   |
| <b>Acciones del Actor</b>   |  | <b>Respuesta del Sistema</b>  |
| 1. El profesorGD selecciona la opción <b>Crear Plantilla de Guardia.</b>  |  | 1.1 El sistema muestra una interfaz para crear una nueva plantilla.   |
| 2. El profesorGD entra los datos de la guardia (Responsable, Asignatura, Turno, Semana, Local) y oprime el botón. |  | 2.1 El sistema verifica que todos los campos obligatorios estén llenos.<br>2.2 Se insertan los datos de la plantilla en el sistema.<br>2.3 Se muestra un mensaje informando que fue adicionada la plantilla y termina el Caso de Uso. |
| <b>Curso alternativo de los eventos</b>   |  |   |
| <b>Acciones del Actor</b>   |  | <b>Respuesta del Sistema</b>  |
|   |  | 1.1 El sistema muestra un mensaje informando que no fueron llenados todos los campos.   |

|                  |  |
|------------------|--|
|                  |  |
| <b>Prioridad</b> |  |

**Anexo 1.4 Descripción del caso de uso del sistema Mostrar Guardia Docente**

|   |   |  |
|---|---|--|
| <b>Nombre del Caso de Uso</b>   | <b>Mostrar Guardia Docente.</b>   |  |
| <b>Actores</b>  | ProfesorGD (inicia)   |  |
| <b>Propósito</b>  | Permitir conocer la planificación de exámenes.  |  |
| <b>Resumen</b>  | El CUS se inicia cuando el profesor selecciona la opción <b>Listado de la Guardia Docente</b> y puede observar dicha planificación. |  |
| <b>Referencias</b>  |   |  |
| <b>Precondiciones</b>   | Debe estar hecha la planificación de la guardia docente.  |  |
| <b>Poscondiciones</b>   |   |  |
| <b>Curso normal de los eventos</b>  |   |  |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
| 1. El profesorgd selecciona la opción <b>Listado de la Guardia Docente.</b> | 1.1 El sistema muestra la planificación con sus datos correspondientes como lo son Nombre, Día, Sesión, Paridad.                    |  |
| <b>Curso alterno de los eventos</b>   |   |  |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
|   |   |  |
| <b>Prioridad</b>  | Crítico   |  |

**Anexo 1.5 Descripción del caso de uso del sistema Mostrar Cuidado de Exámenes**

|                               |  |
|-------------------------------|--|
| <b>Nombre del Caso de Uso</b> | <b>Mostrar Cuidado de Exámenes.</b>  |
| <b>Actores</b>                | Profesor (inicia)  |
| <b>Propósito</b>              | Permitir conocer la planificación de exámenes.   |
| <b>Resumen</b>                | El CUS se inicia cuando el profesor selecciona la opción Listado del Cuidado de Examen y puede observar dicha planificación. |

|   |   |
|---|---|
| <b>Referencias</b>  |   |
| <b>Precondiciones</b>   | Debe estar hecha la planificación de al menos un examen.  |
| <b>Poscondiciones</b>   | Se listan las asignaturas con la fecha y el turno del examen.   |
| <b>Curso normal de los eventos</b>  |   |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |
| 1. El profesor selecciona la opción <b>Listado del Cuidado de Examen.</b> | 1.1 El sistema muestra la planificación con sus datos correspondientes como lo son Nombre, Asignatura, Turno, Local, Grupo y fecha. |
| <b>Curso alterno de los eventos</b>                                       |   |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |
|   |   |
| <b>Prioridad</b>  | Crítico   |

#### Anexo 1.6 Descripción del caso de uso del sistema Gestionar Grupos

|   |   |
|---|---|
| <b>Nombre del Caso de Uso</b>   | <b>Gestionar Grupos</b>   |
| <b>Actores</b>  | Planificadora (inicia)  |
| <b>Propósito</b>  | Permitir conocer los grupos que están en la facultad 5.   |
| <b>Resumen</b>  | El CUS se inicia cuando la planificadora decide adicionar un grupo de la facultad 5.                        |
| <b>Referencias</b>  |   |
| <b>Precondiciones</b>   | -   |
| <b>Poscondiciones</b>   | Se adiciona un grupo a la facultad 5.   |
| <b>Curso normal de los eventos</b>  |   |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |
| 1. La planificadora selecciona la opción <b>Gestionar Grupo.</b>              | 1.1 El sistema muestra una interfaz para adicionar un nuevo grupo.  |
| 2. La planificadora entra los datos del grupo (grupo) y oprime el botón Save. | 2.1 El sistema verifica que el campo no este vacio.<br>2.2 El sistema verifica que el grupo no exista en la |

|   |  |
|---|--|
|   | base de datos.<br>2.3 El insertan los datos del grupo en el sistema.<br>2.4 Se muestra un mensaje informando que el grupo fue adicionado correctamente y termina el caso de uso. |
| <b>Curso alternativo de los eventos</b> |  |
| <b>Acciones del Actor</b>               | <b>Respuesta del Sistema</b>   |
|   | 1.1 El sistema muestra un mensaje informando que no fueron llenados todos los campos.<br>2. El sistema emite un mensaje informando que ese grupo existe en el sistema.           |
| <b>Prioridad</b>                        | Crítico  |

**Anexo 1.7 Descripción del caso de uso del sistema Mostrar Plantilla de Guardia**

|  |  |
|--|--|
| <b>Nombre del Caso de Uso</b>  | <b>Mostrar Plantilla de Guardia</b>  |
| <b>Actores</b>   | Planificadora (inicia)   |
| <b>Propósito</b>   | Permitir consultar las plantillas realizadas por los profesores de guardia.  |
| <b>Resumen</b>   | El CUS se inicia cuando la planificadora decide consultar las plantillas de realizadas por los profesores de guardia, terminando el CUS cuando se consulta la plantilla. |
| <b>Referencias</b>   |  |
| <b>Precondiciones</b>  | Debe estar creada al menos una plantilla.  |
| <b>Poscondiciones</b>  | Se listan las plantillas de la guardia.  |
| <b>Curso normal de los eventos</b>   |  |
| <b>Acciones del Actor</b>  | <b>Respuesta del Sistema</b>   |
| 1. La planificadora selecciona la opción <b>Listado de Plantilla de Guardia.</b> | 1.1 El sistema muestra el listado de todas las plantillas de guardia y termina el CU.  |

| Curso alterno de los eventos |                       |
|------------------------------|-----------------------|
| Acciones del Actor           | Respuesta del Sistema |
|                              |                       |
| <b>Prioridad</b>             | Crítico               |

## Anexo 1. 8 Descripción del caso de uso del sistema Listado de Exámenes

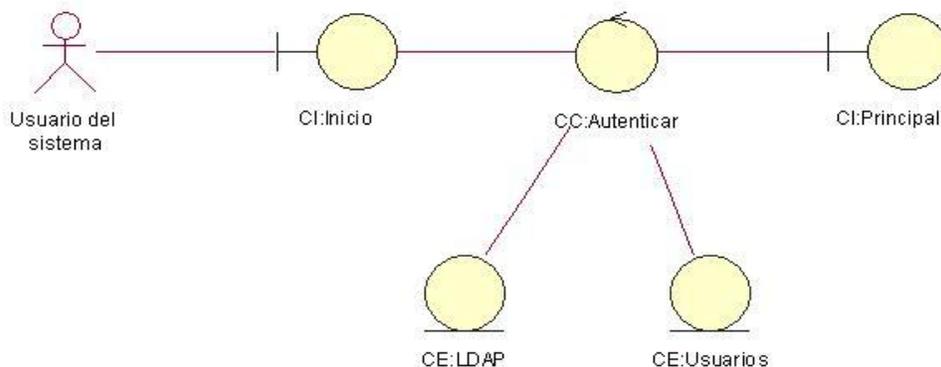
| Nombre del Caso de Uso   | Listado de Exámenes.  |
|--|---|
| <b>Actores</b>   | Planificadora(inicia)   |
| <b>Propósito</b>   | Permite ver la asignatura a la que se le planifico el examen, fecha en que se realizó, ver la distribución de profesores asociados a esa planificación y a demás permite eliminar el examen .                 |
| <b>Resumen</b>   | El CUS se inicia cuando la planificadora selecciona la opción <b>Listado de Exámenes</b> y puede observar los datos correspondientes de un examen como lo son asignatura, fecha y distribución de profesores. |
| <b>Referencias</b>   |   |
| <b>Precondiciones</b>  | Debe estar hecha la planificación de al menos un examen.  |
| <b>Poscondiciones</b>  |   |
| Curso normal de los eventos  |   |
| Acciones del Actor   | Respuesta del Sistema   |
| 1. La planificadora selecciona la opción <b>Listado de Exámenes.</b> | 1.1 El sistema muestra la asignatura a la cual se le planifico el examen, la fecha y el turno en que se realizaran y termina el CU.   |
| Curso alterno de los eventos   |   |
| Acciones del Actor   | Respuesta del Sistema   |
|  |   |
| <b>Prioridad</b>   | Crítico   |

## Anexo 1.9 Descripción del caso de uso del sistema Mostrar Grupos

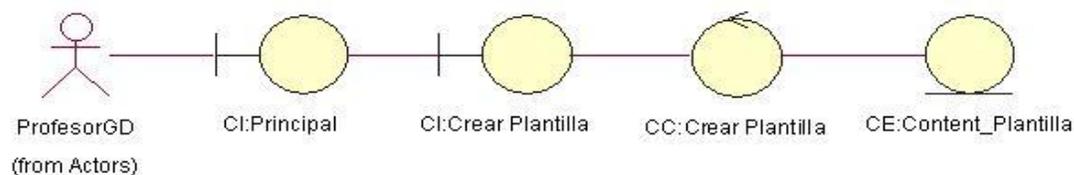
|   |   |  |
|---|---|--|
| <b>Nombre del Caso de Uso</b>                                     | <b>Mostrar Grupos</b>   |  |
| <b>Actores</b>  | Planificadora (inicia)  |  |
| <b>Propósito</b>  | Permitir conocer los grupos que están en la facultad 5.   |  |
| <b>Resumen</b>  | El CUS se inicia cuando la planificadora decide conocer los grupos de la facultad 5, terminando el CUS cuando se tienen los grupos. |  |
| <b>Referencias</b>  |   |  |
| <b>Precondiciones</b>   | Debe estar creado al menos un grupo docente.  |  |
| <b>Poscondiciones</b>   | Se listan los grupos docentes.  |  |
| <b>Curso normal de los eventos</b>                                |   |  |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
| 1. La planificadora selecciona la opción <b>Listado de Grupo.</b> | 1.1 El sistema muestra el listado de todos los grupos y termina el CU.  |  |
| <b>Curso alternativo de los eventos</b>                           |   |  |
| <b>Acciones del Actor</b>   | <b>Respuesta del Sistema</b>  |  |
|   |   |  |
| <b>Prioridad</b>  | Crítico   |  |

**Anexo 2. Diagramas de clases de análisis.**

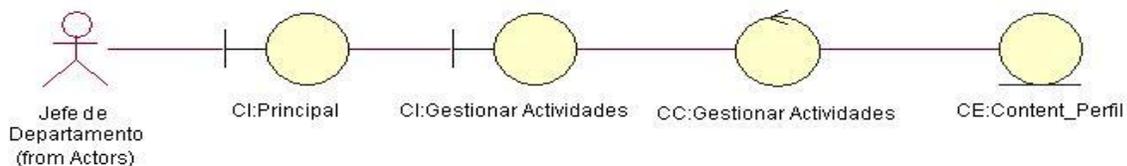
**Anexo 2.1 Diagrama de clase de análisis del caso de uso Autenticar Usuario**



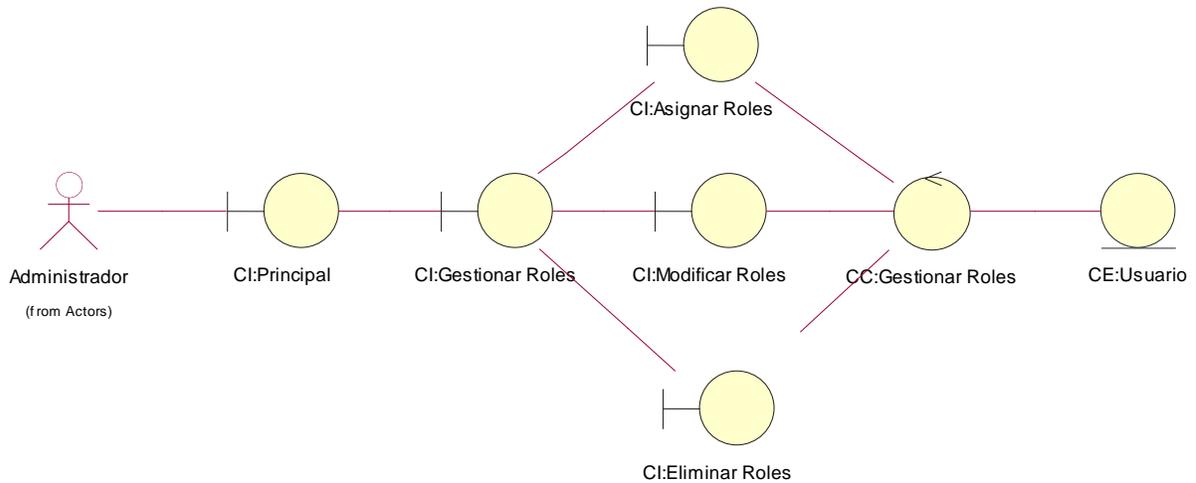
**Anexo 2.2 Diagrama de clases de análisis del caso de uso Crear Plantilla en la Guardia.**



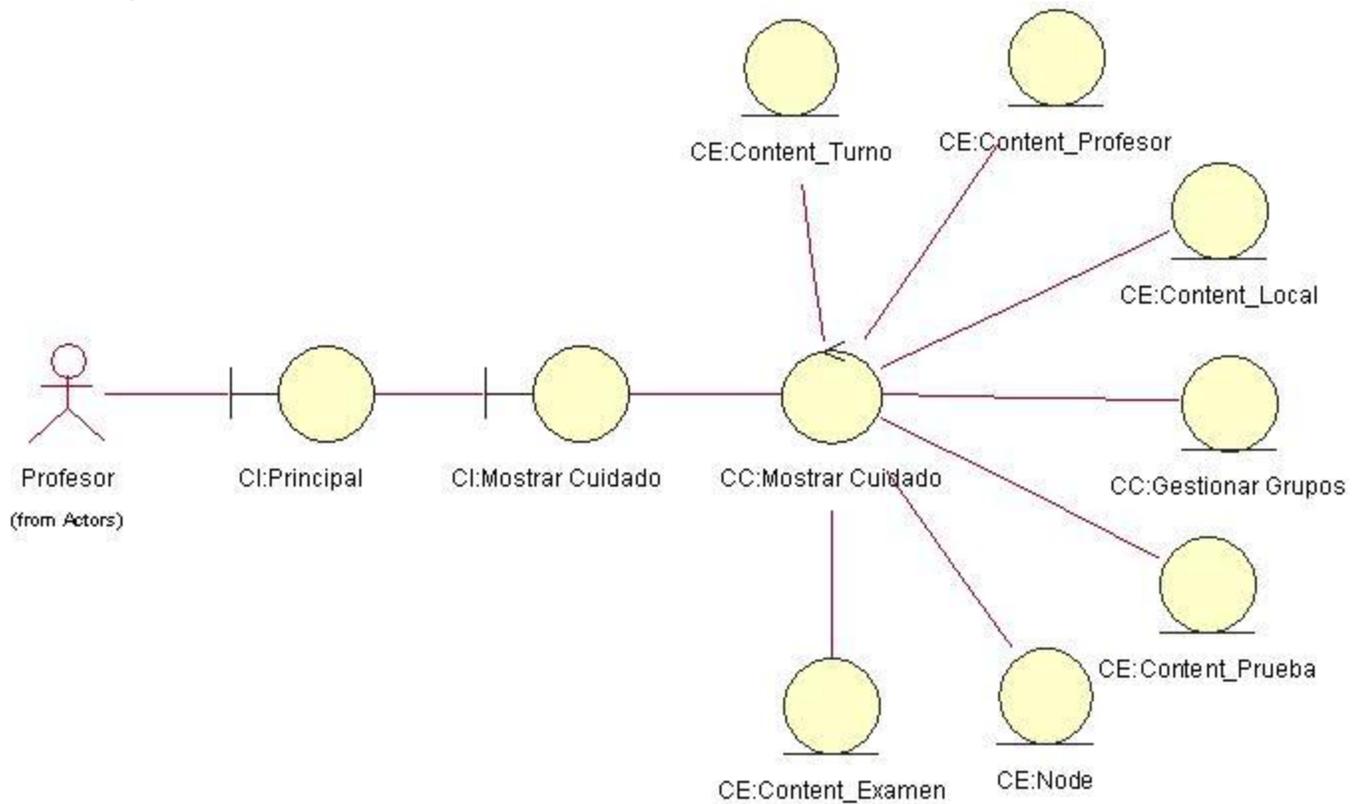
**Anexo 2.3 Diagrama de clases de análisis del caso de uso Gestionar Actividades.**



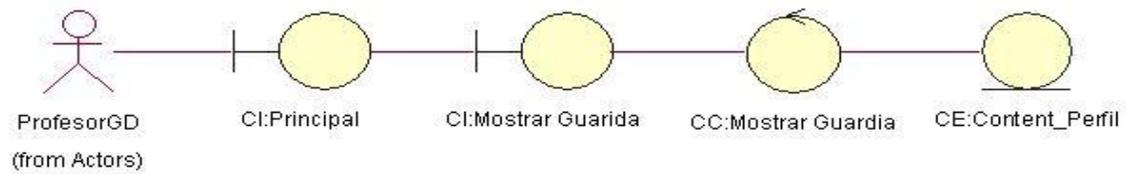
**Anexo 2.4 Diagrama de clases de análisis del caso de uso Gestionar Roles.**



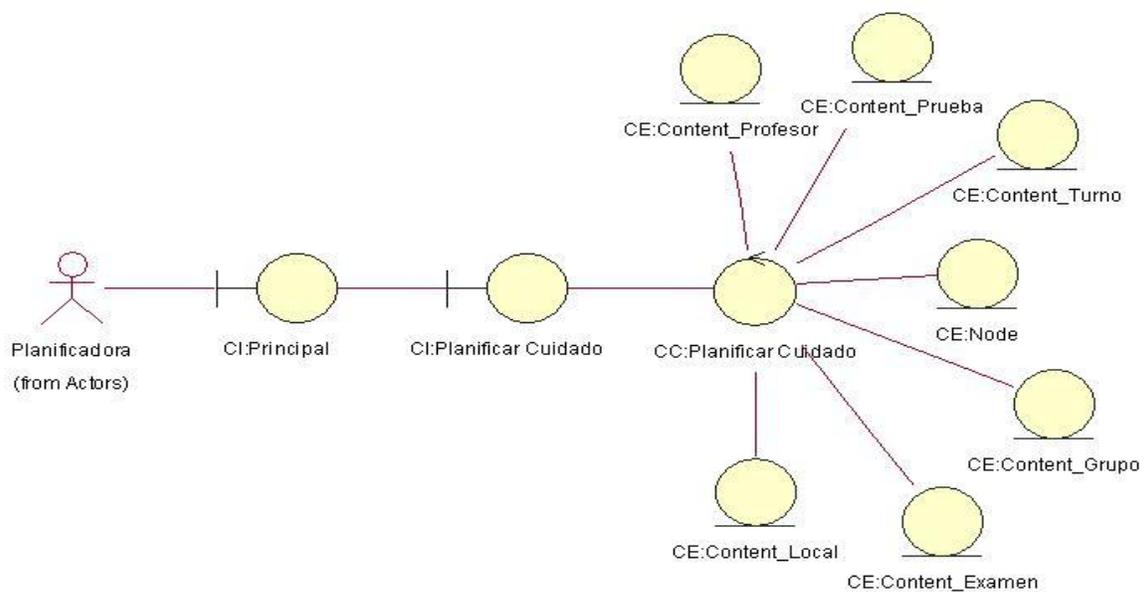
Anexo 2.5 Diagrama de clases de análisis del caso de uso Mostrar Cuidado de Exámenes



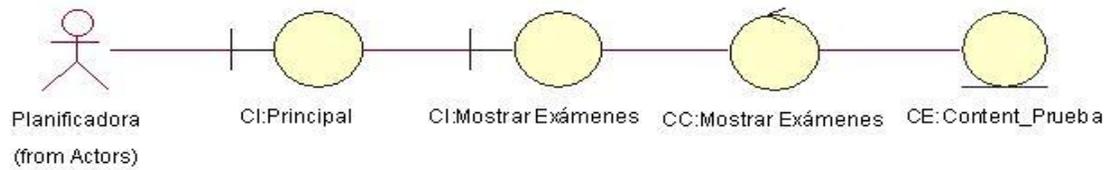
Anexo 2.6 Diagrama de clases de análisis del caso de uso Mostrar Guardia Docente.



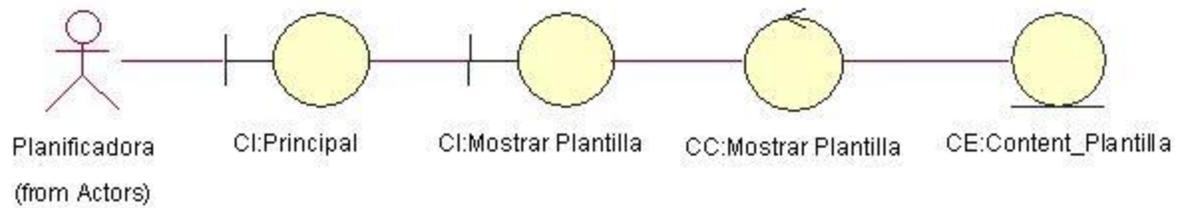
Anexo 2.7 Diagrama de clases de análisis del caso de uso Planificar Cuidado de Exámenes.



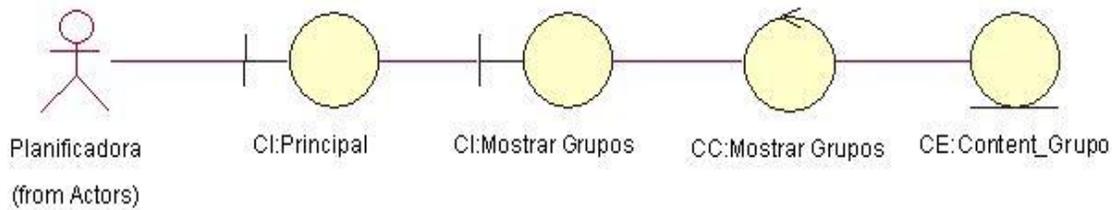
Anexo 2.8 Diagrama de clases de análisis del caso de uso Listar Asignaturas



**Anexo 2.9 Diagrama de clases de análisis del caso de uso Mostrar Plantilla**

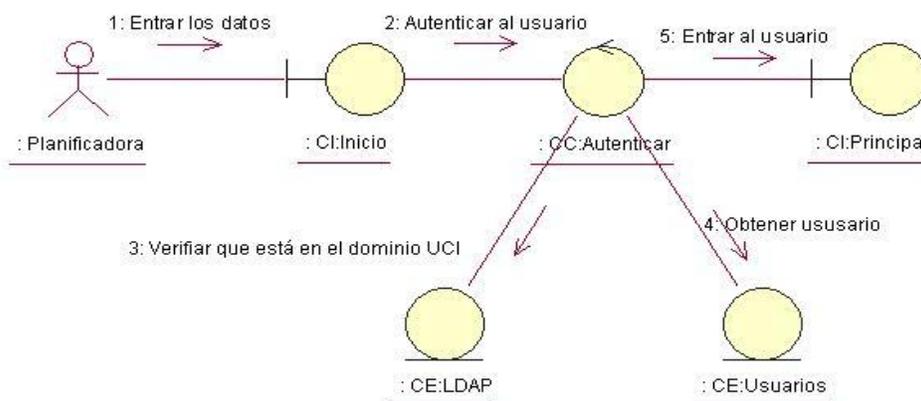


**Anexo 2.10 Diagrama de clases de análisis del caso de uso Mostrar Grupos**

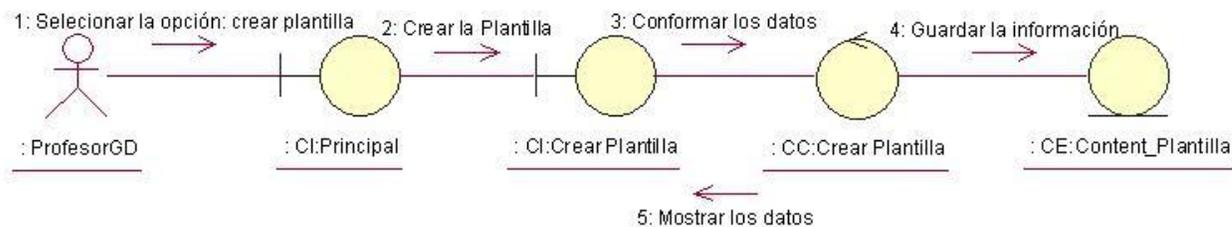


### Anexo 3 Diagramas de Colaboración

#### Anexo 3.1 Diagrama de colaboración del caso de uso Autenticar Usuario



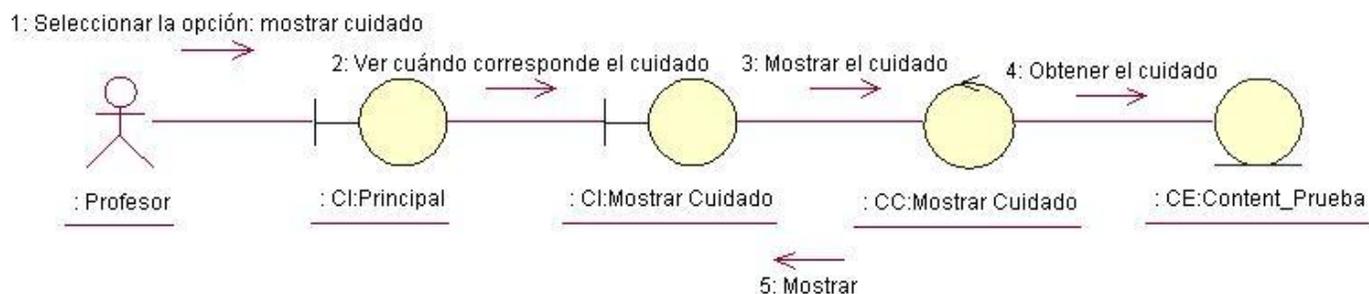
#### Anexo 3.2 Diagrama de colaboración del caso de uso Crear Plantilla en la Guardia.



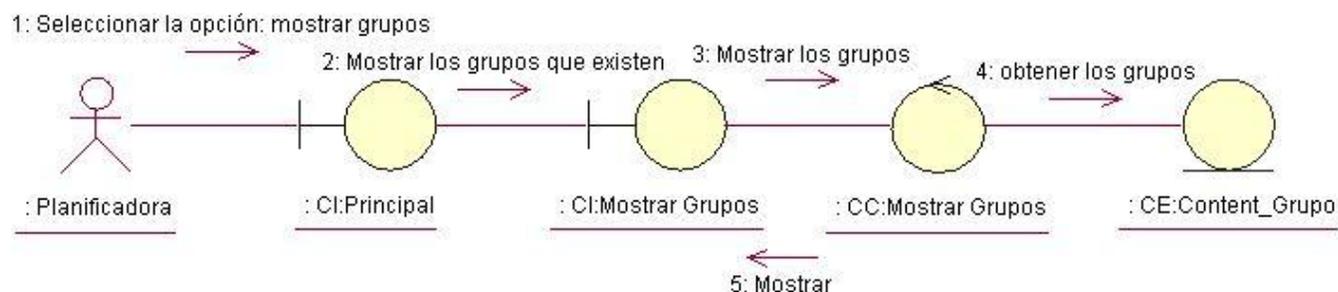
**Anexo 3.3 Diagrama de colaboración del caso de uso Gestionar Actividades.**



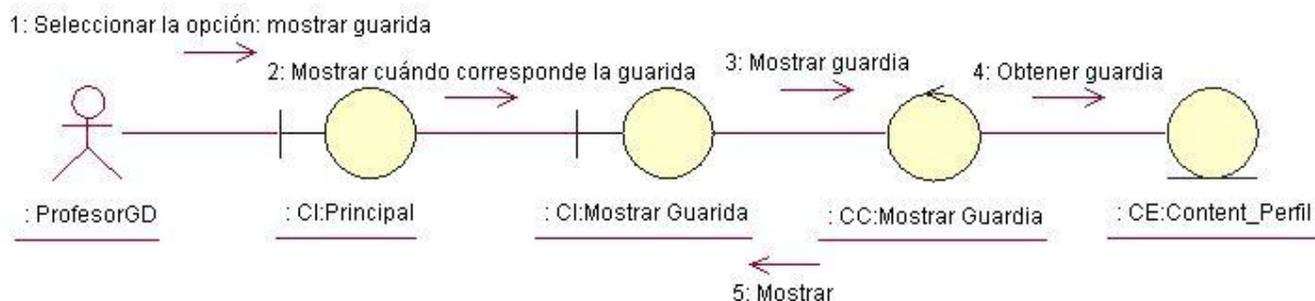
**Anexo 3.4 Diagrama de colaboración del caso de uso Mostrar Cuidado de Exámenes.**



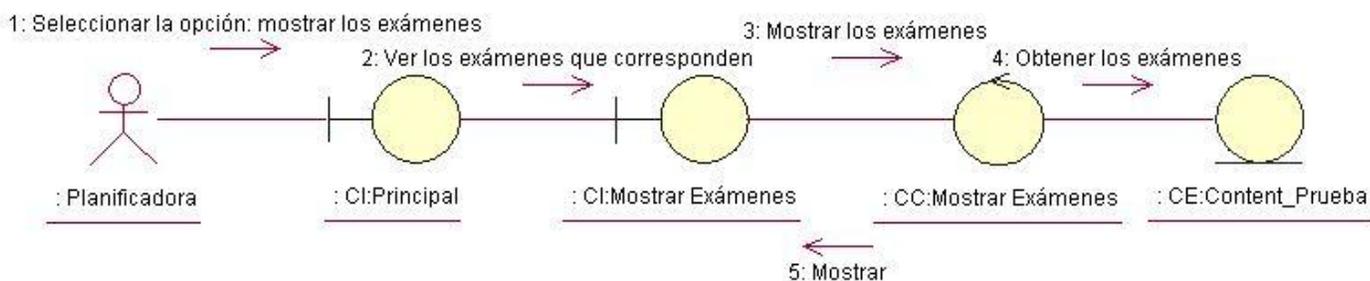
**Anexo 3.5 Diagrama de colaboración del caso de uso Mostrar Grupo.**

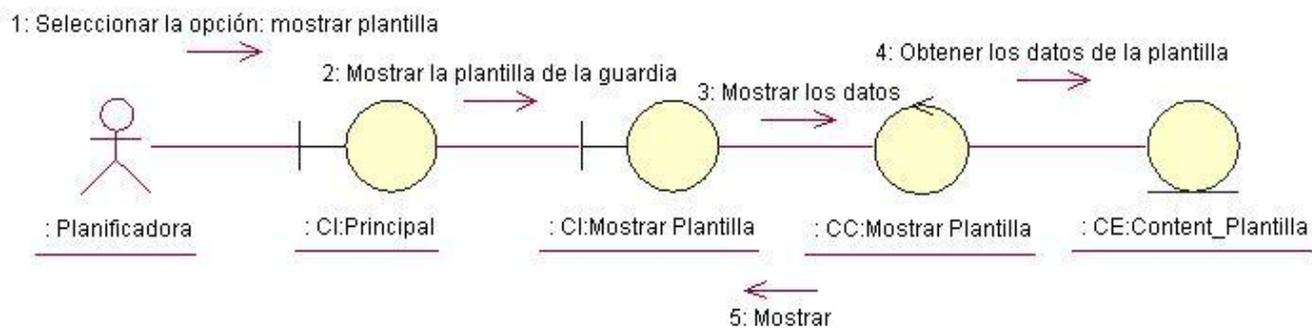
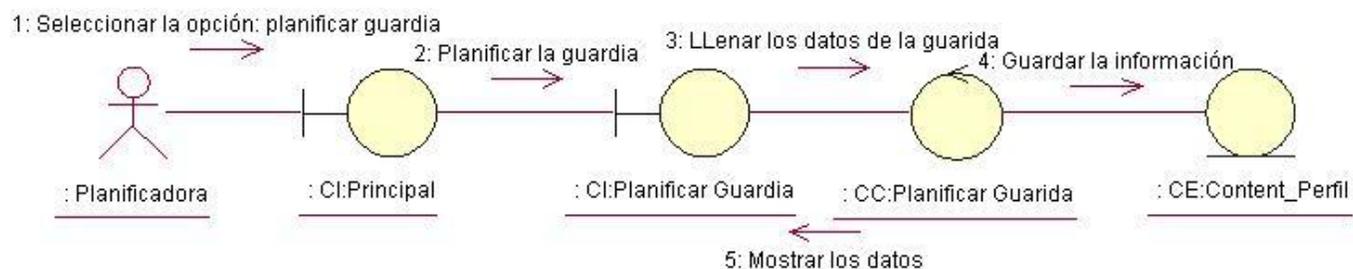


Anexo 3.6 Diagrama de colaboración del caso de uso Mostrar Guardia Docente.

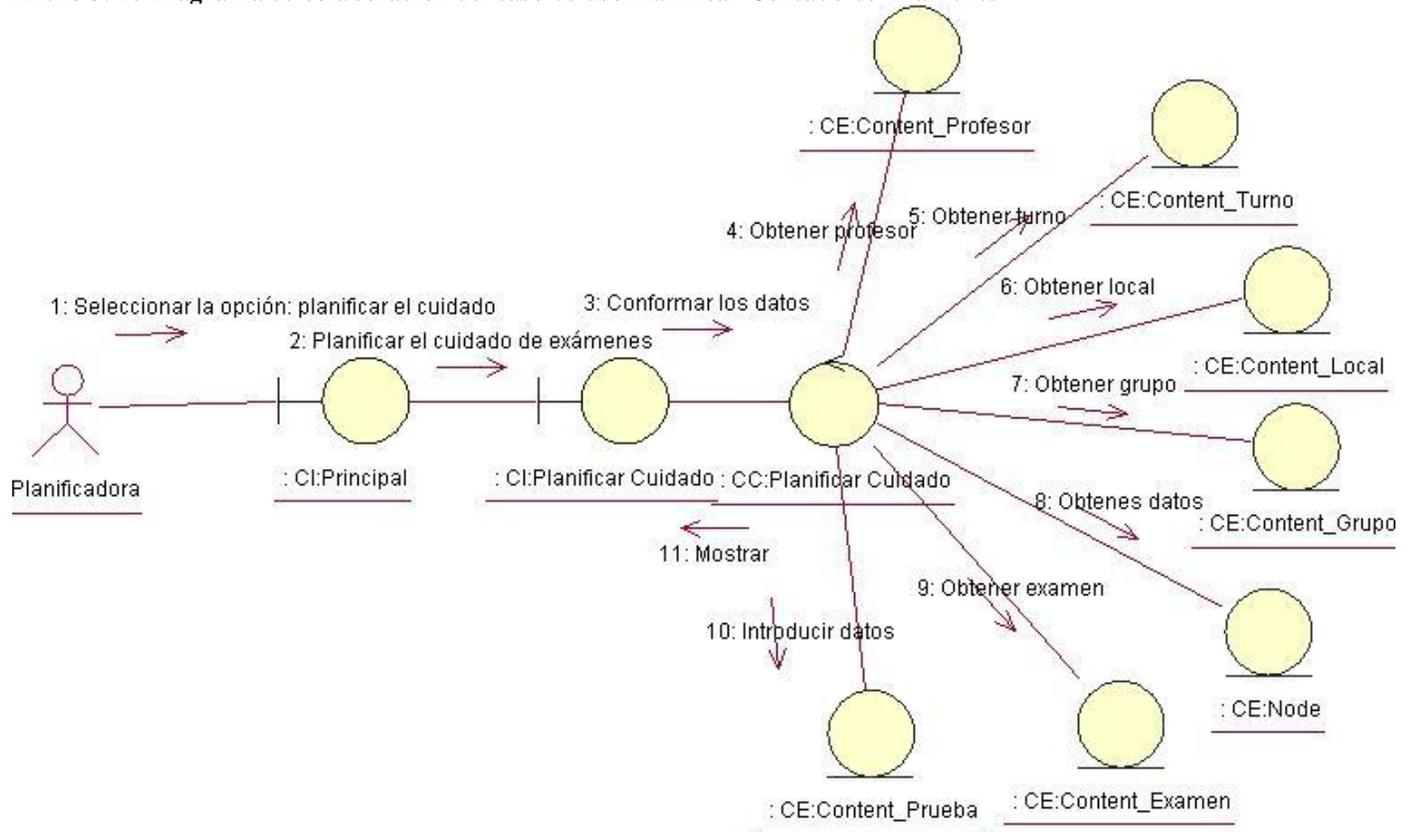


Anexo 3.7 Diagrama de colaboración del caso de uso Listar Exámenes.



**Anexo 3.8 Diagrama de colaboración del caso de uso Mostrar Plantilla.**

**Anexo 3.9 Diagrama de colaboración del caso de uso Planificar Guardia Docente.**


Anexo 3. 10 Diagrama de colaboración del caso de uso Planificar Cuidado de Exámenes



## Anexos 4 Descripción de las tablas del Modelo de datos.

### Anexo 4.1 Modelo de datos (Content\_Field\_Asignatura)

| <b>Nombre:</b> Content_Field _ Asignatura  |          |                                 |
|--|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de la Asignaturas. |          |                                 |
| Atributo   | Tipo     | Descripción                     |
| <b>Vid</b>   | Int(10)  | Identificador de la tabla.      |
| nid  | Int(10)  | Identificador de la tabla Node. |
| field_asignatura_value   | Int (11) | Identificador de la Asignatura  |

### Anexo 4.2 Modelo de datos (Content\_Field\_Grupo)

| <b>Nombre:</b> Content_Field _ Grupo   |          |                                 |
|--|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de los Grupos. |          |                                 |
| Atributo   | Tipo     | Descripción                     |
| <b>Vid</b>   | Int(10)  | Identificador de la tabla .     |
| nid  | Int(10)  | Identificador de la tabla Node. |
| field_grupo_value  | Int (11) | Identificador del grupo.        |

### Anexo 4.3 Modelo de datos (Content\_Field\_Local)

| <b>Nombre:</b> Content_Field _ Local  |          |                                 |
|---|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de los Locales. |          |                                 |
| Atributo  | Tipo     | Descripción                     |
| <b>Vid</b>  | Int(10)  | Identificador de la tabla.      |
| nid   | Int(10)  | Identificador de la tabla Nodo. |
| field_local_value   | Int (11) | Identificador del Local.        |

Anexo 4.4 Modelo de datos (Content\_Field\_Profesor)

| <b>Nombre:</b> Content_Field_Profesor  |          |                                 |
|--|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de los Profesores. |          |                                 |
| Atributo   | Tipo     | Descripción                     |
| <b>Vid</b>   | Int(10)  | Identificador de la tabla.      |
| nid  | Int(10)  | Identificador de la tabla Nodo. |
| field_profesor_value   | Int (11) | Identificador del Profesor.     |

Anexo 4.5 Modelo de datos (Content\_Field\_Semana)

| <b>Nombre:</b> Content_Field_Semana   |          |                                 |
|---|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de la semana. |          |                                 |
| Atributo  | Tipo     | Descripción                     |
| <b>Vid</b>  | Int(10)  | Identificador de la tabla.      |
| nid   | Int(10)  | Identificador de la tabla Nodo. |
| field_semana_value  | Int (11) | Identificador de la semana.     |

Anexo 4.6 Modelo de datos (Content\_Field\_Turno)

| <b>Nombre:</b> Content_Field_Turno   |          |                                 |
|--|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se guarda los identificadores de los Turnos. |          |                                 |
| Atributo   | Tipo     | Descripción                     |
| <b>Vid</b>   | Int(10)  | Identificador de la tabla.      |
| nid  | Int(10)  | Identificador de la tabla Nodo. |
| field_turno_value  | Int (11) | Identificador de los turnos.    |

Anexo 4.7 Modelo de datos (Content\_type\_dexamen)

| <b>Nombre:</b> Content_type_dexamen                                     |      |             |
|---|------|-------------|
| <b>Descripción:</b> En esta tabla se guarda la prueba se que planifico. |      |             |
| Atributo  | Tipo | Descripción |

|                  |          |  |
|------------------|----------|--|
| <b>Vid</b>       | Int(10)  | Identificador de la tabla.                                     |
| nid              | Int(10)  | Identificador de la tabla Nodo.                                |
| field_prueba_nid | Int (10) | Identificador de la tabla Node a la que pertenece la prueba.   |
| field_profe_nid  | Int (10) | Identificador de la tabla Node a la que pertenece el profesor. |

Anexo 4. 8 Modelo de datos (Content \_type\_perfil)

| <b>Nombre:</b> Content _type_perfil                                  |             |                                       |
|--|-------------|---------------------------------------|
| <b>Descripción:</b> En esta tabla se almacena los datos del perfil . |             |                                       |
| <b>Atributo</b>  | <b>Tipo</b> | <b>Descripción</b>                    |
| <b>Vid</b>   | Int(10)     | Identificador de la tabla.            |
| nid  | Int(10)     | Identificador de la tabla Nodo.       |
| field_solapin_value  | Lontext     | Numero del solapín.                   |
| field_nombre_value   | Longtext    | Nombre del usuario.                   |
| field_guardiad_value   | Int(11)     | Si es profesor de la guardia docente. |
| field_cprueba_value  | Int(11)     | Si es profesor del cuidado de examen. |
| field_session_value  | Longtext    | Sesión mañana y tarde.                |
| field_dsemana_value  | Longtext    | Días de la semana.                    |
| field_departamento_value   | Int(11)     | Departamento al que pertenece         |
| field_cantpruebas_value  | Int(11)     | Cantidad de pruebas planificadas.     |

Anexo 4. 9 Modelo de datos (Content \_type\_planilla)

| <b>Nombre:</b> Content _type_planilla  |             |                                 |
|--|-------------|---------------------------------|
| <b>Descripción:</b> En esta tabla se almacena el identificador de la Planilla. |             |                                 |
| <b>Atributo</b>  | <b>Tipo</b> | <b>Descripción</b>              |
| <b>Vid</b>   | Int(10)     | Identificador de la tabla.      |
| nid  | Int(10)     | Identificador de la tabla Nodo. |

|                      |          |                       |
|----------------------|----------|-----------------------|
| field_ausentes_value | Int (10) | Cantidad de ausentes. |
|----------------------|----------|-----------------------|

#### Anexo 4. 10 Modelo de datos (Content\_type\_prueba)

| <b>Nombre:</b> Content_type_prueba   |          |                                 |
|--|----------|---------------------------------|
| <b>Descripción:</b> En esta tabla se almacena la fecha en la que se realizó la prueba. |          |                                 |
| Atributo   | Tipo     | Descripción                     |
| Vid  | Int(10)  | Identificador de la tabla.      |
| nid  | Int(10)  | Identificador de la tabla Nodo. |
| field_fecha_value  | Int (10) | Fecha                           |

#### Anexo 4. 11 Modelo de datos (departamento)

| <b>Nombre:</b> Departamento   |             |                            |
|---|-------------|----------------------------|
| <b>Descripción :</b> En esta tabla se almacenan el id y el nombre del departamento. |             |                            |
| Atributo  | Tipo        | Descripción                |
| did   | Int(10)     | Identificador de la tabla. |
| nombre  | Varchar(40) | Nombre del departamento    |

#### Anexo 4. 12 Modelo de datos (locales)

| <b>Nombre:</b> Locales   |          |                             |
|--|----------|-----------------------------|
| <b>Descripción :</b> En esta tabla se guarda los locales que se están utilizando y en el docente que se encuentra. |          |                             |
| Atributo   | Tipo     | Descripción                 |
| lid  | Int(10)  | Identificador de la tabla.  |
| nombre   | Int(10)  | Nombre del local            |
| lugar  | Int (10) | Docente el que está ubicado |

#### Anexo 4. 13 Modelo de datos (node)

|   |  |  |
|---|--|--|
| <b>Nombre:</b> Node   |  |  |
| <b>Descripción :</b> Contiene todos los contenidos del sistema de gestión para la |  |  |

| planificacion docente en la facultad 5. |         |   |
|---|---------|---|
| Atributo                                | Tipo    | Descripción   |
| <b>nid</b>                              | Int(10) | Identificador de la tabla Node.   |
| vid                                     | Int(10) | Identificador de la tabla Vocabulary.   |
| type                                    | varchar | Tipo de contenido.  |
| title                                   | Varchar | Título del contenido.   |
| uid                                     | Int     | Identificador de la table User.   |
| status                                  | varchar | Indica el estado de publicación del contenido.  |
| created                                 | Int(10) | Fecha en la que se creó el contenido  |
| changed                                 | Int(10) | Fecha en la que se modifico el contenido.   |
| comment                                 | Int(10) | Si permite o no el envío de comentarios referente a un contenido.   |
| promote                                 | Int(10) | Indica el grado de promoción de un contenido, en dependencia del valor que tenga el contenido, aparece o no en la página principal. |
| moderate                                | Int(10) | Indica si el contenido se encuentra en la cola de moderación.   |
| sticky                                  | Int(10) | Indica si el contenido tiene prioridad sobre otros contenidos.  |

Anexo 4. 14 Modelo de datos (planilla)

| <b>Nombre:</b> Planilla  |         |   |
|--|---------|---|
| <b>Descripción:</b> En esta tabla se almacenan todos los datos correspondientes a la planilla que llena el profesor de la guardia docente. |         |   |
| Atributo   | Tipo    | Descripción   |
| <u>vid</u>   | Int(10) | Identificador de la tabla.                          |
| nid  | Int(10) | Identificador de la tabla Nodo.                     |
| semana   | Varchar | Semana Par o Impar.                                 |
| local  | Varchar | Local en el que se está dando clase.                |
| profesor   | Varchar | Profesor que está impartiendo clase en ese momento. |
| asignatura   | Varchar | Asignatura que se está impartiendo en ese local.    |
| grupo  | Varchar | Grupo que está en ese local.                        |
| turno  | Int     | Turno en que paso el profesor .                     |
| ausentes   | Text    | Cantidad de ausentes que hay en el grupo.           |

Anexo 4. 15 Modelo de datos (roles)

| <b>Nombre:</b> Roles   |         |                            |
|--|---------|----------------------------|
| <b>Descripción :</b> En esta tabla se almacenan el identificador y el nombre del rol |         |                            |
| Atributo   | Tipo    | Descripción                |
| <u>rid</u>   | Int(10) | Identificador de la tabla. |
| nombre   | Int(10) | Nombre del Rol             |

Anexo 4. 16 Modelo de datos (user)

|   |  |  |
|---|--|--|
| <b>Nombre:</b> user   |  |  |
| <b>Descripción :</b> Esta tabla contiene información de los usuarios. |  |  |

| Atributo   | Tipo    | Descripción  |
|------------|---------|--|
| <u>uid</u> | Int(10) | Identificador de la tabla users  |
| name       | varchar | Nombre de usuario  |
| pass       | varchar | Contraseña de usuario  |
| mail       | varchar | Correo electrónico del usuario   |
| mode       | Int     | Es utilizado para llevar el control de los usuarios, es como el campo status |
| sort       | Int     | Rol del usuario  |
| threshold  | Int     | Es un contador   |
| theme      | Varchar | Tema especificado para el usuario  |
| signature  | Varchar | Para guardar información de comentarios hechos por el usuario                |
| created    | Int     | Fecha en que se creó el usuario  |
| access     | Int     | Fecha del último acceso.   |
| login      | Int     | Fecha y hora de la última vez que se logueo el usuario.                      |
| status     | Int     | Si esta logueado o no.   |
| timezone   | Varchar | Zona horaria que emplea el usuario.  |

---

|          |         |  |
|----------|---------|--|
| language | Varchar | Lenguaje en que se le presenta la interfaz al usuario. |
| picture  | Varchar | Foto.  |

**Anexo 4.17 Modelo de datos (user Roles)**

| <b>Nombre:</b> user Roles  |             |                                 |
|--|-------------|---------------------------------|
| <b>Descripción :</b> Esta tabla se crea de la unión de mucho a mucho entre las tablas User y Roles |             |                                 |
| <b>Atributo</b>  | <b>Tipo</b> | <b>Descripción</b>              |
| <b><u>uid</u></b>  | Int(10)     | Identificador de la tabla user. |
| <b><u>rid</u></b>  | Int(10)     | Identificador la tabla roles    |

## ÍNDICE DE FIGURAS

|   |     |
|---|-----|
| FIGURA 1: MODELO DE CASOS DE USO DEL NEGOCIO .....  | 39  |
| FIGURA 2: DIAGRAMA DE CASOS DE USO DEL SISTEMA.....   | 47  |
| FIGURA 3 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CASO DE USO PLANIFICAR GUARDIA DOCENTE.....      | 53  |
| FIGURA 4: DIAGRAMA DE CLASES DEL DISEÑO DE DRUPAL.....  | 55  |
| FIGURA 5: ESTRUCTURA INTERNA DEL MÓDULO SGPD .....  | 56  |
| FIGURA 6: DIAGRAMA DE CLASES PERSISTENTES.....  | 58  |
| FIGURA 7: MODELO DE DATOS .....   | 59  |
| FIGURA 8: DIAGRAMA DE DESPLIEGUE.....   | 63  |
| FIGURA 9: DIAGRAMA DE COMPONENTES .....   | 64  |
|   |     |
| ANEXO 2. 1 DIAGRAMA DE CLASE DE ANÁLISIS DEL CASO DE USO AUTENTICAR USUARIO .....             | 92  |
| ANEXO 2. 2 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO CREAR PLANTILLA EN LA GUARDIA. .... | 92  |
| ANEXO 2. 3 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO GESTIONAR ACTIVIDADES. ....         | 92  |
| ANEXO 2. 4 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO GESTIONAR ROLES. ....               | 92  |
| ANEXO 2. 5 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO MOSTRAR CUIDADO DE EXÁMENES.....    | 93  |
| ANEXO 2. 6 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO MOSTRAR GUARDIA DOCENTE. ....       | 93  |
| ANEXO 2. 7 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO PLANIFICAR CUIDADO DE EXÁMENES..... | 94  |
| ANEXO 2. 8 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO LISTAR ASIGNATURAS .....            | 94  |
| ANEXO 2. 9 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO MOSTRAR PLANTILLA.....              | 95  |
| ANEXO 2. 10 DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO MOSTRAR GRUPOS.....                | 95  |
|   |     |
| ANEXO 3. 1 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO AUTENTICAR USUARIO.....                   | 96  |
| ANEXO 3. 2 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO CREAR PLANTILLA EN LA GUARDIA. ....       | 96  |
| ANEXO 3. 3 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO GESTIONAR ACTIVIDADES. ....               | 97  |
| ANEXO 3. 4 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO MOSTRAR CUIDADO DE EXÁMENES. ....         | 97  |
| ANEXO 3. 5 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO MOSTRAR GRUPO. ....                       | 97  |
| ANEXO 3. 6 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO MOSTRAR GUARDIA DOCENTE. ....             | 98  |
| ANEXO 3. 7 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO LISTAR EXÁMENES. ....                     | 98  |
| ANEXO 3. 8 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO MOSTRAR PLANTILLA.....                    | 99  |
| ANEXO 3. 9 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO PLANIFICAR GUARDIA DOCENTE. ....          | 99  |
| ANEXO 3. 10 DIAGRAMA DE COLABORACIÓN DEL CASO DE USO PLANIFICAR CUIDADO DE EXÁMENES.....      | 100 |

---

## ÍNDICE DE TABLAS

|  |     |
|--|-----|
| TABLA 1: ACTORES DEL NEGOCIO.....  | 38  |
| TABLA 2: TRABAJADORES DEL NEGOCIO.....   | 39  |
| TABLA 3: DESCRIPCIÓN DEL CASO DE USO PLANIFICAR GUARDIA DOCENTE.....                     | 40  |
| TABLA 4: DESCRIPCIÓN DEL CASO DE USO PLANIFICAR CUIDADO DE EXÁMENES.....                 | 41  |
| TABLA 5: ACTORES DEL SISTEMA.....  | 46  |
| TABLA 6: DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA PLANIFICAR CUIDADO DE EXÁMENES.....     | 48  |
| TABLA 7: DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA PLANIFICAR GUARDIA DOCENTE.....         | 50  |
| TABLA 8: PAQUETES DE DRUPAL.....   | 56  |
| TABLA 9: MODELO DE DATOS (ASIGNATURA).....   | 59  |
| TABLA 10: DESCRIPCIÓN DEL DIAGRAMA DE COMPONENTES.....                                   | 64  |
| TABLA 11: DESCRIPCIÓN DEL CASO DE PRUEBA PARA EL CU AUTENTICAR USUARIO.....              | 67  |
| TABLA 12: DESCRIPCIÓN DEL CASOS DE PRUEBA PARA EL CU PLANIFICAR GUARDIA DOCENTE.....     | 69  |
| TABLA 13: DESCRIPCIÓN DEL CASOS DE PRUEBA PARA EL CU PLANIFICAR CUIDADO DE EXÁMENES..... | 70  |
| TABLA 14: DESCRIPCIÓN DEL CASOS DE PRUEBA PARA EL CU GESTIONAR GRUPOS.....               | 72  |
|  |     |
| ANEXO 1.1 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA AUTENTICAR USUARIO.....                | 84  |
| ANEXO 1.2 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA GESTIONAR ACTIVIDADES.....             | 85  |
| ANEXO 1.3 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA CREAR PLANTILLA EN LA GUARDIA.....     | 86  |
| ANEXO 1.4 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA MOSTRAR GUARDIA DOCENTE.....           | 87  |
| ANEXO 1.5 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA MOSTRAR CUIDADO DE EXÁMENES.....       | 87  |
| ANEXO 1.6 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA GESTIONAR GRUPOS.....                  | 88  |
| ANEXO 1.7 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA MOSTRAR PLANTILLA DE GUARDIA.....      | 89  |
| ANEXO 1.8 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA LISTADO DE EXÁMENES.....               | 90  |
| ANEXO 1.9 DESCRIPCIÓN DEL CASO DE USO DEL SISTEMA MOSTRAR GRUPOS.....                    | 91  |
|  |     |
| ANEXO 4.1 MODELO DE DATOS (CONTENT_FIELD_ASIGNATURA).....                                | 101 |
| ANEXO 4.2 MODELO DE DATOS (CONTENT_FIELD_GRUPO).....                                     | 101 |
| ANEXO 4.3 MODELO DE DATOS (CONTENT_FIELD_LOCAL).....                                     | 101 |
| ANEXO 4.4 MODELO DE DATOS (CONTENT_FIELD_PROFESOR).....                                  | 102 |
| ANEXO 4.5 MODELO DE DATOS (CONTENT_FIELD_SEMANA).....                                    | 102 |
| ANEXO 4.6 MODELO DE DATOS (CONTENT_FIELD_TURNO).....                                     | 102 |
| ANEXO 4.7 MODELO DE DATOS (CONTENT_TYPE_DEXAMEN).....                                    | 102 |
| ANEXO 4.8 MODELO DE DATOS (CONTENT_TYPE_PERFIL).....                                     | 103 |
| ANEXO 4.9 MODELO DE DATOS (CONTENT_TYPE_PLANILLA).....                                   | 103 |
| ANEXO 4.10 MODELO DE DATOS (CONTENT_TYPE_PRUEBA).....                                    | 104 |
| ANEXO 4.11 MODELO DE DATOS (DEPARTAMENTO).....   | 104 |
| ANEXO 4.12 MODELO DE DATOS (LOCALES).....  | 104 |
| ANEXO 4.13 MODELO DE DATOS (NODE).....   | 104 |
| ANEXO 4.14 MODELO DE DATOS (PLANILLA).....   | 106 |
| ANEXO 4.15 MODELO DE DATOS (ROLES).....  | 106 |
| ANEXO 4.16 MODELO DE DATOS (USER).....   | 106 |
| ANEXO 4.17 MODELO DE DATOS (USER ROLES).....   | 108 |