

**Universidad de las Ciencias Informáticas**  
**Facultad 5**



**Sistema para el control del funcionamiento del  
Movimiento de Estudiantes Técnicos de  
Laboratorios de la UCI.**

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

Autor: *Raimé Bercourt Quesada*

Tutor(es): *M.Sc. Lidioy Alonso Hernández*

*Ing. Dayren Martínez Sousa*

29 de Junio del 2010

## I. DECLARACIÓN DE AUTORÍA

Yo Raimé Bercourt Quesada declaro ser autor de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autor:

Raimé Bercourt Quesada

\_\_\_\_\_

Firma del Autor

Tutores:

MSc. Lidiexy Alonso Hernández

\_\_\_\_\_

Firma del Tutor

Ing. Dayren Martínez Sousa

\_\_\_\_\_

Firma del Tutor

## II. DATOS DE CONTACTO

MSc. Lidiexy Alonso Hernández ([lidiexy@uci.cu](mailto:lidiexy@uci.cu))

Licenciado en Ciencias de la Computación de la Universidad Central “Marta Abreu” de las Villas.  
Profesor Instructor de la Facultad 5, con 6 años de graduado y 8 años de experiencia en el tema.

Ing. Dayren Martínez Sousa ([dsousa@uci.cu](mailto:dsousa@uci.cu))

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas.  
Profesor Adiestrado de la Facultad 5, con 2 años de graduada y 4 años de experiencia en el tema.

## AGRADECIMIENTOS

A mi familia en primer lugar, en especial a mi mamá, a mi padrastro, a mi abuela y a mis tíos, que sin su apoyo, yo no hubiera estado donde estoy en estos momentos. Gracias por confiar en mí y apoyarme siempre.

A mi hermano Adrián García, quién me ha ayudado mucho en estos últimos años de la carrera, y que a pesar de nuestras diferencias, lo quiero mucho, gracias por estar siempre ahí y por soportarme, nunca te olvidaré.

A mis tutores por saberme guiar, enseñar, soportar, por la paciencia que han tenido y por el apoyo que me han dado durante toda la carrera y en especial, durante el desarrollo de este trabajo.

A mis amigos, en especial a Julio, a Adrian Pérez, a Danny y a todos los que de una forma u otra me apoyaron y me ayudaron durante toda la carrera y durante este último año, gracias por ser como son, siempre los recordaré.

A la Facultad y la Universidad de las Ciencias Informáticas por ayudarme a ser quien soy y por todo lo que me han enseñado.

A los compañeros de la Dirección de Laboratorios, en especial a Oneyvis, Dunay, Zobeida y Dafne, por su apoyo incondicional durante todos estos años, por su ayuda en los momentos difíciles y por todo lo que me enseñaron. Gracias, siempre las tendré presente.

A la FEU por ser como una escuela durante toda la carrera y por todas las cosas buenas y malas que pasamos juntos.

A todos los que de una forma u otra me ayudaron durante todos estos años.

## Dedicatoria

A mi mamá, mi padrastro y mi abuela, por vivir sólo para mí y ayudarme en todo lo que he necesitado.

A la memoria de mi tío Nico, que siempre me quiso y me apoyó, siempre te voy a recordar y siempre voy a tenerte en el corazón.

A mi hermano Adrián García, por tu apoyo y ayuda durante estos años.

## **RESUMEN**

La Universidad de las Ciencias Informáticas está compuesta entre otras áreas, por los laboratorios, los cuales están divididos en docentes y productivos. Los laboratorios docentes, son atendidos por los técnicos de la Dirección de Laboratorios apoyados en la noche por el Movimiento de Estudiantes Técnicos de Laboratorios (METLab).

El Movimiento a nivel de Universidad y la Dirección de Laboratorios manejan gran cúmulo de información relacionada con el mismo, tanto en las facultades como a nivel UCI, toda esa información se lleva manualmente en papel, lo cual es un gran gasto de recursos ya que esa información es manejada por los jefes de turnos y los jefes del Movimiento en cada una de las facultades y de la Universidad, a veces perdiéndose estos documentos y haciéndose difícil tener un control de toda esta información, siendo engorroso obtener un dato determinado por no estar el documento o no tenerse acceso al mismo. En el Movimiento se realiza un proceso de evaluación a los estudiantes que lo integran, el cual no es eficiente, ya que no existen indicadores por los cuales evaluar a los estudiantes por igual en cada guardia o en cada facultad.

El objetivo de la investigación se centra en el diseño e implementación de un sistema de gestión que permita controlar el funcionamiento del METLab y el proceso de evaluación de los estudiantes del mismo. De esta manera, se analizan diferentes sistemas de gestión de información laboral de los recursos humanos, concluyendo que no existe una aplicación que de solución a este problema. También se analiza el proceso de evaluación de los estudiantes, proponiendo un sistema de indicadores para el mismo. Por tanto, se describe todo el estudio de las tendencias actuales de la web y los CMS, escogiendo las tecnologías apropiadas para la realización del sistema, el cual se documenta debidamente.

## **PALABRAS CLAVE**

Sistema de Gestión de Información, CMS, Movimiento de Estudiantes Técnicos de Laboratorios.

## Índice de Contenidos

<b>RESUMEN</b> .....	3
<b>INTRODUCCIÓN</b> .....	6
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	9
<b>1. Movimiento de Estudiantes Técnicos de Laboratorios de la UCI.</b> .....	9
1.1. Antecedentes. ....	9
1.2. Organización y control de la información generada. ....	10
1.2.1. ¿Qué se entiende por organización? .....	10
1.2.2. ¿Qué se entiende por control? .....	10
1.3. Información generada en los procesos del METLab. ....	10
1.4. Evaluación de los estudiantes. ....	11
1.4.1. ¿Qué se entiende por evaluación? .....	11
1.4.2. Propuesta de un sistema de indicadores para la evaluación. ....	11
<b>2. Sistemas de Gestión de Información.</b> .....	12
2.1. Sistemas de Gestión de información Laboral de los Recursos Humanos. ....	12
<b>CAPÍTULO 2: TECNOLOGÍAS DE DESARROLLO.</b> .....	14
<b>1. Metodologías de desarrollo del software</b> .....	14
1.1. ¿Qué es una metodología de desarrollo del software? .....	14
<b>2. Metodología de Desarrollo de Software RUP</b> .....	14
<b>3. Lenguaje para el modelado: UML.</b> .....	15
<b>4. Diferentes Lenguajes de programación.</b> .....	16
4.1. PHP.....	16
4.2. JavaScript.....	16
<b>5. Sistemas de Gestión de Contenido.</b> .....	17
5.1. ¿Qué es un CMS?.....	17
5.2. Necesidad de un CMS.....	18
5.3. Ejemplos de CMS.....	19
5.3.1. Joomla.....	19
5.3.2. WordPress.....	19
5.3.3. Drupal.....	20
5.4. ¿Por qué usar el CMS Drupal? .....	20
<b>6. Arquitectura cliente – servidor.</b> .....	21
6.1. Características del modelo cliente – servidor. ....	21
<b>7. Servidor Web Apache.</b> .....	21
<b>8. Sistemas Gestores de Base de Datos.</b> .....	22
8.1. MySQL .....	22
8.2. PostgreSQL.....	22
<b>9. ¿Por qué usar PostgreSQL?</b> .....	24
<b>10. ¿Por qué usar la trilogía Apache + PHP + PostgreSQL?</b> .....	24
<b>11. Herramientas utilizadas para la elaboración del sistema.</b> .....	24
11.1. Visual Paradigm para UML 6.4. ....	24
11.2. Embarcadero ERStudio 8.0. ....	25

11.3 PHPDesigner 7. ....	26
<b>CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA</b> .....	28
<b>1. Modelo del negocio.</b> .....	28
1.1. Descripción de los procesos del negocio y las mejoras propuestas. ....	28
1.2. Actores y trabajadores del negocio. ....	29
1.3. Diagrama de los Casos de Usos del Negocio. ....	30
1.4. Listado de casos de uso del negocio expandidos. ....	32
1.5. Diagrama de Actividades de los Casos de Usos del Negocio. ....	40
1.6. Diagrama de clases del modelo de objetos. ....	41
<b>2. Especificación de los requisitos de software.</b> .....	42
2.1. Requisitos Funcionales ....	42
2.2 Requisitos no funcionales ....	43
<b>3. Definición de los casos de uso del sistema.</b> .....	44
3.1. Definición de los actores del sistema.....	44
3.2. Diagrama de casos de uso del sistema.....	45
3.3. Casos de uso expandidos.....	47
<b>CAPÍTULO 4: CONSTRUCCIÓN DEL SISTEMA</b> .....	70
<b>1. Definición del modelo de análisis.</b> .....	70
1.1. Modelo de clases de análisis. ....	70
<b>2. Diseño.</b> .....	70
2.1. Diagramas de colaboración. ....	70
<b>3. Diseño de la Base de Datos</b> .....	71
3.1. Diagrama Entidad Relación de la Base de Datos. ....	71
<b>4. Implementación.</b> .....	72
4.1. Diagrama de despliegue. ....	72
<b>CONCLUSIONES</b> .....	73
<b>RECOMENDACIONES</b> .....	; <b>ERROR! MARCADOR NO DEFINIDO.</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	73
<b>BIBLIOGRAFÍA</b> .....	; <b>ERROR! MARCADOR NO DEFINIDO.</b>
<b>GLOSARIO</b> .....	; <b>ERROR! MARCADOR NO DEFINIDO.</b>

## INTRODUCCIÓN

Con el origen y desarrollo de la red de redes, o Internet como se le suele llamar, se ha posibilitado la creación de aplicaciones Web que resuelvan las necesidades de los usuarios así como las diferentes problemáticas de las empresas, permitiéndoles automatizar sus procesos e informaciones, posibilitando en algunos casos que la misma esté al alcance de un clic.

Para un mejor desarrollo de estas aplicaciones Web sin necesidad de tener mucho conocimiento de la informática, se crean los Sistemas de Gestión de Contenido (CMS por sus siglas en inglés), las cuales permiten al usuario crear un Portal Web en pocos pasos. Ejemplo de CMS son: Joomla, Plone, Mitra, WordPress, Drupal y entre otros, cada uno con sus diferentes características. Estos CMS se utilizaron para la realización de Portales Web por muy complejos que fueran éstos y para el control de la información de diferentes empresas. Siendo así, que hoy en día muchos sitios en Internet están diseñados tomando como base estos sistemas.

En Cuba la utilización de la Internet ha mejorado en los últimos años, aumentando el acceso de los cubanos a la red de redes y aumentando el conocimiento y desarrollo en las diferentes esferas de la Informática. Con la inserción de los CMS en Cuba, algunos sitios han sido realizados sobre estos sistemas permitiéndoles una mejor interacción con los usuarios siendo así una ventaja tanto para los desarrolladores como para los usuarios.

En la Universidad de las Ciencias Informáticas (UCI) la Internet se usa diariamente, principalmente en la realización de Portales Web, siendo éstos diseñados sobre CMS, usándose en muchos casos para la publicación de noticias e informaciones de interés para la institución o algún área específica y en menos casos, para el control de información generada a partir de los procesos que tienen lugar en la Universidad. Ejemplo de esto es en la Dirección de Laboratorios donde solo existen Portales Web que permiten el control de la información relacionada con la reservación del tiempo de máquina y el control de los problemas de hardware o software que existen en los laboratorios, así como la publicación de noticias relacionadas con la Informática.

El Movimiento de Estudiantes Técnicos de Laboratorios(METLab) a nivel de Universidad y la Dirección de Laboratorios manejan gran cúmulo de información relacionada con el mismo, tanto en las facultades como a nivel UCI, dentro de la cual se encuentra el control de todos los estudiantes que pertenecen al Movimiento en cada facultad, el control de los datos de todos los estudiantes, los resultados de las inspecciones mensuales avisadas y sorpresivas a los laboratorios de cada facultad, el sistema de guardia mensual de cada laboratorio y facultad, la emulación por mes, del semestre y del curso en general entre las facultades, el control de asistencia de los estudiantes a las guardias, así como la entrega de los laboratorios por parte de los técnicos a los estudiantes y viceversa.

Actualmente toda esa información se lleva manualmente en papel, lo cual es un gran gasto de recursos al ser necesario imprimir modelos, partes, hojas de firma, ya que esa información es manejada por los jefes de turnos y los jefes del Movimiento en cada una de las facultades y de la Universidad, a veces perdiéndose estos documentos y haciéndose difícil tener un control de toda esta información de los estudiantes y de las facultades en el mes o en el semestre, siendo engorroso obtener un dato determinado por no estar el documento o no tenerse acceso al mismo.

En el Movimiento se realiza un proceso de evaluación a los estudiantes que lo integran, dentro del cual se les evalúa cuando están de guardia. Esta evaluación se realiza con periodicidad mensual, semestral y una final tras concluido el curso, teniendo en cuenta siempre las evaluaciones obtenidas en el periodo a evaluar, es decir, si se va a emitir una evaluación mensual se tienen en cuenta las evaluaciones obtenidas

en ese mes durante los días que estuvo de guardia ese estudiante. Actualmente este proceso no es eficiente, ya que no existe una medida estándar por la cual evaluar a los estudiantes por igual en cada guardia o en cada facultad, y como las evaluaciones quedan plasmadas en papel al igual que el resto de la información a la hora de emitir una evaluación del semestre o del curso, no se puede dar una evaluación adecuada al estudiante por no haber constancia de sus evaluaciones durante el curso por pérdida de esta información o porque no es accesible.

A partir de la situación problemática presentada anteriormente, se plantea el siguiente **problema científico** *¿Cómo controlar el funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios de la UCI, que permita gestionar el proceso evaluativo de sus miembros?*

Atendiendo a las necesidades de la dirección del METLab a nivel UCI y de la Dirección de Laboratorios, el **objeto de estudio** del presente trabajo son los sistemas de gestión de información web y el **campo de acción** se centra en los sistemas web para la gestión de la información de recursos humanos.

Para darle solución a este problema, se plantea como **objetivo general** *diseñar e implementar un sistema de gestión que permita controlar el funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios y evaluar el trabajo de cada uno de sus miembros.*

Como **tareas de investigación** se tienen:

- Caracterizar el funcionamiento del METLab en la UCI.
- Describir las fases del proceso de evaluación.
- Identificar los indicadores que deben regir el proceso de evaluación de los Estudiantes Técnicos de Laboratorios.
- Describir las dificultades que presentan los diferentes roles para realizar el control de la información generada en METLab.
- Identificar la Metodología de desarrollo de Software, el Lenguaje de Programación, CMS y Sistema de Gestión de Bases de Datos (SGBD) idónea en la implementación de la aplicación web, de acuerdo a las ventajas y desventajas que poseen y al interés del METLab y de la Dirección de Laboratorios.
- Caracterizar las aplicaciones existentes que son análogas a la solución propuesta para determinar en qué medida dan solución al problema propuesto.
- Determinar qué información es necesaria gestionar por parte de la dirección del METLab y la Dirección de Laboratorios.
- Fundamentar las tendencias actuales, tecnologías y conceptos más importantes relacionados con la implementación y uso de los sistemas de gestión de información.
- Seleccionar el CMS adecuado, que cumpla las condiciones necesarias para darle solución al problema.

Entre los **Métodos de investigación** ha utilizar se encuentran:

- Teóricos:
  - Histórico-lógico: Se analizará cómo han ido evolucionando los sistemas de gestión de información en aplicaciones WEB, así como el desarrollo de las herramientas más usadas en la realización de aplicaciones WEB.

- Analítico – sintético, que permitirá realizar un análisis del proceso de evaluación para sintetizar los indicadores que se usarán en el mismo. Otro método teórico fundamental será la Modelación, que posibilitará representar, a partir de un modelo, cómo interactúan todos los roles que intervienen en el proceso de evaluación, así como el funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios en general.
- Empíricos:
  - Observación: Se realizará un seguimiento de toda la información que se recoge mensualmente en las facultades, así como el acceso y la búsqueda efectiva de la misma.
  - Entrevistas: Se realizarán entrevistas a los Jefes del METLab en las facultades así como a los Especialistas de Áreas por parte de la Dirección de Laboratorios para determinar la importancia y necesidad que existe de tener la información controlada, centralizada y digitalizada.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1. Movimiento de Estudiantes Técnicos de Laboratorios de la UCI.**

#### **1.1. Antecedentes.**

La Universidad de las Ciencias Informáticas está compuesta entre otras áreas, por los laboratorios, los cuales están divididos en docentes y productivos. Los laboratorios productivos son atendidos por los jefes de proyectos, los cuales son responsables de los mismos durante su servicio. En cuanto a los laboratorios docentes, son controlados por los técnicos de la Dirección de Laboratorios. En los primeros cursos de la Universidad, en los laboratorios existían muchos problemas, ya que los técnicos que habían en ese momento no daban abasto para cubrir todos los laboratorios docentes, lo cual provocaba ineficiencia en el servicio a los estudiantes. También se producían robos en los laboratorios, se encontraban keyloggers en las máquinas y no todas estaban en óptimas condiciones para ser usadas por los estudiantes. Otro problema era la situación económica en ese momento, ya que según cálculos realizados en un futuro cuando estuvieran los 10 docentes funcionando y trabajado en cada uno alrededor de 20 técnicos, sería una gran cantidad de dinero invertido solo en el pago de los técnicos, manteniéndose o aumentando las deficiencias antes mencionadas.

Con el objetivo de apoyar a los técnicos en los laboratorios, atenuar todas estas deficiencias y ayudar a la situación económica de la Universidad se decide crear un grupo de estudiantes que apoyaran a los técnicos en el trabajo de los laboratorios por la noche, a los cuales inicialmente se les dio el nombre de Movimiento de Alumnos Ayudantes de Laboratorios. El mismo comenzó primeramente en la Facultad 5, la cual sirvió de prueba piloto para determinar si era factible o no trabajar con los estudiantes como técnicos de laboratorios en la noche. Luego de 6 meses de trabajo y gracias al esfuerzo de muchos estudiantes, profesores y a la Dirección de Laboratorio, se logró demostrar que los mismos sí podían apoyar a los técnicos en los laboratorios por la noche, siendo así, en el siguiente curso se decidió extender esa experiencia a todas la facultades de la Universidad.

A principio el Movimiento no tenía una estructura centralizada, no contaba con uniformidad en todas las facultades en cuanto a los turnos de trabajo, la cantidad de estudiantes por laboratorios, los sistemas de evaluación y control, la estructuración de las guardias y no se regían por un mismo reglamento. Todo esto estaba dado porque el Movimiento no contaba con una dirección a nivel de Universidad que controlara el trabajo en las facultades, diera respuesta a inquietudes de mayor índole y realizara evaluaciones e inspecciones al trabajo en el Movimiento y en los laboratorios.

Producto de esto, la Dirección de Laboratorios en conjunto con la FEU de la Universidad, idearon una estrategia para crear una dirección del Movimiento a nivel central, la cual fue aprobada en todas las facultades. Entre los elementos que se plantearon en dicha estrategia se encontraban:

- La realización de un reglamento central por el cual se regirían todas las facultades, en el cual se planteaba las características que debía tener un estudiante para formar parte del Movimiento, cómo serían las guardias en los laboratorios y las responsabilidades que debía tener un estudiante del Movimiento cuando estaba como responsable de un laboratorio.
- La estructura del Movimiento tanto a nivel central como en las facultades, es decir, de cómo estaría estructurada la dirección del Movimiento.
- Un plan de trabajo del Movimiento en el cual estarían las principales actividades que debían realizar todas las facultades y la dirección del Movimiento central en el curso, entre las cuales se encontraba la realización de activos tanto a nivel de facultad como a nivel de Universidad, las inspecciones a las facultades para controlar el trabajo en los laboratorios, entre otras.

- Un sistema de evaluación para los estudiantes del Movimiento y para el Jefe del Movimiento a nivel de facultad para lograr una mayor eficiencia en el trabajo del Movimiento.
- Un sistema de inspecciones avisadas y sorpresivas a los laboratorios en las facultades para tener un mayor control de cómo se estaba trabajando en las facultades y velar por el cumplimiento del reglamento de los laboratorios y el del Movimiento, tanto por los estudiantes que estaba de guardia en el laboratorio, como por los estudiantes que estaban haciendo uso del mismo en ese momento.
- Un sistema de emulación entre las facultades, la cual se realizaría mediante las inspecciones a las facultades y las evaluaciones de los estudiantes, logrando así que las facultades trabajaran con una mayor calidad.
- Se decidió cambiarle el nombre al Movimiento y llamarle Movimiento de Estudiantes Técnicos de Laboratorios (METLab).

## **1.2. Organización y control de la información generada.**

### **1.2.1. ¿Qué se entiende por organización?**

Organizar es agrupar y ordenar las actividades necesarias para alcanzar los fines establecidos creando unidades administrativas, asignando en su caso funciones, autoridad, responsabilidad y jerarquía y estableciendo las relaciones que en dicha unidad deben existir. (González, 2009)

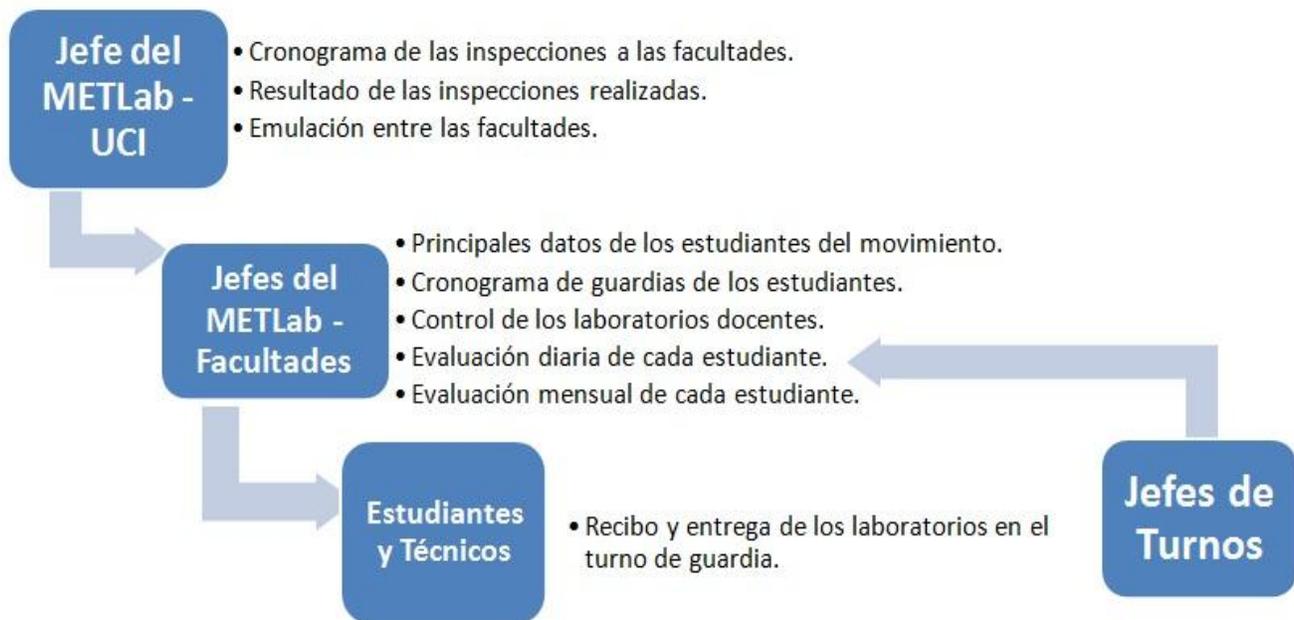
### **1.2.2. ¿Qué se entiende por control?**

Consiste en verificar si todo ocurre de conformidad con el plan adoptado, con las instrucciones emitidas y con los principios establecidos. Tiene como fin señalar las debilidades y errores para poder rectificarlos e impedir que se produzcan nuevamente. (Fayol, 2007)

## **1.3. Información generada en los procesos del METLab.**

Dentro de la información que se gestiona en el METLab de la UCI se encuentra:

- Jefe del Movimiento de la Universidad:
  - El cronograma de las inspecciones a realizar en cada una de las facultades.
  - El resultado de las inspecciones realizadas a las facultades, tanto sorpresivas como avisadas.
  - La emulación entre las facultades en cada mes, semestre y en el curso en general.
- Jefe del Movimiento de la Facultad:
  - El control de los principales datos de los estudiantes que pertenecen al Movimiento.
  - El cronograma de las guardias de cada uno de los estudiantes en el mes.
  - El control de los laboratorios que se van a destinar para la docencia.
  - La evaluación mensual de cada estudiante.
- Jefe del Movimiento de la Facultad y el Jefe de Turno:
  - La evaluación de cada estudiante en su guardia.
- Estudiante Técnico de Laboratorio de guardia y el Técnico:
  - El recibo y entrega de los laboratorios del estudiante cada vez que está de guardia.



**Figura 1 Procesos del METLab**

#### **1.4. Evaluación de los estudiantes.**

##### **1.4.1. ¿Qué se entiende por evaluación?**

Proceso sistemático y complejo donde se usa la información cuantitativa y cualitativa para determinar el nivel en el cual se logran los objetivos de la enseñanza. Se usa para juzgar y tomar decisiones respecto a la ejecución de un alumno, maestro o programa. (Meléndez, 2008)

La evaluación de los Estudiantes Técnicos de Laboratorios en la UCI se realiza para determinar la calidad del desempeño de los estudiantes en el Movimiento, permitiendo mediante ésta juzgar y tomar decisiones a la hora de determinar si ha cumplido o no con el reglamento del Movimiento. Las evaluaciones diarias las realizan los Jefes del Movimiento en cada una de las facultades y los Jefes de Turnos, no siendo así con las evaluaciones mensuales, las cuales sólo las dan los Jefes del Movimiento de las facultades. Las mismas se emiten diariamente, mensuales, semestrales y a finales del curso teniendo en cuenta las evaluaciones obtenidas en el periodo a evaluar. Las evaluaciones que se dan a los estudiantes es Bien (B), Regular (R) y Mal (M).

##### **1.4.2. Propuesta de un sistema de indicadores para la evaluación.**

Los indicadores a medir actualmente en el sistema de evaluación no están bien definidos ni uniformes para todas las facultades ni en todas las guardias, ya que cada facultad y cada Jefe del Movimiento o Jefe de Turno evalúa de acuerdo a su criterio.

Por tanto se propone un sistema de indicadores para estandarizar las evaluaciones de los estudiantes en las facultades.

Indicadores a medir:

- Cumplir y hacer cumplir el Reglamento de los laboratorios.

- Disciplina en los laboratorios.
  - Cumplimiento de las Normas de Vestuario de la Universidad.
  - Mantener la organización y la disciplina en el laboratorio.
  - Velar por los medios que se encuentran en los laboratorios.
  - Cumplir con las orientaciones de la dirección de los laboratorios y del METLab de la facultad y/o la Universidad.
- Cumplir las Normativas del METLab.
- Cumplir con el horario de la guardia.
  - No ausentarse del laboratorio por tiempos prolongados.
  - No ausentarse a la guardia sin autorización.

## **2. Sistemas de Gestión de Información.**

Los sistemas de gestión de información son un conjunto de políticas y normas relacionadas entre sí que se establecen para el acceso y tratamiento de los recursos de información. Incluye los registros administrativos y los archivos, el soporte tecnológico de los recursos y el público a que se destina. En su evolución, el sistema puede manejar la función de inteligencia corporativa y generar productos de inteligencia. (Gutián, 2007)

La gestión de la información es el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas. La información para la gestión es la información necesaria para tomar decisiones de gestión. (Bartle, 2009)

### **2.1 Sistemas de Gestión de información Laboral de los Recursos Humanos.**

Los Sistemas de Gestión de Información Laboral de los Recursos Humanos es un software donde se obtienen datos entregados por los empleados, del ambiente interno y externo de la organización y del complejo y multifacético entorno que afecta la actividad empresarial de dicha organización. Su finalidad es la de facilitar a la gerencia de la empresa información procesada y actualizada en formato electrónico a aplicar en instancias de evaluación y diagnóstico de su fuerza de trabajo, de sus fortalezas y debilidades en cuanto al modo en que están organizados los empleados y de las características del mercado laboral al cual acudir en situaciones de búsqueda de nuevo personal. Se trata de una herramienta que sirva de soporte en la toma de las mejores decisiones posibles a nivel de la gestión de Recursos Humanos de la empresa. (García, 2008)

Son usados en dependencia de las necesidades de cada una de las empresas, teniendo en cuenta las características de la misma ya que cada cual es diferente y está diseñado con un objetivo específico para esa empresa.

Entre estos se encuentra:

- ASSETS, es un Sistema de Gestión Integral que contiene un módulo de Recursos Humanos concebido para calcular las nóminas y controlar los recursos laborales de una entidad.
- Sistema de Información de Gestión de Recursos Humanos del Ministerio de Educación de Colombia.
- Sistema de Gestión de Recursos Humanos de la Universidad Nacional de Asunción.

- Sistema Hum&Select, empresa especializada en el desarrollo de sistemas de información para Recursos Humanos.

## **CAPÍTULO 2: TECNOLOGÍAS DE DESARROLLO.**

### **1. Metodologías de desarrollo del software**

El concepto de metodología, dentro de la Ingeniería del Software es, sin duda, uno de los más oscuros y que más confusión produce tanto en estudiantes como en profesionales involucrados en procesos de desarrollo de software. Tanto es así, que en muchos proyectos de desarrollo (no todos, por supuesto), la aplicación de una metodología brilla por su ausencia, siendo éste un concepto casi desconocido. La ausencia de metodología en el desarrollo de un proyecto de software garantiza con seguridad también la ausencia de calidad. (Roberto, 2009)

#### **1.1. ¿Qué es una metodología de desarrollo del software?**

La definición no es sencilla. Si autores de supuesto renombre llevan un montón de años en el tema y todavía no han logrado ponerse de acuerdo.

En el ciclo de vida del software se debían completar una serie de tareas para obtener un producto de software. A menudo, se dice que los distintos componentes de software deben pasar por distintas fases o etapas durante el ciclo de vida.

Cada una de esas tareas puede ser abordada y resuelta de múltiples maneras, con distintas herramientas y utilizando distintas técnicas. Es necesario saber cuándo podemos dar por concluida una tarea, quién debe realizarla, qué tareas preceden o anteceden a una dada, qué documentación utilizaremos para llevar a cabo esa tarea.

Estamos hablando de detalles organizativos, de un "estilo" de hacer las cosas. Pero yendo un poco más allá que un simple estilo, formalizando ese "estilo" añadiendo algo de rigurosidad y normas obtenemos una metodología.

Volvamos a las metodologías: todos los integrantes de un equipo de desarrollo deben seguir un criterio común a la hora de realizar las tareas del ciclo de vida. Ese criterio, esa manera común es una metodología de desarrollo.

A lo largo de los años se han propuesto numerosas metodologías. Algunas han sido escritas por autores del ámbito académico, otras por autores del ámbito empresarial de desarrollo del software y otras por administraciones públicas.

### **2. Metodología de Desarrollo de Software RUP**

Nosotros seleccionamos como la metodología de desarrollo de software a usar, el Proceso Unificado de Rational (RUP), el cual describe cómo aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas.

RUP provee a cada miembro del equipo las guías del proceso, plantillas y herramientas necesarias para que el equipo completo tome ventaja de, entre otras, las siguientes mejores prácticas:

- **Desarrollar software iterativamente**

Ya que nos permite mantener un enfoque iterativo que ayuda a detectar de forma temprana los desajustes e inconsistencias entre los requerimientos, el diseño, el desarrollo y la implementación del sistema.

- **Administrar los requerimientos**

Mediante los requerimientos podemos obtener, organizar y documentar todas las funcionalidades y restricciones requeridas por el cliente, así como documentar y monitorear las alternativas y decisiones a tomar.

- **Utilizar arquitecturas basadas en componentes**

Nos permitirá diseñar una arquitectura flexible, que se acomode a los cambios, permitiendo una efectividad en la reutilización de software. También nos permite trabajar basados en componentes, es decir, en módulos no triviales que completen una función clara.

- **Verificar la calidad de software**

Nos permite realizar pruebas para evaluar la calidad del sistema de acuerdo con sus requerimientos de funcionalidad, confiabilidad y rendimiento, permitiéndonos encontrar fallas antes de la puesta en explotación.

- **Controlar los cambios al software**

Nos permite tener un control de todos los cambios de modelos, código o documentos, realizados durante el desarrollo del software, permitiendo un desarrollo iterativo exitoso.

### **3. Lenguaje para el modelado: UML.**

Una exigencia de la gran mayoría de instituciones dentro de su Plan Informático estratégico, es que los desarrollos de software bajo una arquitectura en Capas, se formalicen con un lenguaje estándar y unificado. Es decir, se requiere que cada una de las partes que comprende el desarrollo de todo software de diseño orientado a objetos, se visualice, especifique y documente con lenguaje común. Se necesitaba un lenguaje que fuese gráfico, a fin de especificar y documentar un sistema de software, de un modo estándar incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema. Este lenguaje unificado que cumple con estos requerimientos, es ciertamente UML, el cual cuenta con una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos.

#### **3.1 ¿Qué es UML?**

El Lenguaje de Modelado Unificado (UML) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos, de un lenguaje de modelado y de un proceso. El UML, fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE. UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios.

El lenguaje de modelado es la notación que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.

Una de las metas principales de UML es avanzar en el estado de la integración institucional proporcionando herramientas de interoperabilidad para el modelado visual de objetos. Sin embargo, para lograr un intercambio exitoso de modelos de información entre herramientas, se requirió definir a UML una semántica y una notación. La notación es la parte gráfica que se ve en los modelos y representa la sintaxis del lenguaje de modelado. Por ejemplo, la notación del diagrama de clases define cómo se

representan los elementos y conceptos, como son: una clase, una asociación y una multiplicidad. ¿Y qué significa exactamente una asociación o multiplicidad en una clase? Un metamodelo es la manera de definir esto: un diagrama, usualmente de clases, que define la notación.

Una herramienta de UML debe mantener la consistencia entre los diagramas en un mismo modelo. Bajo esta definición una herramienta que sólo dibuje, no puede cumplir con la notación de UML.

## **4. Diferentes Lenguajes de programación.**

### **4.1 PHP**

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor, gratuito, rápido, con una gran librería de funciones y abundante documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, rapidez y seguridad. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

PHP, en el caso de estar instalado sobre un servidor Linux u Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria y esto evita las comunicaciones entre componentes COM que se realizan entre todas las tecnologías implicadas en una página ASP.

Por último la seguridad, en este punto también es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. Incluye funciones para el envío de correo electrónico, descarga de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales.

Otra de las capacidades más importantes de PHP es su compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por citar algunas.

### **4.2 JavaScript**

Javascript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con Javascript podemos crear efectos especiales en las páginas y definir formas de interacción con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan Movimiento, cambien de color o cualquier otro efecto. Por el otro, Javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes orientados a objetos, con funciones, estructuras de datos complejas, y otros. Además, pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. De modo que el programador se convierte en el verdadero dueño y controlador de cada cosa que ocurre en la página cuando la está visualizando el cliente.

## **5. Sistemas de Gestión de Contenido.**

Realizar un sistema web puede ser un trabajo complicado y muy laborioso si no se dispone de las herramientas adecuadas. En el pasado las herramientas eran básicamente editores que permitían generar una página, éstos evolucionaron para incorporar el control de la estructura de la web y otras funcionalidades, pero en general estaban enfocadas más a la creación que al mantenimiento. En los últimos años se ha desarrollado el concepto de sistema de gestión de contenidos (*Content Management Systems* o CMS). Se trata de herramientas que permiten crear y mantener una web con facilidad, encargándose de los trabajos más tediosos que hasta ahora ocupaban el tiempo de los administradores.

Teniendo en cuenta el ahorro que supone la utilización de estas herramientas y el coste de desarrollarlas, sería lógico esperar que su precio fuera muy elevado. Eso es cierto para algunos productos comerciales, pero existen potentes herramientas de gestión de contenidos de acceso libre, disponibles con licencias de código abierto.

Los gestores de contenidos proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios.

### **5.1 ¿Qué es un CMS?**

Los sistemas de gestión de contenidos (CMS, por sus siglas en inglés) son un software que se utilizan principalmente para facilitar la creación 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).

James Robertson (2003 b) propone una división de la funcionalidad de los sistemas de gestión de contenidos en cuatro categorías: creación de contenido, gestión de contenido, publicación y presentación.

Un CMS puede gestionar automáticamente la accesibilidad de la web, con soporte de normas internacionales de accesibilidad como WAI<sup>30</sup> y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas

las plataformas (Windows, Linux, Mac, Palm, y otros) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del visitante.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro de la web, añadiendo enlaces de forma automática. También gestiona todos los módulos, internos o externos, que incorpore al sistema. Así por ejemplo, con un módulo de noticias se presentarían las novedades aparecidas en otra web, con un módulo de publicidad se mostraría un anuncio o mensaje animado, y con un módulo de foro se podría mostrar, en la página principal, el título de los últimos mensajes recibidos. Todo eso con los enlaces correspondientes y, evidentemente, siguiendo el patrón que los diseñadores hayan creado.

### **Tipos de CMS.**

Existen diferentes tipos de CMS entre los que se encuentran:

- Sitios Web.
- Foros.
- Galerías.
- Groupware.
- Blogs

### **5.2 Necesidad de un CMS**

Se han presentado varios motivos para ver la utilidad de un sistema que gestione un entorno web, pero se podría pensar que no es necesario para la web relativamente pequeña o cuando no se necesitan tantas funcionalidades. Eso sólo podría ser cierto para la web con unas pocas páginas estáticas para el que no se prevea un crecimiento futuro ni muchas actualizaciones, lo que no es muy realista. En cualquier otro caso, la flexibilidad y escalabilidad que permiten estos sistemas justifican su utilización en prácticamente cualquier tipo de web.

Éstos son algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

- **Inclusión de nuevas funcionalidades en la web.** Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.
- **Gestión de usuarios:** Admite el registro de usuarios y la gestión de roles, permitiendo no tener que programar este módulo.
- **Reutilización de objetos o componentes.** Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- **Páginas interactivas.** Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámicas no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. Para conseguir esta interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la web. Esto es posible realizarlo sin un CMS pero toma más tiempo.
- **Cambios del aspecto de la web.** Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los

CMS facilitan los cambios con la utilización, por ejemplo, del estándar *Cascading Style Sheets* (CSS), que se traduce al español como Hojas de Estilo en Cascada, con lo que se consigue la independencia de presentación y contenido.

- **Control de acceso.** Controlar el acceso a la web no consiste simplemente permitir la entrada a la web, sino que soporta gestionar los diferentes permisos a cada área de la web aplicados a grupos o individuos.

## 5.3 Ejemplos de CMS.

### 5.3.1 Joomla.

Es un CMS de código libre, creado en PHP.

Joomla permite organizar eficientemente los contenidos de su sitio en secciones y categorías. Crear páginas ilimitadas y editarlas desde un sencillo editor que permite formatear los textos con los estilos e imágenes deseados. Ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos. Permite almacenar datos de usuarios registrados. Admite cambiar todo el aspecto del sitio, navegación y menú, administrador de imágenes, disposición de módulos modificable, encuestas, sindicación de noticias. Es posible hacer publicidad en el sitio usando el Administrador de Banners. Estadísticas de visitas.

El código HTML generado no es semántico. No existen acciones sólidas del Grupo de Desarrollo para hacer la aplicación más accesible. A pesar de la creciente importancia de la accesibilidad en las webs, poco se ha hecho al respecto con Joomla. Dependencia excesiva del JavaScript en su Panel de Administración. Limitación en las opciones para personalizar rangos de usuarios no sólo para personalizarlos, sino para crear nuevos rangos y permisos. Muchas veces se hace insuficiente con los actuales. Existe un componente denominado Community Builder que mejora un poco lo presente. Lo ideal sería un buen motor de usuarios que venga de forma nativa. Deficiente planificación de la interfaz administrativa. En especial nos referimos a la presencia de funciones redundantes que muchas veces han causado confusión a la hora de configurar el sitio. Igualmente hay quienes lo consideran cómodo desde la primera vez.

Para mayor información sobre este CMS puede visitarse su sitio oficial en la dirección: <http://joomla.org/>

### 5.3.2 WordPress:

El CMS para la creación de blogs por excelencia. El más utilizado y el mejor valorado, también creado en PHP y gratuito.

En WordPress la administración está hecha para administrar artículos, categorías, comentarios y usuarios. Es una plataforma muy fácil de usar, quizás también porque es una plataforma creada exclusivamente para blogs. Existe una enorme cantidad de complementos que son compatibles entre sí, esto permite instalar 2 o 3 de ellos para tener la funcionalidad necesaria sin escribir una línea de código. El sistema de WordPress es bastante potente y elegante, gran parte de las modificaciones se realizan mediante etiquetas en la plantilla. Es fácil de modificar pero nunca deja de ser un blog<sup>2</sup>.

Una gran desventaja es que en WordPress no se puede modificar el código del sitio. La mayoría de los usuarios usa Blogger<sup>5</sup>, y la cuenta de WordPress no se presta para dejar comentarios en los blogs de Blogger que requieren identificación. Es muy fácil pasar todos los artículos y comentarios de Blogger a WordPress, pero es imposible modificar todos los artículos antiguos en bloque para adaptarlos al nuevo blog. Dependiendo de la cantidad y frecuencia de las publicaciones, la administración del Blog puede dificultarse si no se hace una revisión periódica. Por ser un espacio de acceso público pueden recibirse

comentarios no deseados que no se encuentren relacionados con las temáticas. No existe la opción de obtener una versión local del blog.

Para mayor información sobre este CMS puede visitarse su sitio oficial en la dirección: <http://wordpress.org/>

### **5.3.3 Drupal.**

Uno de los CMS más populares, en este caso gratuito y de código abierto. Creado en PHP y con posibilidad de utilizar varias bases de datos distintas.

Drupal permite configurar un "libro" y autorizar a otras personas a contribuir con el contenido. Permite personalizar las URL<sup>29</sup> para la optimización en los motores de búsqueda. Proporcionan la funcionalidad extra a Drupal. Cuenta con un robusto sistema de ayuda en línea. El código fuente está libremente disponible bajo los términos de la Licencia Pública General de GNU (GPL) 2. Tiene disponible el código fuente en su totalidad para ampliar o adaptar, según sea necesario. Posee un robusto entorno de personalización, tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario. Los administradores no tienen que configurar los permisos para cada usuario, simplemente asignar usuarios a roles. Todo el contenido está en el índice de búsqueda, lo que facilita la información al usuario. Los usuarios se pueden registrar y autenticar a nivel local ó utilizando una fuente de autenticación externa como Jabber, Blogger, LiveJournal u otro sitio de Drupal. Se puede integrar con un servidor LDAP<sup>16</sup>. Contiene un módulo de encuesta que permite a los administradores o usuarios crear encuestas y mostrarlas en distintas páginas. El sistema de plantillas separa el contenido de la presentación permitiendo controlar el aspecto del sitio. Ofrece un potente modelo de hilos de comentarios para permitir discusiones sobre el contenido publicado. Soporte para Apache o IIS<sup>13</sup>, Unix / Linux / BSD / Solaris / Windows / Mac OS X. Es construido con una capa de abstracción de bases de datos que le permite ser utilizado con MySQL y PostgreSQL, al igual que otras bases de datos propietarias. Puede ser traducido a varios idiomas. El caché elimina el mecanismo de búsqueda en bases de datos incrementando el rendimiento y reduciendo la carga del servidor. Gran cantidad de opciones de configuración con una interfaz amigable para el administrador. De forma nativa tiene mayor cantidad de opciones que otros CMS, su comunidad es muy activa y existen instalaciones personalizadas.

Drupal no es muy recomendable para sitios web empresariales, sus funciones escapan a lo que debería ofrecer un portal empresarial. Alta curva de aprendizaje.

Para mayor información sobre este CMS puede visitarse su sitio oficial en la dirección: <http://drupal.org/>

## **5.4 ¿Por qué usar el CMS Drupal?**

Drupal es un sistema de gestión de contenidos de código abierto que sirve para administrar recursos web. Es un sistema multiusuario, multiplataforma, multilenguaje, extensible, modular. Está desarrollado en PHP y puede instalarse sobre Apache o IIS. Como bases de datos soporta tanto MySQL y PostgreSQL. La gran mayoría de módulos y temas disponibles también están disponibles bajo esta licencia.

Se destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la Web, y un énfasis especial en la usabilidad y consistencia de todo el sistema. El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar diferentes tipos de sitios Web.

Si bien las características básicas de Drupal son suficientes como para lanzar un sitio completo, el núcleo incluye: administración de usuarios, páginas, vocabulario, comentarios y sindicación. Su potencialidad

reside en su estructura modular que permite ser extendida y customizada a gusto y placer del administrador para distintas funciones por medio de módulos de fácil instalación. Entonces este multifacético CMS nos permitiría armar desde un blog personal hasta una intranet corporativa, pasando por bibliotecas digitales, wikis, sitios de comercio electrónico, álbumes de fotos, etcétera.

## **6. Arquitectura cliente – servidor.**

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización. El cliente es el que inicia un requerimiento de servicio. El requerimiento inicial puede convertirse en múltiples requerimientos de trabajo a través de redes LAN<sup>15</sup> o WAN<sup>31</sup>. La ubicación de los datos o de las aplicaciones es totalmente transparente para el cliente. Un servidor es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes y otros.

### **6.1 Características del modelo cliente – servidor.**

La arquitectura cliente - servidor es una relación entre procesos corriendo en máquinas separadas donde el servidor es un proveedor de servicios, el cliente es un consumidor de servicios, por lo que interactúan por un mecanismo de pasaje de mensajes: pedido de servicio y respuesta.

En el modelo cliente – servidor podemos encontrar las siguientes características:

- Las funciones de cliente y servidor pueden estar en la misma plataforma.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un sólo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un sólo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.
- Designa un modelo de construcción de sistemas informáticos de carácter distribuido. Su representación típica es una computadora personal (PC), donde el usuario dispone de sus propias aplicaciones de oficina y bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los recursos de este host central.

En conclusión cliente – servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC<sup>3</sup>, TCP/IP<sup>28</sup>, OSI<sup>24</sup>, NFS<sup>22</sup> sólo por mencionar algunas de las posibilidades.

## **7. Servidor Web Apache.**

Apache es el servidor web líder en el mercado. Su coste gratuito, gran fiabilidad y extensibilidad le convierten en una herramienta potente y muy configurable.

Ventajas principales:

- **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan bajo servidores Apache.
- **Software Libre:** Apache es totalmente gratuito y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.
- **Extensibilidad:** se pueden añadir módulos para ampliar las capacidades que posee.
- También dispone de una amplia variedad de módulos, que permiten generar contenido dinámico (con PHP, Java, Perl, Python), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, crear servidores virtuales por IP o por nombre (varias direcciones web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso modificados por cualquier persona con conocimientos de programación.

## 8. Sistemas Gestores de Base de Datos.

### 8.1 MySQL

Su principal objetivo de diseño fue la velocidad, por ello se suprimieron algunas características de los demás SGBD (Sistema de Gestión de Base de Datos), como las transacciones y las subconsultas. Consume pocos recursos y se distribuye bajo licencia GPL (a partir de la versión 3.22).

#### Ventajas:

- Integración perfecta con PHP
- Sin límites en los tamaños de los registros
- Mejor control de acceso de usuarios
- Está escrito en C y C++ y probado con multitud de compiladores. Dispone de APIs<sup>2</sup> para muchas plataformas diferentes como por ejemplo Java, Unix, Windows, entre otras.
- Conectividad: permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es común que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris y otros.
- Es multihilo, con lo que puede beneficiarse de sistemas multiprocesadores.
- Permite manejar multitud de tipos de datos.
- Permite manejar registros de longitud fija o variable.

#### Desventajas:

- No tiene integridad referencial lo cual hace de este gestor una solución pobre sobre todo para aquellos programadores que provienen de otros gestores que sí poseen esta característica.
- Lento con grandes bases de datos.
- No tiene tantas capacidades como otros gestores profesionales.
- Dificultad de manejo.
- Un gran porcentaje de utilidades no están documentadas. (Illera, 2009)

### 8.2 PostgreSQL

PostgreSQL surgió para el desarrollo de un sistema de bases de datos que se ha mantenido bajo la filosofía de desarrollo de software libre. Ha sido descrito como un sistema de bases de datos relacional orientado a objetos (ORDBMS por sus siglas en inglés) basado en POSTGRES desarrollado en el Departamento de Ciencias de la Computación de la Universidad de Berkeley en California.

### **Ventajas:**

- Instalación ilimitada

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto tiene varias ventajas adicionales:

- Modelos de negocios más rentables con instalaciones a gran escala.
  - No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
  - Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- Mejor soporte que los proveedores comerciales.
  - Ahorros considerables en costos de operación
  - Estabilidad y confiabilidad legendarias
  - Extensible

El código fuente está disponible para todos sin costo. Si se necesita extender o personalizar PostgreSQL de alguna manera, se puede hacer con un mínimo esfuerzo, sin costos adicionales.

- Multiplataforma

PostgreSQL está disponible en casi cualquier Unix.

- Diseñado para ambientes de alto volumen

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Control de Concurrencia para Múltiples Versiones) para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

### **Desventajas:**

- Es más lento ante la respuesta a una única consulta.
- Limitaciones al escribir funciones y procedimientos.
- Carencia de herramientas de desarrollo propia.
- Es un sistema robusto, por lo que consume gran cantidad de memoria. (Eduardo, 2005)

## 9. ¿Por qué usar PostgreSQL?

SGBD PostgreSQL es un poderoso sistema manejador de bases de datos, es un sistema diseñado para administrar grandes cantidades de datos, el cual es considerado como una de las bases de datos de código abierto más avanzada del mundo. PostgreSQL se ha preocupado por ser una solución real a los complejos problemas del mundo empresarial y a la vez mantener la eficiencia al consultar los datos. Con ese fin, se han desarrollado y añadido a PostgreSQL las más interesantes y útiles características que antes sólo podían hallarse en sistemas manejadores de bases de datos comerciales como Oracle, DB2 o Sybase. Tiene además un alto rendimiento, estable y capa. Consultado para lidiar con grandes volúmenes de datos. Estas características de PostgreSQL son las que garantizan la integridad de los datos, la velocidad de acceso y consultas a la base de datos.

## 10. ¿Por qué usar la trilogía Apache + PHP + PostgreSQL?

Apache permite la integración con PHP que es uno de los lenguajes más usados en la realización de aplicaciones web, al igual que el uso de PostgreSQL para el desarrollo de sistemas de base de datos y es el que se usa en la Universidad. Por lo tanto teniendo en cuenta estos aspectos se decidió usar la trilogía Apache + PHP + PostgreSQL. La elección se fundamenta con las prestaciones descritas anteriormente y además con los elementos que a continuación se enuncian:

- Representa la combinación de código abierto que ofrece mayores prestaciones para el desarrollo de aplicaciones web en la actualidad.
- Constituye una variante libre y perfectamente integrada.
- Permite crear aplicaciones con alta portabilidad debido a que son tecnologías multiplataforma.
- Permiten la creación de sistemas altamente seguros.

## 11. Herramientas utilizadas para la elaboración del sistema.

### 11.1 Visual Paradigm para UML 6.4.

Es una herramienta para el modelado UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML.

Lista de características:

- Es software libre.
- Tiene soporte para UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento
- Modelado colaborativo con CVS y Subversion.
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XML.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL.

- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde la base de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde SGBD existentes a diagramas de Entidad-Relación.
- Generador de informes para generación de documentación.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.
- Importación y exportación de ficheros XMI.
- Integración con Visio - Dibujo de diagramas UML con plantillas de MS Visio Editor de figuras.

### **11.2 Embarcadero ERStudio 8.0.**

ER/Studio es una herramienta CASE muy fácil de usar, con soporte de múltiples niveles de análisis y que permite realizar tanto el modelado lógico de los datos como la implantación física en la base de datos elegida para tal fin. ER/Studio, soporta más de 30 de las bases de datos más importantes de la industria, entre ellas: Oracle, Sybase, IBM DB2 (tanto para Sistemas Abiertos como para mainframes con sistema operativo OS/390 & z/OS), IBM DB2/400, Informix, Paradox, Microsoft SQL Server, Access, Interbase, y otros. (Vidal, 2009)

ER/Studio Enterprise posee poderosas características de modelación que aceleran el ciclo de diseño, como son:

- Modelado relacional y multidimensional.
- Integración con productos de “Inteligencia de Negocios” como Business Objects, Cognos, Hyperion, DB2 Cubes, y otros
- Exportación e importación de metadatos a más de 30 formatos diferentes.
- Diseño físico y lógico separados.
- Transformación automática de un diseño lógico a un diseño físico para una plataforma específica de base de datos.
- Propagación bi-direccional de cambios, que permiten sincronizar los modelos físicos y lógicos con las estructuras de las tablas contenidas en la base de datos.
- Modelación por áreas de interés. Mediante la definición de “vistas” y “sub-vistas” es posible enfocar el análisis a una cierta área de interés, sin romper la relación con el resto del modelo.
- Extensa validación de los modelos. Es posible validar ampliamente la calidad e integridad tanto del diseño lógico como del diseño físico del modelo.

- Repositorio para trabajo en colaboración, definición de seguridad y control de versiones sobre los modelos. (Vidal, 2009)

### 11.3 PHPDesigner 7.

El entorno de desarrollo de PHPDesigner es bastante atractivo y funcional, con características comunes de otros editores PHP o IDEs PHP. En un primer momento resulta bastante sencillo para entender el mecanismo de funcionamiento, si es que tiene experiencia con otros entornos. Si no posee soltura en el manejo de aplicaciones de productividad para programadores, quizás se haga un poco más complicado de entender o de obtener de él todo el provecho. En cualquier caso, resulta muy sencillo de utilizar como un simple editor de PHP, seleccionando los ficheros que se desee editar a través del explorador de archivos del ordenador o del gestor de conexiones FTP<sup>10</sup>.

Entre los puntos más interesantes que se encuentran disponibles en este programa se puede destacar:

- Código resaltado, que soporta los lenguajes típicos que se utilizan a la hora de crear aplicaciones web: HTML, CSS, Javascript, XML, además de PHP, claro está. En este caso se ha encontrado una diferencia con respecto a otros programas que se habían probado, puesto que el coloreado de código es dinámico, destacando sólo las porciones de código del lenguaje utilizado en el lugar donde están situados con el cursor. Por ejemplo, si en el código están mezclados PHP y HTML, al situarse en una porción de código PHP todo el scripting en este lenguaje se destacará en colores y el código HTML aparecerá todo en gris. Si se coloca en una parte de código HTML, se coloreará así como todas las otras porciones HTML, y el código PHP se verá en gris.
- Ayuda para el completado de código como cierre automático de paréntesis, comillas y otros elementos. Sin lugar a dudas una bondad llamativa son los mensajes contextuales a la hora de escribir llamadas a funciones, los cuales consisten en pequeños textos flotantes con declaraciones de las funciones que se están utilizando, sus parámetros, una breve descripción de la funcionalidad y en las versiones de PHP donde está disponible. Las ayudas de completado de código incluyen todas las funciones de PHP y de las librerías que se están utilizando en un proyecto o se hayan incluido en un script en particular.
- Administración de proyectos, con múltiples opciones de configuración.
- Traceo de código PHP, por medio de Xdebug.
- Tiene un corrector de sintaxis PHP en línea, que funciona a la vez que se escribe el código. Esto puede ser útil, porque permite ver los errores que se han cometido en sintaxis a la vez que se escribe, pero realmente a veces entorpecen el trabajo los mensajes de error en líneas de código que aún no se han terminado y que dan errores porque todavía no se han colocado un punto y coma, por ejemplo.
- Tiene funcionalidad de conexión remota con servidores de FTP, para editar en línea un archivo que está en otro servidor. Además, los proyectos también se pueden asociar con un espacio de hosting por medio de FTP.
- Búsqueda avanzada de textos en archivos, así como funciones de reemplazo de textos.
- Códigos y plantillas ya listos para simplificar ciertas tareas repetitivas, que se pueden personalizar según las necesidades de cada desarrollador.
- Integración con la documentación de PHP.

Todas estas características lo convierten en un producto muy apetecible para desarrolladores en PHP, que seguro que, con un poco de inversión en tiempo para aprender a manejarlo, ayudará a sacarle un mayor provecho de horas de programación.

## **Capítulo 3: Descripción del sistema**

### **1. Modelo del negocio.**

#### **1.1. Descripción de los procesos del negocio y las mejoras propuestas.**

El Movimiento de Estudiantes Técnicos de Laboratorios (METLab) en la Universidad de Ciencias Informáticas tiene como objetivo, el apoyo a los técnicos de laboratorios durante el horario nocturno, vinculando así a los estudiantes a realizar guardias en los laboratorios en el horario de la noche, logrando con esto una prestación de servicio de los laboratorios con la mayor calidad posible.

En el METLab tanto a nivel de Facultad como central se realizan varios procesos entre los cuales se encuentran las evaluaciones diarias y mensuales a los estudiantes, la planificación de sus guardias, la entrega y el recibo de los laboratorios, la planificación de las inspecciones a las facultades, así como los resultados de la misma y la emulación entre las facultades.

La situación problemática está dada actualmente porque estos procesos no se llevan a cabo con la mayor eficiencia, ya que por el gran cúmulo de información se gastan muchos recursos imprimiendo modelos, partes, etcétera, por lo que no siempre se tiene los medios necesarios para poder realizar un trabajo de excelencia. No se tiene un control de los estudiantes que pertenecen al Movimiento, ya que a veces las facultades tienen un listado diferente a los que realmente están trabajando en el Movimiento, originando esto que existan estudiantes recibiendo una remuneración y que no están trabajando o viceversa. También en muchas ocasiones ocurren pérdidas de los documentos, provocando que no se puedan obtener informaciones necesarias para el buen funcionamiento del Movimiento, como por ejemplo la pérdida de las evaluaciones diarias para poder dar una evaluación mensual teniendo en cuenta la trayectoria del estudiante durante ese mes, provocando así que no se pueda dar una valoración adecuada durante ese periodo, lo cual ocasiona que a veces hayan estudiantes que no cumplen con su trabajo, pero que por no tener constancia del mismo, no se les pueda dar la evaluación que merece. Tampoco existen los aspectos a tener en cuenta en el proceso de evaluación, lo cual provoca que no se evalúen a todos los estudiantes de igual manera en todas las facultades, si no que cada facultad y en cada guardia un estudiante es evaluado de forma diferente al anterior, causando esto insatisfacciones hacia el sistema de evaluación por parte de los mismos estudiantes.

Dado el contexto problemático planteado se propone un software que permita tener un control de los estudiantes que pertenecen al Movimiento, de todas las evaluaciones de los mismos, su planificación de la guardia, la entrega y el recibo de los laboratorios, las inspecciones a las facultades, el resultado de las mismas y la emulación entre las facultades. Permitiendo así que todas las personas que lo necesiten puedan tener acceso a esta información sin ningún problema, y que la misma no sufra ningún tipo de pérdida. Con esto se logra un buen funcionamiento del Movimiento, una mayor calidad en la prestación de servicio y gran satisfacción hacia el sistema de evaluación por parte de los estudiantes.

Para lograr tener un control sobre los estudiantes que pertenecen al Movimiento, el Jefe del METLab de la Facultad a inicio de semestre realiza la ratificación de los estudiantes que pertenecen al mismo. También verifica con la dirección de la facultad la cantidad de laboratorios docentes de que va a disponer en ese curso, y teniendo en cuenta los laboratorios docentes, así como la cantidad de estudiantes ratificados y las orientaciones del Jefe del METLab de la UCI en cuanto al sistema de trabajo para ese semestre, se decide si se seleccionan nuevos estudiantes para el Movimiento o no. Luego de tener estructurado el Movimiento en la Facultad, el Jefe del METLab de la Facultad realiza

el cronograma de la guardia del mes correspondiente a partir de recibir la orientación de cuándo comienza a trabajar. El cronograma de la guardia se realiza teniendo en cuenta la cantidad de estudiantes por laboratorios y que cada estudiante descansa el mismo periodo de tiempo que el resto. En cada guardia a los estudiantes se le realiza una evaluación para controlar el desempeño de su trabajo durante su guardia, para eso el Jefe del METLab de la Facultad pasa por los laboratorios revisando el cumplimiento de los deberes y responsabilidades del Estudiante Técnico de Laboratorios (ETL), dándole una evaluación al mismo como resultado de su trabajo. Esta evaluación es revisada por el Jefe de Turno de la Dirección de los Laboratorio y aprobada por él de acuerdo a su criterio por el cumplimiento de alguna tarea asignada al ETL o por su trayectoria durante su guardia. Al final de cada guardia se le informa al ETL la evaluación obtenida durante la misma. Al final de cada mes el Jefe del METLab de la UCI le solicita del Jefe del METLab de la Facultad el resumen de las evaluaciones mensuales de los ETL durante el mes anterior, en el cual el Jefe del METLab de la Facultad teniendo en cuenta las evaluaciones diarias obtenidas por cada ETL le da una evaluación mensual como resultado al trabajo realizado durante ese mes, le informa al ETL su evaluación mensual y éste la aprueba. Los ETL al comenzar su guardia reciben el laboratorio, en el cual el Técnico les entrega el laboratorio con todos los medios, reflejando la cantidad de los mismo en el modelo del parte, el ETL revisa el modelo comprobando que todo este bien y firma el parte. Al terminar su guardia el ETL le informa al Técnico que va a entregar el laboratorio llenando así el parte con la cantidad física de los medios en el laboratorio, el Técnico revisa el laboratorio comprobando que todo está como se refleja en el parte y firma el mismo. Durante el mes la Dirección de Laboratorio y el Jefe del METLab de la UCI realizan inspecciones a los laboratorios docentes de las facultades para controlar y verificar el funcionamiento del Movimiento en las facultades. Este cronograma es solicitado por la Dirección de Laboratorios a finales de cada mes, en el cual el Jefe del METLab de la UCI debe entregar la planificación de las inspecciones en el mes siguiente. En cada inspección se tienen en cuenta algunos parámetros a medir, los cuales arrojan el resultado de la inspección, el cual es solicitado por los Jefes del METLab de la Facultad para saber cómo salió su Facultad en la misma. Este resultado es entregado por el Jefe del METLab de la UCI y es revisado y aprobado por el Jefe del METLab de la Facultad inspeccionada. Entre las facultades se lleva a cabo una emulación la cual permite que se logre un trabajo con mayor eficiencia. A finales de cada mes se entregan los resultados de la emulación entre las facultades donde se discuten con los Jefes del METLab de la Facultad, el cual es quien aprueba los resultados de la misma.

## 1.2. Actores y trabajadores del negocio.

Actores del negocio	Justificación
Directivo de la Dirección de Laboratorios	Es quien solicita el cronograma de las inspecciones a las facultades.
Jefe del METLab de la Facultad	Es quien solicita los resultados de las inspecciones, evalúa diariamente a los Estudiantes Técnicos de Laboratorios.

Jefe del METLab de la UCI	Es quien da a conocer los resultados de la emulación y solicita el cronograma de las guardias en el mes y las evaluaciones mensuales del mes anterior.
Estudiante Técnico de Laboratorio	Es el encargado de recibir y entregar el laboratorio.

<b>Trabajadores del negocio</b>	<b>Justificación</b>
Jefe del METLab de la UCI	Es el encargado de realizar el cronograma de las inspecciones a las facultades, dar los resultados de las mismas y de la emulación.
Jefe del METLab de la Facultad	Es quien aprueba los resultados de la emulación, realiza el cronograma de las guardias y da las evaluaciones diarias y mensuales.
Jefe de Turno de la Dirección de Laboratorios	Es el encargado de aprobar las evaluaciones diarias de los estudiantes.
Estudiante Técnico de Laboratorio	Es quien aprueba su cronograma de guardia en el mes y su evaluación mensual.
Técnico	Es quien entrega y recibe el laboratorio.

### 1.3. Diagrama de los Casos de Usos del Negocio.

### Diagrama de casos de uso del negocio

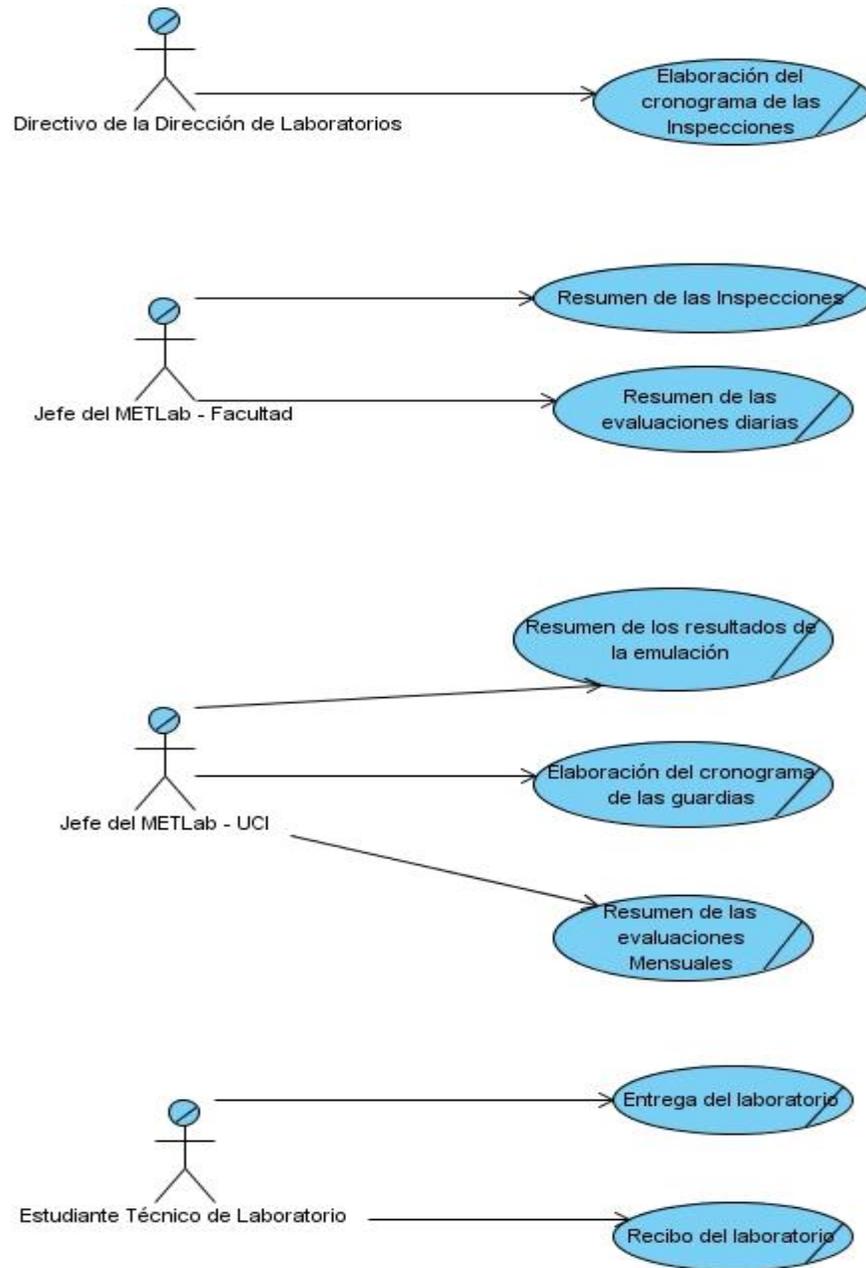


Figura 2 Diagrama de Casos de Usos del Negocio

#### 1.4. Listado de casos de uso del negocio expandidos.

<b>Caso de Uso</b>	Elaboración del cronograma de las Inspecciones	
<b>Actor</b>	Directivo de la Dirección de Laboratorios	
<b>Trabajadores</b>	Jefe del METLab de la UCI	
<b>Resumen</b>	El caso de uso se inicia cuando el Directivo de la Dirección de Laboratorios le solicita el Jefe del METLab de la UCI el cronograma de las inspecciones a realizar a los laboratorios de cada una de las facultades. El Jefe del METLab de la UCI elabora el cronograma. El caso de uso termina cuando el Directivo de la Dirección de Laboratorios recibe el cronograma.	
<b>Precondiciones:</b>	-	
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
1. Solicita el cronograma de las inspecciones.  4. Recibe el cronograma	2. El Jefe del METLab de la UCI, elabora el cronograma de las inspecciones.  3. Entrega el cronograma.	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
<b>Poscondiciones</b>	Documento con el cronograma de las inspecciones.	
<b>Mejoras</b>	Informatizar este proceso para poder gestionar las inspecciones a los laboratorios para que los interesados puedan tener acceso al cronograma previsto y saber de cualquier cambio.	

<b>Caso de Uso</b>	Resumen de las inspecciones
<b>Actor</b>	Jefe del METLab de la Facultad (inicia)
<b>Trabajadores</b>	Jefe del METLab de la UCI
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad le solicita al

	Jefe del METLab de la UCI los resultados de las inspecciones realizadas. El caso de uso termina cuando el Jefe del METLab de la Facultad está de acuerdo con la información o recibe la misma.
<b>Precondiciones:</b>	Documento con el cronograma de las inspecciones
<b>Flujo Normal de Eventos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
1. Solicita los resultados de las inspecciones realizadas. 3. Revisa la información. 4. Informa que está de acuerdo con la información.	2. El Jefe del METLab de la UCI entrega la información.  5. Recibe confirmación de aprobación de la información
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
4. Informa que no está de acuerdo con la información y da a conocer el error.  7. Recibe la información.	5. El Jefe del METLab de la UCI revisa y arregla la información.  6. Entrega la información.
<b>Poscondiciones</b>	Documento con los resultados de las inspecciones
<b>Mejoras</b>	Informatizar este proceso de forma tal que el Jefe del METLab de la Facultad pueda tener acceso a los resultados de la inspección en el momento que se termine, realizando cualquier reclamación referente a la misma y pueda ser corregida.

<b>Caso de Uso</b>	Resumen de las evaluaciones diarias
<b>Actor</b>	Jefe del METLab de la Facultad (inicia)
<b>Trabajadores</b>	Jefe de Turno de la Dirección de Laboratorios
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad recorre los laboratorios evaluando a cada estudiante técnico de laboratorio y le pregunta al Jefe de Turno de la Dirección de los Laboratorios la opinión sobre cada uno. El Jefe de Turno aprueba la evaluación de cada estudiante. El caso de uso

	termina cuando se les informa a los estudiantes técnicos de laboratorios su evaluación.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
<ol style="list-style-type: none"> <li>1. Emite una evaluación de cada estudiante técnico de laboratorio.</li> <li>2. Le informa al Jefe de Turno sobre la evaluación de cada Estudiante Técnico de Laboratorios.</li> <li>4. Le informa a los Estudiantes Técnicos de Laboratorios su evaluación.</li> </ol>	<ol style="list-style-type: none"> <li>3. El Jefe de Turno aprueba las evaluaciones de cada Estudiante Técnico de Laboratorio.</li> </ol>
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
<ol style="list-style-type: none"> <li>4. Revisa las evaluaciones y las corrige.</li> <li>5. Le informa a los Estudiantes Técnicos de Laboratorios su evaluación.</li> </ol>	<ol style="list-style-type: none"> <li>3. El Jefe de Turno no aprueba las evaluaciones de los Estudiantes Técnicos de Laboratorios y plantea los elementos justificantes.</li> </ol>
<b>Poscondiciones</b>	Documento con el registro de las evaluaciones diarias.
<b>Mejoras</b>	Informatizar este proceso de forma tal que el Jefe de Turno pueda tener acceso a la evaluación de los Estudiantes Técnicos de Laboratorios y pueda gestionar la misma en conjunto con el Jefe del METLab de la Facultad, permitiéndole también al Estudiante Técnico de Laboratorios conocer su evaluación al finalizar su guardia.

<b>Caso de Uso</b>	Resumen de los resultados de la emulación	
<b>Actor</b>	Jefe del METLab de la UCI (inicia)	
<b>Trabajadores</b>	Jefe del METLab de la Facultad	
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI le da a conocer los resultados de la emulación del mes al Jefe del METLab de la Facultad, el cual aprueba los resultados de la emulación. El caso de uso termina cuando el Jefe del METLab de la UCI da a conocer a la FEU los resultados de la emulación.	
<b>Precondiciones:</b>	Documento con los resultados de las inspecciones	
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
1. Informa los resultados de la emulación.  4. Da a conocer a la FEU los resultados de la emulación.	2. El Jefe del METLab de la Facultad revisa la información. 3. Aprueba la emulación.	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
4. Revisa los resultados de la emulación y los corrige. 5. Da a conocer a la FEU los resultados de la emulación.	3. El Jefe del METLab de la Facultad no aprueba los resultados de la emulación y plantea los elementos justificantes.	
<b>Poscondiciones</b>	Documento con los resultados de la emulación	
<b>Mejoras</b>	Informatizar este proceso de forma tal que el Jefe del METLab de la UCI pueda gestionar los resultados de la emulación y permitiéndole al Jefe del METLab de la Facultad tener acceso a la misma.	

<b>Caso de Uso</b>	Elaboración del cronograma de las Guardias	
<b>Actor</b>	Jefe del METLab de la UCI (inicia)	
<b>Trabajadores</b>	Jefe del METLab de la Facultad Estudiante Técnico de Laboratorios	
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI le solicita el Jefe del METLab de la Facultad el cronograma de las Guardias del mes entrante. El Jefe del METLab de la Facultad elabora el cronograma. El ETL lo revisa y lo aprueba. El caso de uso termina cuando el Jefe del METLab de la UCI recibe el cronograma.	
<b>Precondiciones:</b>	-	
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
1. Solicita el cronograma de las guardias.  7. Recibe el cronograma	2. El Jefe del METLab de la Facultad elabora el cronograma. 3. Envía el cronograma a los ETL. 4. El ETL revisa el cronograma. 5. El ETL aprueba el cronograma. 6. Entrega el cronograma.	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
8. Recibe el cronograma.	5. El ETL no aprueba el cronograma y plantea los elementos. 6. El Jefe del METLab de la Facultad revisa el cronograma y lo arregla. 7. Entrega el cronograma.	
<b>Poscondiciones</b>	Documento con el cronograma de las guardias.	
<b>Mejoras</b>	Informatizar este proceso para poder gestionar las guardias de los laboratorios para que los interesados puedan tener acceso a la misma y saber de cualquier cambio.	

<b>Caso de Uso</b>	Resumen de las evaluaciones mensuales	
<b>Actor</b>	Jefe del METLab de la UCI (inicia)	
<b>Trabajadores</b>	Jefe del METLab de la Facultad Estudiante Técnico de Laboratorio	
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI le solicita al Jefe del METLab de la Facultad las evaluaciones mensuales de los ETL. Este revisa la evaluación con los ETL y las aprueba. El caso de uso termina cuando el Jefe del METLab de la UCI recibe las evaluaciones mensuales.	
<b>Precondiciones:</b>	Documento con el registro de las evaluaciones diarias.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
1. Solicita las evaluaciones mensuales.       7. Recibe las evaluaciones.	2. El Jefe del METLab de la Facultad revisa las evaluaciones diarias de cada ETL y determina la evaluación mensual.  3. Le informa al ETL su evaluación mensual.  4. El ETL revisa su evaluación.  5. El ETL aprueba su evaluación.  6. Entrega las evaluaciones.	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
8. Recibe las evaluaciones	5. El ETL no aprueba su evaluación y plantea los elementos. 6. El Jefe del METLab revisa las evaluaciones y las corrige. 7. Entrega las evaluaciones	
<b>Poscondiciones</b>	Documento con el registro de las evaluaciones mensuales.	

<b>Mejoras</b>	Informatizar este proceso de forma tal que el Jefe de Turno pueda tener acceso a la evaluación de los Estudiantes Técnicos de Laboratorios y pueda gestionar la misma en conjunto con el Jefe del METLab de la Facultad, permitiéndole también al Estudiante Técnico de Laboratorios conocer su evaluación al finalizar su guardia.
----------------	---

<b>Caso de Uso</b>	Recibo del laboratorio	
<b>Actor</b>	Estudiante Técnico de Laboratorio (inicia)	
<b>Trabajadores</b>	Técnico	
<b>Resumen</b>	El caso de uso se inicia cuando el Estudiante Técnico de Laboratorio le solicita al Técnico la entrega del laboratorio. El Técnico llena el modelo del parte y procede a la entrega del laboratorio. El Estudiante Técnico de Laboratorio revisa el laboratorio. El caso de uso termina cuando el Estudiante Técnico de Laboratorio firma el modelo del parte.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
<ol style="list-style-type: none"> <li>1. Llega al laboratorio y solicita la entrega del mismo.</li> <li>3. Revisa el laboratorio verificando que todo esté como se refleja en el parte.</li> <li>5. Recibe el laboratorio y firma el modelo del parte.</li> </ol>	<ol style="list-style-type: none"> <li>2. El Técnico llena el modelo del parte.</li> <li>4. Realiza la entrega del laboratorio.</li> </ol>	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>	
<ol style="list-style-type: none"> <li>4. No coincide lo que esta en el modelo del parte con lo que esta físico en el laboratorio.</li> <li>7. Recibe el laboratorio y firma el modelo del parte.</li> </ol>	<ol style="list-style-type: none"> <li>5. El Técnico revisa el laboratorio y arregla el modelo del parte.</li> <li>6. Realiza la entrega del laboratorio.</li> </ol>	

<b>Poscondiciones</b>	Modelo del parte de la entrega
<b>Mejoras</b>	Informatizar este proceso para poder gestionar el recibo de los laboratorios y tener control de las pérdidas en los laboratorios.

<b>Caso de Uso</b>	Entrega del laboratorio
<b>Actor</b>	Estudiante Técnico de Laboratorio (inicia)
<b>Trabajadores</b>	Técnico
<b>Resumen</b>	El caso de uso se inicia cuando el Estudiante Técnico de Laboratorio le solicita al Técnico que reciba el laboratorio. El Estudiante Técnico de Laboratorio llena el modelo del parte y procede a la entrega del laboratorio. El Técnico revisa el mismo. El caso de uso termina cuando el Técnico firma el modelo del parte y recibe el laboratorio.
<b>Precondiciones:</b>	Modelo del parte de la entrega

#### Flujo Normal de Eventos

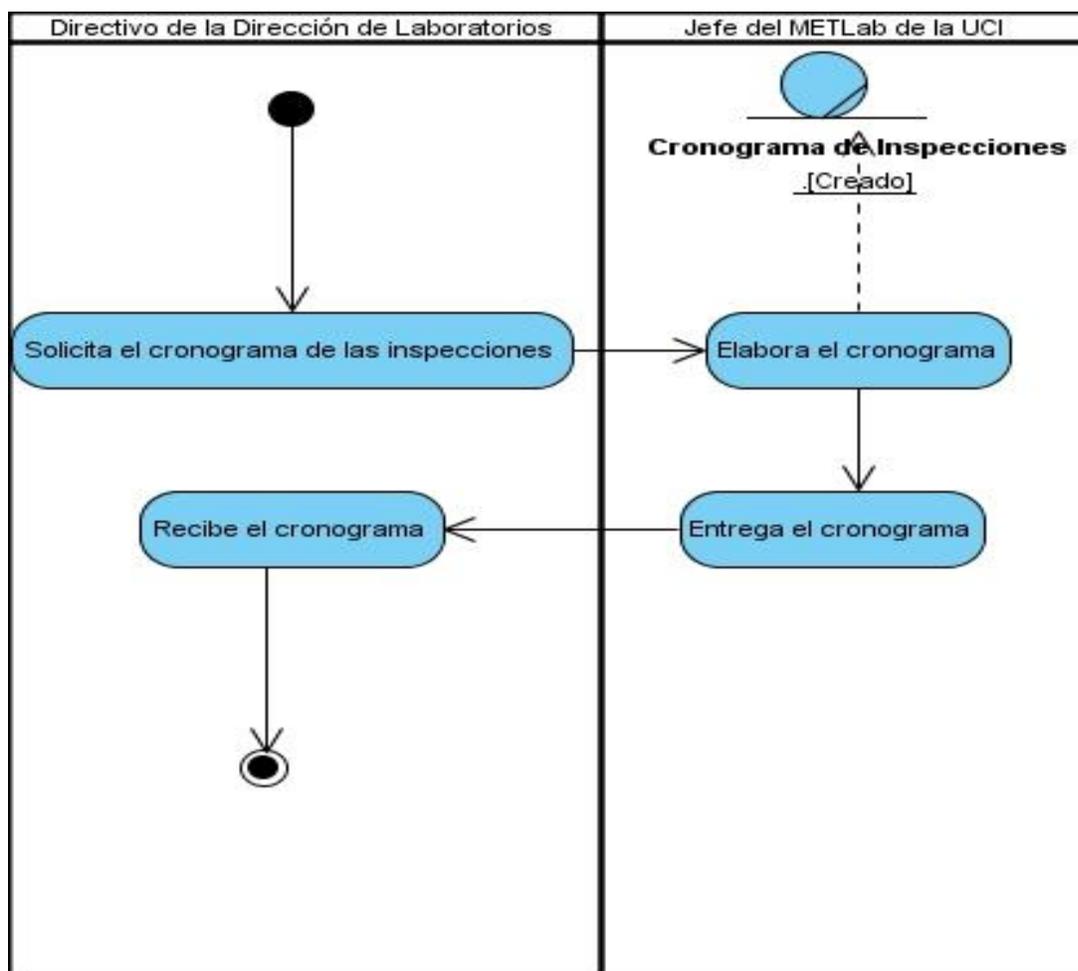
<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
1. Solicita que le reciban el laboratorio. 2. Llena el modelo del parte. 4. Realiza la entrega del laboratorio.	3. El Técnico revisa el laboratorio verificando que todo coincida con el modelo del parte. 5. Firma el modelo del parte y recibe el laboratorio.

#### Flujos Alternos

<b>Acción de Actor</b>	<b>Respuesta del Negocio</b>
5. El Estudiante Técnico de Laboratorio revisa el laboratorio y arregla el modelo del parte.	1. No coincide lo que está en el modelo del parte con lo que está físicamente en el laboratorio.

6. Realiza la entrega el laboratorio	7. Firma el modelo del parte y recibe el laboratorio.
<b>Poscondiciones</b>	Modelo del parte.
<b>Mejoras</b>	Informatizar este proceso permitirá gestionar la entrega de los laboratorios y tener control de las pérdidas en los mismos.

**1.5. Diagrama de Actividades de los Casos de Usos del Negocio.**



**Figura 3 Diagrama de Actividades del Caso de Uso del Negocio “Elaboración del cronograma de las inspecciones”.**

Los diagramas de actividades de los casos de usos del negocio se encuentran en los Anexos del 1 al 7.

1.6. Diagrama de clases del modelo de objetos.

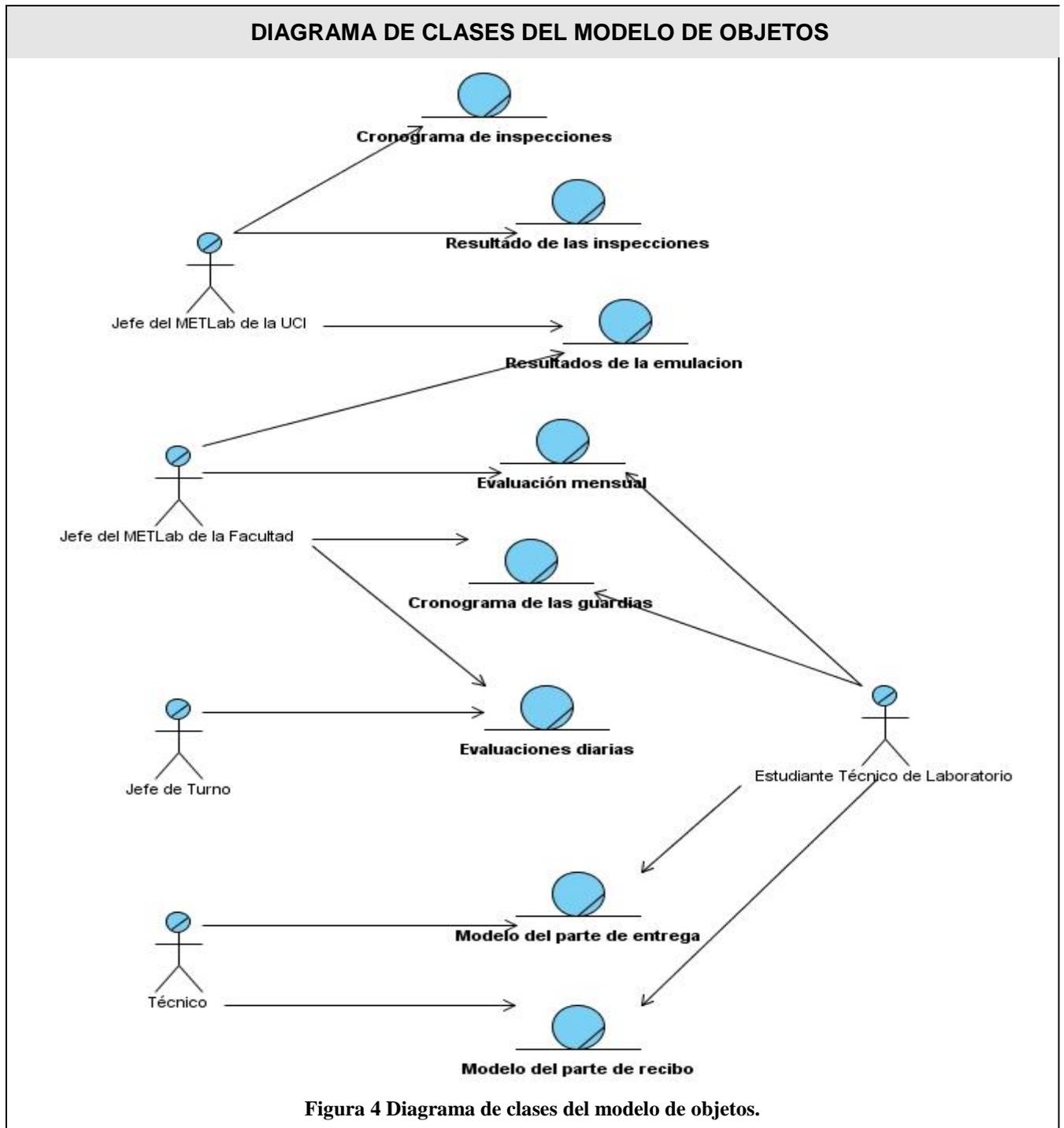


Figura 4 Diagrama de clases del modelo de objetos.

## **2. Especificación de los requisitos de software.**

### **2.1. Requisitos Funcionales**

- R1. Autenticar usuarios.
- R2. Gestionar roles.
  - 1. Asignar rol.
  - 2. Modificar rol.
  - 3. Eliminar rol.
- R3. Gestionar estudiantes
  - 1. Insertar estudiantes.
  - 2. Modificar estudiantes.
  - 3. Eliminar estudiantes.
- R4. Gestionar guardia.
  - 1. Insertar guardia.
  - 2. Modificar guardia.
  - 3. Eliminar guardia.
- R5. Gestionar evaluación diaria.
  - 1. Insertar evaluación diaria.
  - 2. Modificar evaluación diaria.
  - 3. Eliminar evaluación diaria.
- R6. Gestionar evaluación mensual.
  - 1. Insertar evaluación mensual.
  - 2. Modificar evaluación mensual.
  - 3. Eliminar evaluación mensual.
- R7. Gestionar inspecciones.
  - 1. Insertar inspección.
  - 2. Modificar inspección.
  - 3. Eliminar inspección.
- R8. Gestionar resultados de las inspecciones.
  - 1. Insertar resultados de las inspecciones.
  - 2. Modificar resultados de las inspecciones.
  - 3. Eliminar resultados de las inspecciones.
- R9. Gestionar los laboratorios docentes.
  - 1. Insertar laboratorios.
  - 2. Modificar laboratorios.
  - 3. Eliminar laboratorios.
- R10. Gestionar los resultados de la emulación.
  - 1. Insertar resultados de la emulación.
  - 2. Modificar resultados de la emulación.
  - 3. Eliminar resultados de la emulación.
- R11. Gestionar la entrega del laboratorio.
  - 1. Insertar entrega.
  - 2. Modificar entrega.
  - 3. Eliminar entrega.
- R12. Gestionar el recibo del laboratorio.
  - 1. Insertar recibo.
  - 2. Modificar recibo.

3. Eliminar recibo.

R13. Generación de reportes.

1. Mostrar todos los estudiantes de una facultad.
2. Mostrar estudiantes por año.
3. Mostrar estudiante por solapín.
4. Mostrar estudiante por usuario.
5. Mostrar estudiante por año y facultad.
6. Mostrar guardias por fecha.
7. Mostrar guardias por laboratorios y facultad.
8. Mostrar guardias de un estudiante.
9. Mostrar guardia por mes.
10. Mostrar guardia por laboratorio, fecha y facultad.
11. Mostrar guardia por mes y facultad.
12. Mostrar guardia por mes, facultad y laboratorio.
13. Mostrar todos los estudiantes.
14. Mostrar guardia por fecha y facultad.
15. Mostrar guardia por estudiante y mes.
16. Mostrar las evaluaciones diarias de un estudiante.
17. Mostrar la evaluación de un estudiante por fecha.
18. Mostrar las evaluaciones diarias de todos los estudiantes de una facultad en un mes.
19. Mostrar las evaluaciones mensuales de todos los estudiantes de una facultad.
20. Mostrar las evaluaciones mensuales de todos los estudiantes de una facultad en un mes.
21. Mostrar el cronograma de las inspecciones en un mes.
22. Mostrar los resultados de las inspecciones realizadas a una facultad en un mes.
23. Mostrar los resultados de una inspección realizada a una facultad.
24. Mostrar los resultados de la emulación en un mes.
25. Mostrar los resultados de la emulación de una facultad en un mes.
26. Mostrar todos los laboratorios de un docente.
27. Mostrar todos los laboratorios de una facultad.
28. Mostrar todas las entregas de guardia de una facultad en una fecha.
29. Mostrar todas las entregas de laboratorio de un estudiante.
30. Mostrar todas las entregas de laboratorio de un técnico.
31. Mostrar todos los recibos de laboratorio de una facultad en una fecha.
32. Mostrar todos los recibos de laboratorio de un estudiante.

R14. Imprimir todos los listados.

## **2.2 Requisitos no funcionales**

### **Requerimientos de software**

- Servidor web Apache versión 1.3 o superior.
- PHP versión 5.x.
- Servidor de Bases de Datos PostgreSQL.
- CMS Drupal 6.13 o superior.
- Sistema Operativo Windows XP o superior o Linux.
- Navegador web Internet Explorer versión 6.0 o superior, Mozilla Firefox versión 1.5 o superior, Google Chrome 4.0 o superior.

## Requerimientos de hardware

- **Requerimientos mínimos para la máquina servidora:**
  - ✓ Procesador de tipo Pentium III con velocidad de 1 GHz o superior.
  - ✓ Memoria RAM de 512 MB o superior.
  - ✓ Disco duro de 40 GB o superior.
  - ✓ Tarjeta de red o módem.
- **Requerimientos mínimos para la estación de trabajo del usuario:**
  - ✓ Navegador web Internet Explorer versión 6.0 o superior, Mozilla Firefox versión 1.5 o superior, Google Chrome 4.0 o superior.
  - ✓ Tarjeta de red o módem.

## Requerimientos de apariencia o interfaz externa

- Interfaz amigable con facilidades para su uso por usuarios de cualquier nivel, reduciendo el tiempo de entrenamiento a los mismos.
- Funcionalidades visibles en todo momento que faciliten la navegación.

## Requerimientos de seguridad

- Sistema de permisos y usuarios para el acceso a la información, para protegerla de accesos no autorizados y divulgación.
- La autenticación se realizará por los usuarios del Dominio UCI.
- Validación de datos en el cliente, para evitar estados inconsistentes en la información.

## Requerimientos de usabilidad

- Familiarización con el entorno web.
- Diseño adaptable a diferentes resoluciones de pantalla.
- Diseño orientado a la simplicidad, con mínimas distracciones y una clara arquitectura de la información.

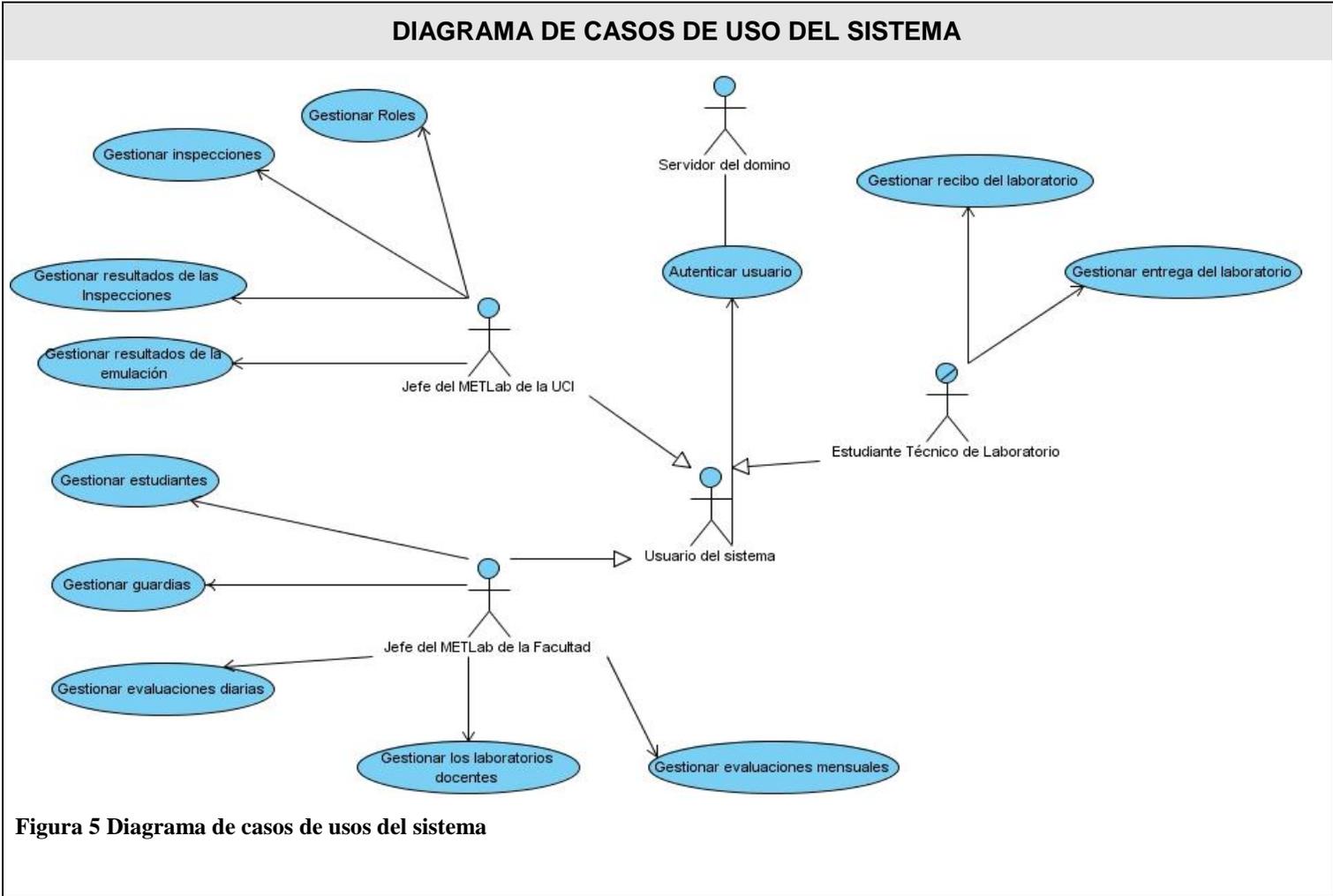
## 3. Definición de los casos de uso del sistema.

### 3.1. Definición de los actores del sistema.

Actores	Justificación
Jefe del METLab de la UCI	Es el encargado de gestionar las inspecciones, los resultados de las mismas y de la emulación, así como gestionar los roles en el sistema. Además es el encargado de asignar privilegios a los usuarios que van a interactuar con el sistema.
Jefe del METLab de la Facultad	Es el encargado de la gestión de los estudiantes que pertenecen al Movimiento, así como las guardias, las evaluaciones diarias y mensuales y los laboratorios docentes de la facultad.
Estudiante Técnico de Laboratorio	Es el encargado de la gestión del recibo y entrega del laboratorio durante la guardia.

Usuario del sistema	Es quien pretende 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 del dominio	Es el responsable de realizar la validación de la contraseña del usuario perteneciente al dominio UCI.

### 3.2. Diagrama de casos de uso del sistema.



### 3.3. Casos de uso expandidos.

<b>Caso de Uso</b>	Autenticar usuario	
<b>Actor</b>	Usuario del sistema	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario intenta acceder al sistema. Se comprueba su autenticidad y recibe los permisos según el rol que tiene asignado en el sistema. Solo usuarios autenticados podrán interactuar con el sistema. El caso de uso termina cuando se le han dado los privilegios correspondientes al usuario y este accede al sistema.	
<b>Precondiciones:</b>	El usuario debe tener privilegios para interactuar con el sistema.	
<b>Flujo Normal de Eventos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
1. Introduce nombre de usuario y contraseña.	2. Comprueba validez de los datos. 3. Muestra la página de inicio según los privilegios del usuario autenticado	
<b>Flujos Alternos</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
	3. Si los datos no son válidos muestra un mensaje de error.	
<b>Referencia</b>	RF1	

<b>Caso de Uso</b>	Gestionar roles	
<b>Actor</b>	Jefe del METLab de la UCI	
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI solicita crear (Sección A), modificar (Sección B) o eliminar (Sección C) un rol. Si decide crear un nuevo rol, introduce los datos para dicho rol. Si decide modificar los datos de un rol ya existente, obtiene la información almacenada de ese rol y la modifica. Si decide eliminar el rol, selecciona el rol deseado y lo elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.	
<b>Precondiciones:</b>		

<b>Flujo Normal de Eventos</b>	
<b>Sección A “Crear un nuevo rol”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a crear un nuevo rol.	2. Muestra el formulario para crear el nuevo rol.
3. Introduce los datos para el nuevo rol.	4. Guardar el rol. 5. Muestra el listado de roles actualizados
<b>Precondiciones</b>	El rol a modificar se debe haber creado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar un rol”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de un rol previamente creado.	2. Muestra el listado de roles existentes.
3. Selecciona el rol a modificar.	4. Muestra la información del rol seleccionado.
5. Modifica los datos del rol.	6. Guarda los cambios efectuados al rol. 7. Muestra la información del rol actualizado.
<b>Precondiciones</b>	El rol a eliminar se debe haber creado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar un rol”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar un rol previamente creado.	2. Muestra el listado de roles existentes.
3. Selecciona el rol a eliminar.	

5. Elige una opción.	<p>4. Muestra un mensaje de confirmación para la acción.</p> <p>6. Elimina el rol de la base de datos.</p> <p>7. Muestra el listado de roles actualizados.</p>
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar el rol, la acción no se ejecuta.
<b>Referencia</b>	RF2 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar estudiantes
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) un estudiante. Si decide insertar un nuevo estudiante, busca el solapín del estudiante y muestra los datos de dicho estudiante y verifica que éste no esté insertado ya en la base de datos. Si decide modificar los datos de un estudiante ya existente, obtiene la información almacenada del mismo y la modifica. Si decide eliminar el estudiante, selecciona el mismo y lo elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar un nuevo estudiante”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Accede a insertar un nuevo estudiante.</p> <p>4. Rellena los datos del nuevo estudiante.</p>	<p>2. Muestra el formulario para buscar el solapín del estudiante</p> <p>3. Muestra los datos del estudiante del solapín insertado.</p> <p>5. Verifica que no se encuentre ya en la base de datos.</p> <p>6. Guarda los datos del estudiante.</p>

	7. Muestra el listado de estudiantes actualizados.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	3. Si el solapín del estudiante no se encuentra muestra un mensaje de error
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si el estudiante ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	El estudiante a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar un estudiante”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de un estudiante previamente insertado.  3. Selecciona el estudiante a modificar.  5. Modifica los datos del estudiante.	2. Muestra el listado de estudiantes existentes.  4. Muestra la información del estudiante seleccionado.  6. Guarda los cambios efectuados al estudiante. 7. Muestra la información del estudiante actualizado.
<b>Precondiciones</b>	El estudiante a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar un estudiante”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>

1. Accede a eliminar un estudiante previamente insertado.	2. Muestra el listado de estudiantes existentes.
3. Selecciona el estudiante a eliminar.	4. Muestra un mensaje de confirmación para la acción.
5. Elige una opción.	6. Elimina el estudiante de la base de datos. 7. Muestra el listado de estudiantes actualizados.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	7. Si el usuario escoge no eliminar el estudiante, la acción no se ejecuta.
<b>Referencia</b>	RF3 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar guardia
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) una guardia. Si decide insertar una nueva guardia, introduce los datos de la guardia y verifica que ésta no esté insertada ya en la base de datos. Si decide modificar los datos de una guardia ya existente, obtiene la información almacenada de la misma y la modifica. Si decide eliminar una guardia, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar una nueva guardia”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar una nueva	2. Muestra el formulario para insertar la nueva guardia.

guardia.	
3. Introduce los datos de la nueva guardia.	4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de la guardia. 6. Muestra el listado de guardias actualizados.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	5. Si la guardia ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	La guardia a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar una guardia”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de una guardia previamente insertada.  3. Selecciona la guardia a modificar.  5. Modifica los datos de la guardia.	2. Muestra el listado de las guardias existentes.  4. Muestra la información de la guardia seleccionada.  6. Guarda los cambios efectuados en la guardia. 7. Muestra la información de la guardia actualizada.
<b>Precondiciones</b>	La guardia a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar una guardia”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar una guardia previamente	2. Muestra el listado de las guardias existentes.

insertada.	
3. Selecciona la guardia a eliminar.	4. Muestra un mensaje de confirmación para la acción.
5. Elige una opción.	6. Elimina la guardia de la base de datos. 7. Muestra el listado de las guardias actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar la guardia, la acción no se ejecuta.
<b>Referencia</b>	RF4 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar evaluación diaria
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) una evaluación diaria. Si decide insertar una nueva evaluación diaria, introduce los datos de la misma y verifica que ésta no esté insertada ya en la base de datos. Si decide modificar los datos de una evaluación diaria ya existente, obtiene la información almacenada de la misma y la modifica. Si decide eliminar una evaluación diaria, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar una nueva evaluación diaria”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar una nueva evaluación diaria.	2. Muestra el formulario para insertar la nueva evaluación diaria.

3. Introduce los datos de la nueva evaluación diaria.	4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de la evaluación diaria. 6. Muestra el listado de evaluaciones diarias actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si la evaluación diaria ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	La evaluación diaria a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar una evaluación diaria”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de una evaluación diaria previamente insertada.  3. Selecciona la evaluación diaria a modificar.  5. Modifica los datos de la evaluación diaria.	2. Muestra el listado de las evaluaciones diarias existentes.  4. Muestra la información de la evaluación diaria seleccionada.  6. Guarda los cambios efectuados en la evaluación diaria. 7. Muestra la información de la evaluación diaria actualizada.
<b>Precondiciones</b>	La evaluación diaria a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar una evaluación diaria”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>

1. Accede a eliminar una evaluación diaria previamente insertada.	2. Muestra el listado de las evaluaciones diarias existentes.
3. Selecciona la evaluación diaria a eliminar.	4. Muestra un mensaje de confirmación para la acción.
5. Elige una opción.	6. Elimina la evaluación diaria de la base de datos.
	7. Muestra el listado de las evaluaciones diarias actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar la evaluación diaria, la acción no se ejecuta.
<b>Referencia</b>	RF5 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar evaluación mensual
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) una evaluación mensual. Si decide insertar una nueva evaluación mensual, introduce los datos de la misma y verifica que ésta no esté insertada ya en la base de datos. Si decide modificar los datos de una evaluación mensual ya existente, obtiene la información almacenada de la misma y la modifica. Si decide eliminar una evaluación mensual, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar una nueva evaluación mensual”</b>	

<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar una nueva evaluación mensual.  3. Introduce los datos de la nueva evaluación mensual.	2. Muestra el formulario para insertar la nueva evaluación mensual.  4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de la evaluación mensual. 6. Muestra el listado de evaluaciones mensuales actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si la evaluación mensual ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	La evaluación mensual a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar una evaluación mensual”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de una evaluación mensual previamente insertada.  3. Selecciona la evaluación mensual a modificar.  5. Modifica los datos de la evaluación mensual.	2. Muestra el listado de las evaluaciones mensuales existentes.  4. Muestra la información de la evaluación mensual seleccionada.  6. Guarda los cambios efectuados en la evaluación mensual. 7. Muestra la información de la evaluación mensual actualizada.
<b>Precondiciones</b>	La evaluación mensual a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	

<b>Sección C “Eliminar una evaluación mensual”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar una evaluación mensual previamente insertada.  3. Selecciona la evaluación mensual a eliminar.  5. Elige una opción.	2. Muestra el listado de las evaluaciones mensuales existentes.  4. Muestra un mensaje de confirmación para la acción.  6. Elimina la evaluación mensual de la base de datos. 7. Muestra el listado de las evaluaciones mensuales actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar la evaluación mensual, la acción no se ejecuta.
<b>Referencia</b>	RF6 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar inspecciones
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) una inspección. Si decide insertar una nueva inspección, introduce los datos de la misma y verifica que ésta no esté insertada ya en la base de datos. Si decide modificar los datos de una inspección ya existente, obtiene la información almacenada de la misma y la modifica. Si decide eliminar una inspección, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	

<b>Sección A “Insertar una nueva inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar una nueva inspección.  3. Introduce los datos de la nueva inspección.	2. Muestra el formulario para insertar la nueva inspección.  4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de la inspección. 6. Muestra el listado de las inspecciones actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si la inspección ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	La inspección a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar una inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de una inspección previamente insertada.  3. Selecciona la inspección a modificar.  5. Modifica los datos de la inspección.	2. Muestra el listado de las inspecciones existentes.  4. Muestra la información de la inspección seleccionada.  6. Guarda los cambios efectuados en la inspección. 7. Muestra la información de la inspección actualizada.
<b>Precondiciones</b>	La inspección a eliminar se debe haber insertado con anterioridad.

<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar una inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar una inspección previamente insertada.  3. Selecciona la inspección a eliminar.  5. Elige una opción.	2. Muestra el listado de las inspecciones existentes.  4. Muestra un mensaje de confirmación para la acción.  6. Elimina la inspección de la base de datos. 7. Muestra el listado de las inspecciones actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar la inspección, la acción no se ejecuta.
<b>Referencia</b>	RF7 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar resultados de las inspecciones
<b>Actor</b>	Jefe del METLab de la UCI
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) los resultados de las inspecciones. Si decide insertar nuevos resultados de las inspecciones, introduce los datos y verifica que estos no estén insertados ya en la base de datos. Si decide modificar los datos de los resultados de una inspección ya existente, obtiene la información almacenada y la modifica. Si decide eliminar los resultados de una inspección, selecciona los mismos y los elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.

<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar los resultados de una inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar los resultados de una inspección.  3. Introduce los datos de los resultados de la inspección.	2. Muestra el formulario para insertar los resultados de la inspección.  4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de los resultados de la inspección. 6. Muestra el listado de los resultados de las inspecciones actualizado.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si los resultados de esa inspección ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	Los resultados de la inspección a modificar se deben haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar los resultados de una inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de los resultados de una inspección previamente insertada.  3. Selecciona los resultados de la inspección a modificar.  5. Modifica los datos de los	2. Muestra el listado de los resultados de las inspecciones existentes.  4. Muestra la información de los resultados de la inspección seleccionada.  6. Guarda los cambios efectuados en los resultados de la

resultados de la inspección.	inspección. 7. Muestra la información de los resultados de la inspección actualizada.
<b>Precondiciones</b>	Los resultados de la inspección a eliminar se deben haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar los resultados de una inspección”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar los resultados de una inspección previamente insertada.  3. Selecciona los resultados de la inspección a eliminar.  5. Elige una opción.	2. Muestra el listado de los resultados de las inspecciones existentes.  4. Muestra un mensaje de confirmación para la acción.  6. Elimina los resultados de la inspección de la base de datos. 7. Muestra el listado de los resultados de las inspecciones actualizadas.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	7. Si el usuario escoge no eliminar los resultados de la inspección, la acción no se ejecuta.
<b>Referencia</b>	RF8 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar laboratorios docentes
<b>Actor</b>	Jefe del METLab de la Facultad
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la Facultad solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) un laboratorio docente. Si decide insertar un nuevo laboratorio docente,

	introduce los datos y verifica que éste no esté insertado ya en la base de datos. Si decide modificar los datos de un laboratorio docente ya existente, obtiene la información almacenada de la misma y la modifica. Si decide eliminar un laboratorio docente, selecciona el mismo y lo elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección A “Insertar un laboratorio docente”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a insertar un nuevo laboratorio docente.  3. Introduce los datos del laboratorio docente.	2. Muestra el formulario para insertar el laboratorio docente.  4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos del laboratorio docente. 6. Muestra el listado de los laboratorios docentes actualizado.
<b>Flujos Alternos</b>	
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>
	4. Si el laboratorio docente ya se encuentra en la base de datos muestra un mensaje de error.
<b>Precondiciones</b>	El laboratorio docente a modificar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección B “Modificar un laboratorio docente”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de un laboratorio docente previamente insertado.  3. Selecciona el laboratorio	2. Muestra el listado de los laboratorios docentes existentes.

docente a modificar. 5. Modifica los datos del laboratorio docente.	4. Muestra la información del laboratorio docente seleccionado. 6. Guarda los cambios efectuados en el laboratorio docente. 7. Muestra la información de los laboratorios docentes actualizada.
<b>Precondiciones</b>	El laboratorio docente a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar un laboratorio docente”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar un laboratorio docente previamente insertado. 3. Selecciona el laboratorio docente a eliminar. 5. Elige una opción.	2. Muestra el listado de los laboratorios docentes existentes. 4. Muestra un mensaje de confirmación para la acción. 6. Elimina el laboratorio docente de la base de datos. 7. Muestra el listado de los laboratorios docentes actualizados.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar el laboratorio docente, la acción no se ejecuta.
<b>Referencia</b>	RF9 (1, 2, 3)

<b>Caso de Uso</b>	Gestionar resultados de la emulación
<b>Actor</b>	Jefe del METLab de la UCI

<b>Resumen</b>	El caso de uso se inicia cuando el Jefe del METLab de la UCI solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) los resultados de la emulación. Si decide insertar nuevos resultados de la emulación, introduce los datos y verifica que estos no estén insertados ya en la base de datos. Si decide modificar los datos de los resultados de la emulación ya existente, obtiene la información almacenada y la modifica. Si decide eliminar los resultados de la emulación, selecciona los mismos y los elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Sección A “Insertar los resultados de la emulación”</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
1. Accede a insertar los resultados de la emulación.	2. Muestra el formulario para insertar los resultados de la emulación.	
3. Introduce los datos de los resultados de la emulación.	4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de los resultados de la emulación. 6. Muestra el listado de los resultados de la emulación actualizado.	
<b>Flujos Alternos</b>		
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>	
	4. Si los resultados de la emulación ya se encuentran en la base de datos muestra un mensaje de error.	
<b>Precondiciones</b>	Los resultados de la emulación a modificar se deben haber insertado con anterioridad.	
<b>Flujo Normal de Eventos</b>		
<b>Sección B “Modificar los resultados de la emulación”</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
1. Accede a modificar los datos de los resultados de la	2. Muestra el listado de los resultados de la emulación.	

<p>emulación previamente insertada.</p> <p>3. Selecciona los resultados de la emulación a modificar.</p> <p>5. Modifica los datos de los resultados de la emulación.</p>	<p>4. Muestra la información de los resultados de la emulación seleccionada.</p> <p>6. Guarda los cambios efectuados en los resultados de la emulación.</p> <p>7. Muestra la información de los resultados de la emulación actualizada.</p>
<b>Precondiciones</b>	Los resultados de la inspección a eliminar se deben haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar los resultados de la emulación”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
<p>1. Accede a eliminar los resultados de la emulación previamente insertada.</p> <p>3. Selecciona los resultados de la emulación a eliminar.</p> <p>5. Elige una opción.</p>	<p>2. Muestra el listado de los resultados de la emulación existentes.</p> <p>4. Muestra un mensaje de confirmación para la acción.</p> <p>6. Elimina los resultados de la emulación de la base de datos.</p> <p>7. Muestra el listado de los resultados de la emulación actualizadas.</p>
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	<p>6. Si el usuario escoge no eliminar los resultados de la emulación, la acción no se ejecuta.</p>

<b>Referencia</b>	RF10 (1, 2, 3)
-------------------	----------------

<b>Caso de Uso</b>	Gestionar entrega del laboratorio	
<b>Actor</b>	Estudiante Técnico de Laboratorio	
<b>Resumen</b>	El caso de uso se inicia cuando el Estudiante Técnico de Laboratorio solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) los la entrega del laboratorio. Si decide insertar una nueva entrega del laboratorio, introduce los datos y verifica que estos no estén insertados ya en la base de datos. Si decide modificar los datos de una entrega de laboratorio ya existente, obtiene la información almacenada y la modifica. Si decide eliminar una entrega de laboratorio, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Sección A “Insertar entrega del laboratorio”</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
1. Accede a insertar la entrega del laboratorio.  3. Introduce los datos de la entrega del laboratorio.	2. Muestra el formulario para insertar la entrega del laboratorio.  4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos de la entrega del laboratorio.	
<b>Flujos Alternos</b>		
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>	
	5. Si la entrega del laboratorio ya se encuentra en la base de datos muestra un mensaje de error.	
<b>Precondiciones</b>	La entrega del laboratorio a modificar se debe haber insertado con anterioridad.	
<b>Flujo Normal de Eventos</b>		

<b>Sección B “Modificar una entrega de laboratorio”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos de la entrega del laboratorio previamente insertada.  3. Selecciona la entrega de laboratorio a modificar.  5. Modifica los datos de la entrega del laboratorio.	2. Muestra el listado de la entrega del laboratorio.  4. Muestra la información de la entrega del laboratorio seleccionada.  6. Guarda los cambios efectuados en la entrega del laboratorio. 7. Muestra la información de la entrega del laboratorio actualizada.
<b>Precondiciones</b>	La entrega de laboratorio a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar una entrega de laboratorio”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar la entrega de laboratorio previamente insertada.  3. Selecciona la entrega de laboratorio a eliminar.  5. Elige una opción.	2. Muestra el listado de la entrega de laboratorios existentes.  4. Muestra un mensaje de confirmación para la acción.  6. Elimina la entrega de laboratorio de la base de datos.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar la entrega del laboratorio, la acción no se ejecuta.

<b>Referencia</b>	RF11 (1, 2, 3)
-------------------	----------------

<b>Caso de Uso</b>	Gestionar recibo del laboratorio	
<b>Actor</b>	Estudiante Técnico de Laboratorio	
<b>Resumen</b>	El caso de uso se inicia cuando el Estudiante Técnico de Laboratorio solicita insertar (Sección A), modificar (Sección B) o eliminar (Sección C) los la entrega del laboratorio. Si decide insertar un nuevo recibo del laboratorio, introduce los datos y verifica que estos no estén insertados ya en la base de datos. Si decide modificar los datos de un recibo del laboratorio ya existente, obtiene la información almacenada y la modifica. Si decide eliminar un recibo del laboratorio, selecciona la misma y la elimina. En todos los casos el caso de uso termina cuando se guardan los cambios en la base de datos.	
<b>Precondiciones:</b>		
<b>Flujo Normal de Eventos</b>		
<b>Sección A “Insertar recibo del laboratorio”</b>		
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>	
1. Accede a insertar el recibo del laboratorio.	2. Muestra el formulario para insertar el recibo del laboratorio.	
3. Introduce los datos del recibo del laboratorio.	4. Verifica que no se encuentre ya en la base de datos. 5. Guarda los datos del recibo del laboratorio.	
<b>Flujos Alternos</b>		
<b>Acción de Autor</b>	<b>Respuesta del Sistema</b>	
	5. Si el recibo del laboratorio ya se encuentra en la base de datos muestra un mensaje de error.	
<b>Precondiciones</b>	El recibo del laboratorio a modificar se debe haber insertado con anterioridad.	
<b>Flujo Normal de Eventos</b>		

<b>Sección B “Modificar un recibo del laboratorio”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a modificar los datos del recibo del laboratorio previamente insertado.  3. Selecciona el recibo del laboratorio a modificar.  5. Modifica los datos del recibo del laboratorio.	2. Muestra el listado del recibo del laboratorio.  4. Muestra la información del recibo del laboratorio seleccionado.  6. Guarda los cambios efectuados en el recibo del laboratorio. 7. Muestra la información del recibo del laboratorio actualizada.
<b>Precondiciones</b>	El recibo de laboratorio a eliminar se debe haber insertado con anterioridad.
<b>Flujo Normal de Eventos</b>	
<b>Sección C “Eliminar un recibo del laboratorio”</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
1. Accede a eliminar el recibo del laboratorio previamente insertado.  3. Selecciona el recibo del laboratorio a eliminar.  5. Elige una opción.	2. Muestra el listado del recibo del laboratorio existente.  4. Muestra un mensaje de confirmación para la acción.  6. Elimina el recibo del laboratorio de la base de datos.
<b>Flujos Alternos</b>	
<b>Acción de Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario escoge no eliminar el recibo del laboratorio, la acción no se ejecuta.
<b>Referencia</b>	RF12 (1, 2, 3)

## CAPÍTULO 4: CONSTRUCCIÓN DEL SISTEMA

### 1. Definición del modelo de análisis.

#### 1.1. Modelo de clases de análisis.

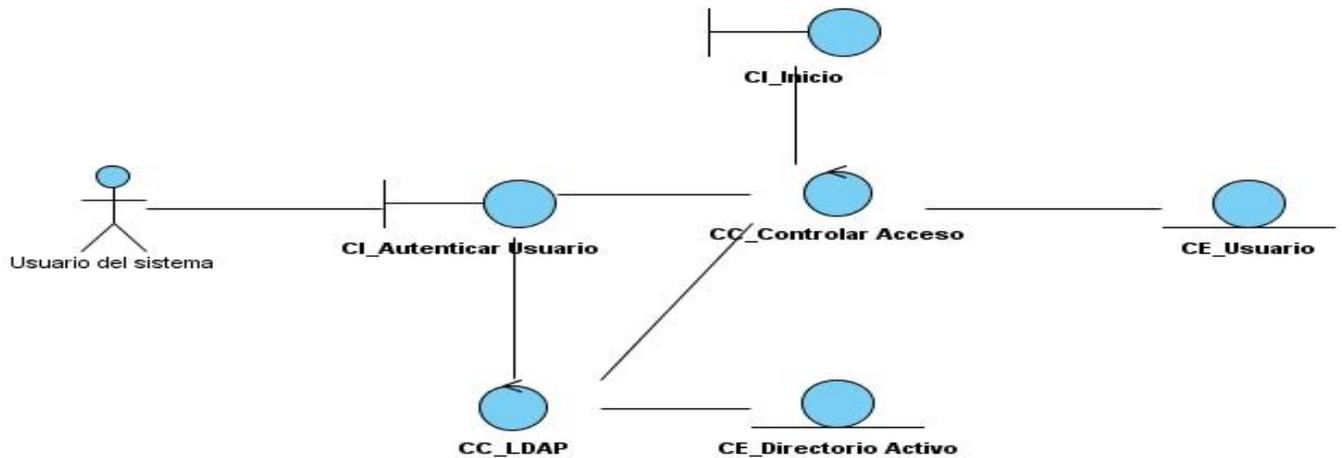


Figura 6 Diagrama de clases del análisis para el caso de uso "Autenticar usuario".

### 2. Diseño.

#### 2.1. Diagramas de colaboración.

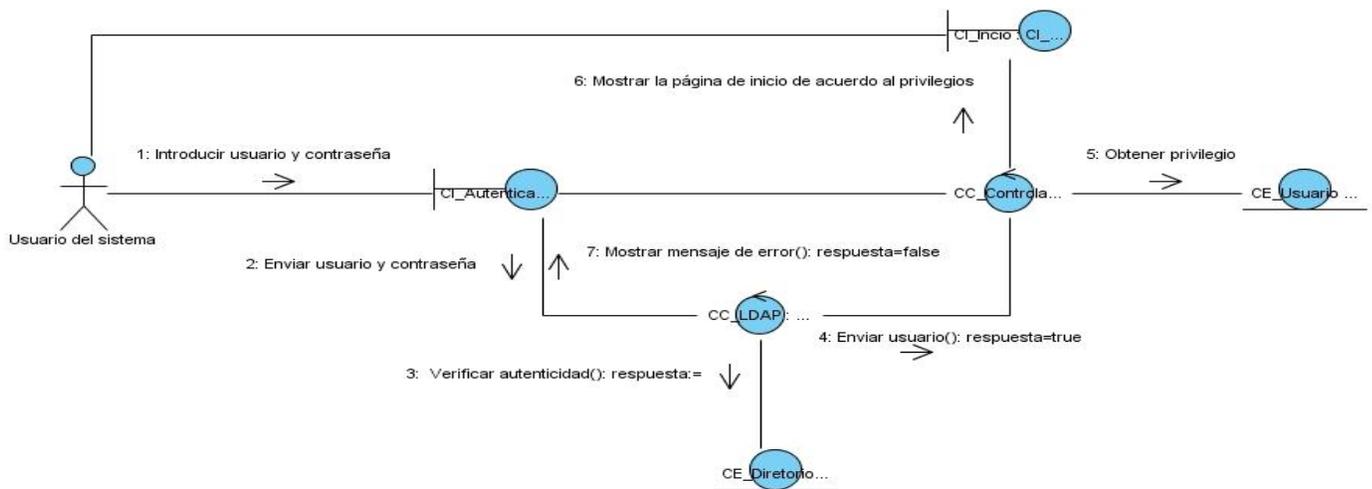


Figura 7 Diagrama de colaboración "Autenticar usuario".

El resto de los diagramas de colaboración se encuentran en los Anexos desde el 8 hasta el 41.

### 3. Diseño de la Base de Datos

#### 3.1. Diagrama Entidad Relación de la Base de Datos.

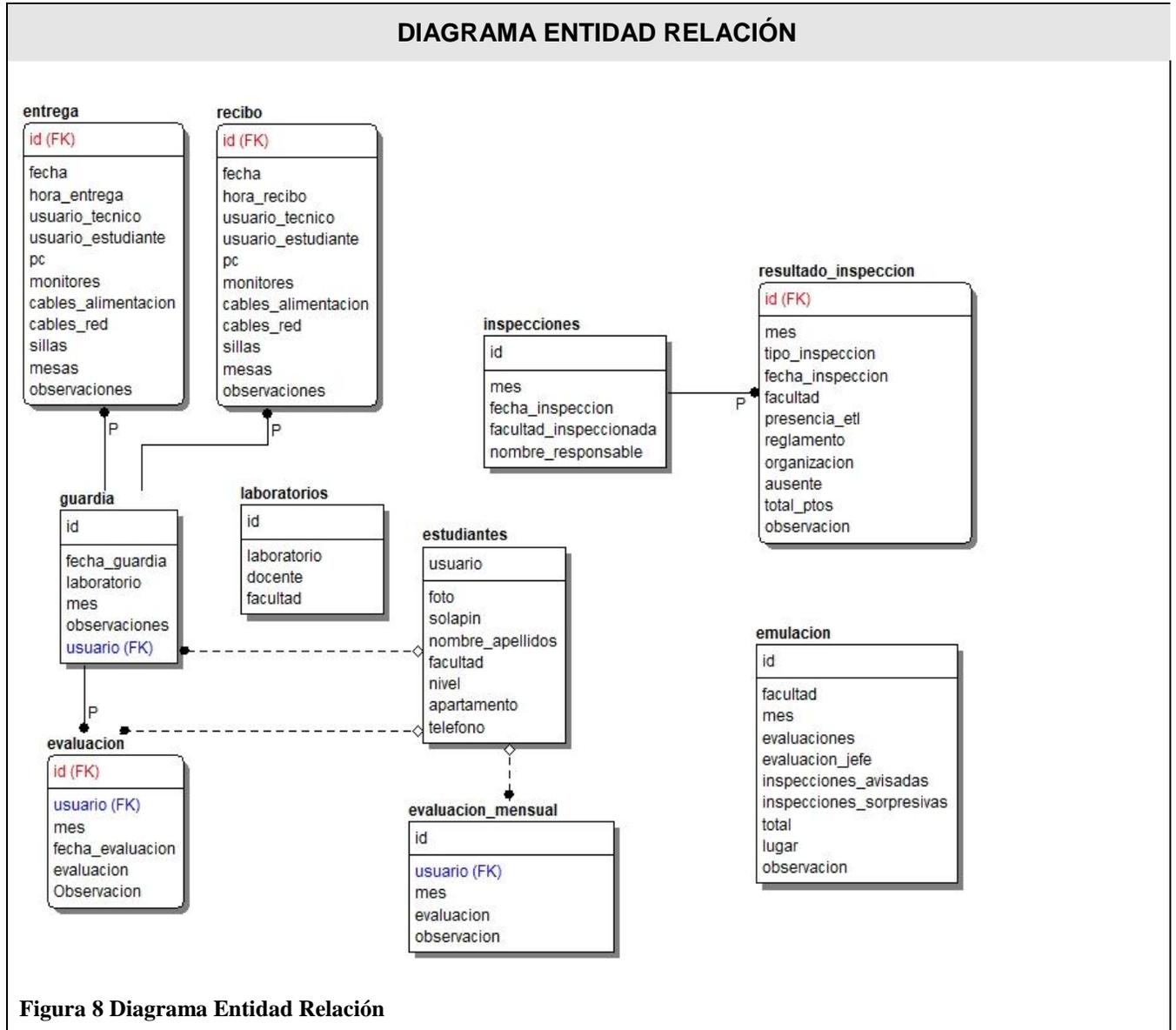
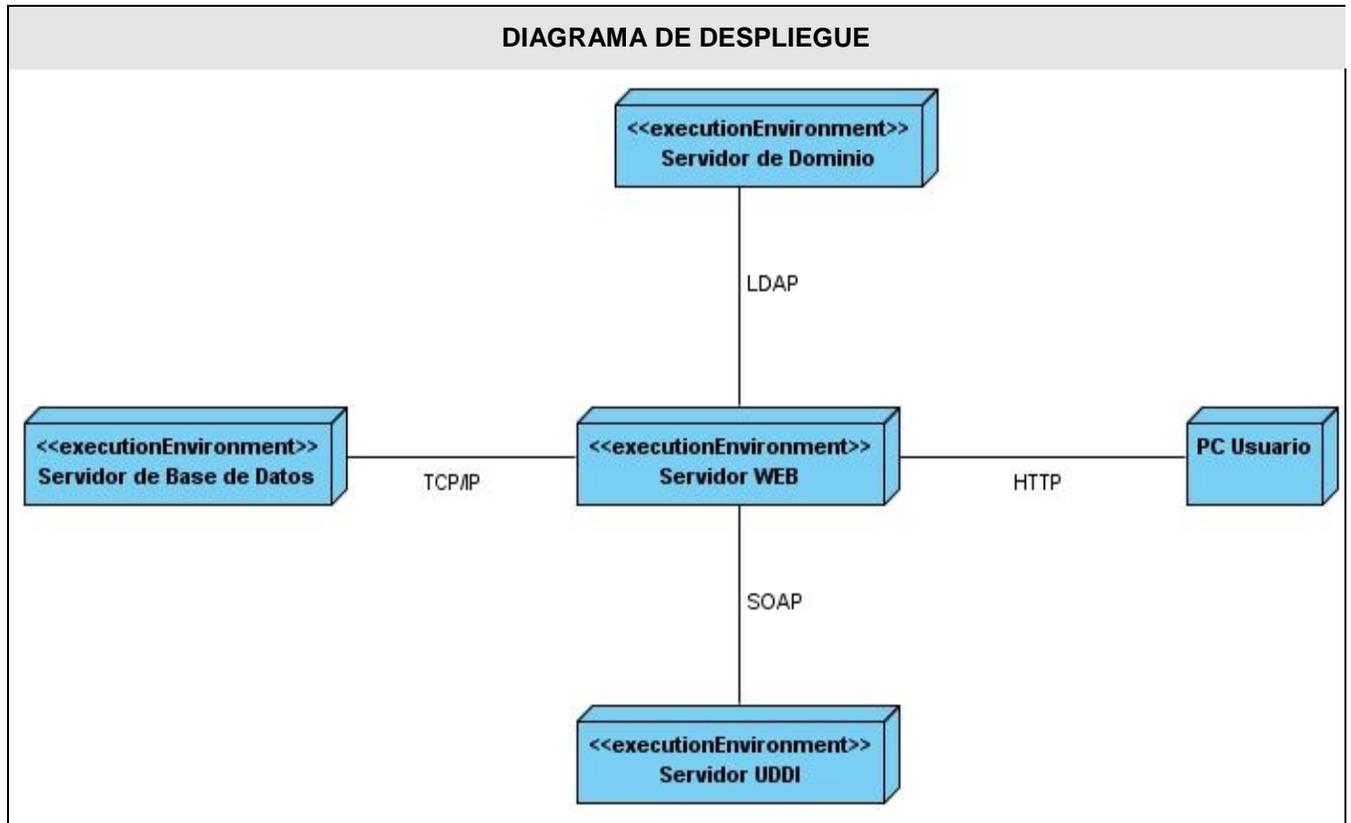


Figura 8 Diagrama Entidad Relación

#### 4. Implementación.

##### 4.1. Diagrama de despliegue.



## CONCLUSIONES

Culminando el desarrollo del sistema para la gestión del funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios de la Universidad de las Ciencias Informáticas y la evaluación de cada uno de sus miembros, se ha llegado a la siguiente conclusión:

- Se detectaron deficiencias en el proceso de evaluación, así como la ausencia de indicadores para evaluar a los estudiantes del Movimiento.
- Se creó un sistema de indicadores para el proceso de evaluación.
- Se obtuvo una aplicación web con la cual automatizar todo el funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios de la UCI.
- Se escogieron las tecnologías propicias teniendo en cuenta las tendencias actuales y escogiendo la más idónea para la solución del problema.

## RECOMENDACIONES

- Estudiar las funcionalidades que se puedan añadir al sistema con vistas a lograr un sistema más completo y con mayores niveles de usabilidad para cada uno de los roles que interactúan.
- Perfeccionar las funcionalidades creadas en el sistema.
- Mantener actualizado el CMS Drupal con las últimas versiones, logrando así una mejor estabilidad y seguridad del mismo.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Bartle, Phil. 2009.** Información para la gestión. *Seattle Community Network*. [En línea] 09 de 09 de 2009. [Citado el: 11 de 02 de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
2. **Eduardo. 2005.** Ventajas de PostgreSQL. *TiendaLinux.com*. [En línea] 31 de 05 de 2005. [Citado el: 18 de 02 de 2010.] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).
3. **Fayol, Henry. 2007.** Control - Administración de Empresas y Negocios. *El Prisma.com - Portal para Investigadores y Profesionales*. [En línea] 2007. [Citado el: 20 de 02 de 2010.] [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/control/](http://www.elprisma.com/apuntes/administracion_de_empresas/control/). 2.
4. **García. 2008.** *Sistema de Información de Gestión de Recursos Humanos*. [En línea] 2008. [Citado el: 20 de 02 de 2010.] <http://190.26.91.148/humanoEL/Ingresar.aspx?Ent=Rionegro>.
5. **González, Elvia. 2009.** Organización de la Información. *Slideshare*. [En línea] 2009. [Citado el: 11 de 02 de 2010.] <http://www.slideshare.net/cuco66/organizacion-de-la-informacion-1144265#>.
6. **Gutián, María Virginia González. 2007.** PROCEDIMIENTO PARA REALIZAR AUDITORIAS DE INFORMACIÓN EN INSTALACIONES HOTELERAS. *eumed.net*. [En línea] 2007. [Citado el: 20 de 02 de 2010.] <http://www.eumed.net/libros/2009d/620/index.htm>. 4.
7. **Illera, Alvaro Marín. 2009.** MySQL. *e-ghost*. [En línea] 2009. [Citado el: 18 de 02 de 2010.] <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html>.
8. **Meléndez, R. 2008.** Un nuevo concepto de evaluación. *Universidad Interamericana de Puerto Rico, Recinto Fajardo*. [En línea] 2008. [Citado el: 11 de 02 de 2010.] [http://fajardo.inter.edu/Resiliencia2parte/Documentos/Un\\_Nuevo\\_Concepto\\_de\\_Evaluacion.pdf](http://fajardo.inter.edu/Resiliencia2parte/Documentos/Un_Nuevo_Concepto_de_Evaluacion.pdf).

9. **Roberto. 2009.** Metodologías de desarrollo del software. *La Tecla de Escape*. [En línea] 25 de 01 de 2009. [Citado el: 21 de 02 de 2010.] <http://latecladeescape.com/w0/ingenieria-del-software/metodologias-de-desarrollo-del-software.html>.
10. **Vidal. 2009.** Embarcadero ER/Studio. *CASE Computación Avanzada y Sistemas Empresariales*. [En línea] 2009. [Citado el: 19 de 02 de 2010.] [http://www.casenet.com.mx/index.php?option=com\\_content&view=article&id=58&Itemid=59](http://www.casenet.com.mx/index.php?option=com_content&view=article&id=58&Itemid=59).

## BIBLIOGRAFÍA

1. *Visual Paradigm*. [En línea] [Citado el: 19 de 02 de 2010.] <http://www.visual-paradigm.com> .
2. *ASSETS: Sistema de Gestión Integral*. [En línea] [Citado el: 20 de 02 de 2010.] <http://assets.co.cu/humanos.asp>.
3. **2007.** 17 Ventajas de Wordpress. *juanantuna*. [En línea] 22 de 08 de 2007. [Citado el: 19 de 02 de 2010.] <http://www.juanantuna.com/2007/08/22/17-ventajas-de-wordpress/>.
4. **Bartle, Phil. 2009.** Información para la gestión. *Seattle Community Network*. [En línea] 09 de 09 de 2009. [Citado el: 11 de 02 de 2010.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
5. Curso de MySQL. *MySQL con Clase*. [En línea] [Citado el: 18 de 02 de 2010.] <http://mysql.conclase.net/curso/>.
6. **Eduardo. 2005.** Ventajas de PostgreSQL. *TiendaLinux.com*. [En línea] 31 de 05 de 2005. [Citado el: 18 de 02 de 2010.] [http://soporte.tiendalinux.com/portal/Portfolio/postgresql\\_ventajas\\_html](http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html).
7. El Lenguaje de Modelo Unificado (UML). *Doc/RS*. [En línea] [Citado el: 20 de 02 de 2010.] <http://www.docirs.cl/uml.htm>.
8. **Fayol, Henry. 2007.** Control - Administración de Empresas y Negocios. *El Prisma.com - Portal para Investigadores y Profesionales*. [En línea] 2007. [Citado el: 20 de 02 de 2010.] [http://www.elprisma.com/apuntes/administracion\\_de\\_empresas/control/](http://www.elprisma.com/apuntes/administracion_de_empresas/control/). 2.
9. **García. 2008.** *Sistema de Información de Gestión de Recursos Humanos*. [En línea] 2008. [Citado el: 20 de 02 de 2010.] <http://190.26.91.148/humanoEL/Ingresar.aspx?Ent=Rionegro>.
10. **González, Elvia. 2009.** Organización de la Información. *Slideshare*. [En línea] 2009. [Citado el: 11 de 02 de 2010.] <http://www.slideshare.net/cuco66/organizacion-de-la-informacion-1144265#>.
11. **Gutián, María Virginia González. 2007.** PROCEDIMIENTO PARA REALIZAR AUDITORIAS DE INFORMACIÓN EN INSTALACIONES HOTELERAS. *eumed.net*. [En línea] 2007. [Citado el: 20 de 02 de 2010.] <http://www.eumed.net/libros/2009d/620/index.htm>. 4.
12. Hospedaje Web Apache. *Cinfonet*. [En línea] [Citado el: 19 de 02 de 2010.] <http://hospedaje.cinfonet.com/?tik=apache>.
13. **Illera, Alvaro Marín. 2009.** MySQL. *e-ghost*. [En línea] 2009. [Citado el: 18 de 02 de 2010.] <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html>.
14. **Leo, Junior. 2009.** Drupal: Novedades, ventajas y requisitos. *Aplicaciones empresariales*. [En línea] 13 de 01 de 2009. [Citado el: 19 de 02 de 2010.] <http://www.aplicacionesempresariales.com/drupal-novedades-ventajas-y-requisitos.html>.
15. **Lopez, Carlos.** ¿Qué es control y que elementos son controlables en la empresa? *GestioPolis.com*. [En línea] [Citado el: 11 de 02 de 2010.] <http://www.gestipolis.com/recursos/experto/catsexp/pagans/ger/48/control.htm>.
16. **Meléndez, R. 2008.** Un nuevo concepto de evaluación. *Universidad Interamericana de Puerto Rico, Recinto Fajardo*. [En línea] 2008. [Citado el: 11 de 02 de 2010.] [http://fajardo.inter.edu/Resiliencia2parte/Documentos/Un\\_Nuevo\\_Concepto\\_de\\_Evaluacion.pdf](http://fajardo.inter.edu/Resiliencia2parte/Documentos/Un_Nuevo_Concepto_de_Evaluacion.pdf).

17. **Mora, Francisco. 2002.** Uml. *Departamento de Ciencia de la Computación e Inteligencia Artificial.* [En línea] 2002. [Citado el: 20 de 02 de 2010.] <http://www.dccia.ua.es/dccia/inf/ asignaturas/GPS/archivos/Uml.PDF>.
18. **Roberto. 2009.** Metodologías de desarrollo del software. *La Tecla de Escape.* [En línea] 25 de 01 de 2009. [Citado el: 21 de 02 de 2010.] <http://latecladeescape.com/w0/ingenieria-del-software/metodologias-de-desarrollo-del-software.html>.
19. Sistemas de información de Recursos Humanos. *Opcion Consultores.* [En línea] [Citado el: 20 de 02 de 2010.] [http://opcion.com.uy/recursos/pdf/publicaciones/opcion\\_sist\\_info\\_rrhh.pdf](http://opcion.com.uy/recursos/pdf/publicaciones/opcion_sist_info_rrhh.pdf). 6.
20. **2007.** Tips para mejorar tu desempeño con Joomla! *maestros del web.* [En línea] 14 de 06 de 2007. [Citado el: 19 de 02 de 2010.] <http://www.maestrosdelweb.com/editorial/tips-para-mejorar-tu-desempeno-con-joomla/>.
21. Ventajas de Joomla. *InFoAL.com.* [En línea] [Citado el: 19 de 02 de 2010.] <http://www.infoal.com/infoal-web-factory/79-servicios-joomla/326-ventajas-de-joomla.html>.
22. Ventajas de utilizar Drupal. *Banana Tools.* [En línea] [Citado el: 19 de 02 de 2010.] <http://www.bananatools.com/drupal/conceptos-basicos/18/caracteristicas-generales-de-drupal/>.
23. **Vidal. 2009.** Embarcadero ER/Studio. *CASE Computación Avanzada y Sistemas Empresariales.* [En línea] 2009. [Citado el: 19 de 02 de 2010.] [http://www.casenet.com.mx/index.php?option=com\\_content&view=article&id=58&Itemid=59](http://www.casenet.com.mx/index.php?option=com_content&view=article&id=58&Itemid=59).

## GLOSARIO

1. **Apache** - Servidor de páginas web de código abierto para diferentes plataformas (UNIX, Windows, y otros).
2. **API (Interfaz de Programación de Aplicaciones)** - Es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.
3. **APPC (Comunicación avanzada entre programas)** - Protocolo SNA Trabajo en Grupo de IBM, mayormente utilizado en equipos AS/400. APPC se define como un protocolo de aplicación porque trabaja en el nivel de presentación del modelo OSI. Sin embargo, también se considera un protocolo de transporte porque APPC utiliza el protocolo LU 6.2 que trabaja en los niveles de transporte y de sesión del modelo OSI.
4. **Blog (Web log)** - Diario en formato web. Puede ser un diario personal o un conjunto de noticias, ordenado por fecha.
5. **Blogger** - es un servicio online de administración de web logs.
6. **CMF (Content Management Framework)** - Entorno a programación de aplicaciones enfocado al desarrollo de CMS.
7. **CMS (Content Management System)** - Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como *Web Content Management (WCM)* sistema de gestión de contenido de webs.
8. **GPL (General Public License)** - Licencia que permite el uso y modificación del código para desarrollar software libre, pero no propietario.

9. **CSS (*Cascading Style Sheets*)** - Las hojas de estilo en cascada contienen un conjunto de etiquetas que definen el formato que se aplicará al contenido de las páginas de una web. Se llaman "en cascada" porque una hoja puede heredar los formatos definidos en otra hoja de forma que no hace falta que vuelva a definirlos. Estas hojas permiten la separación entre el contenido y la presentación en una web.
10. **FTP (*Protocolo de Transferencia de Ficheros*)** - utiliza los protocolos de Internet TCP/IP para permitir la transferencia de datos
11. **HTML (*HyperText Markup Language*)** - Lenguaje basado en marcas que indican las características del texto, utilizado para definir documentos de hipertexto en webs.
12. **HTTP (*HyperText Transfer Protocolo*)** - Protocolo cliente-servidor utilizado para el intercambio de páginas web (HTML).
13. **IIS (*Servicio de Información de Internet*)** - es un conjunto de servicios para servidores usando Microsoft Windows.
14. **LAMP (*Linux, Apache, MySQL y PHP, Perl o Python*)** - Arquitectura formada por el sistema operativo Linux, el servidor web Apache, la base de datos MySQL y uno o más de los lenguajes de programación PHP, Perl o Python.
15. **LAN (*Red de área local*)** - es una red que conecta los ordenadores en un área relativamente pequeña y predeterminada (como una habitación, un edificio, o un conjunto de edificios).
16. **LDAP (*Protocolo Ligero de Acceso a Directorios*)** – es un protocolo de tipo cliente-servidor para acceder a un servicio de directorio.
17. **LCMS (*Learning Content Management System*)** - Software para la gestión automatizada de cursos en línea, que incluye gestión de usuarios, de resultados y de recursos. Es un sistema de gestión de cursos con las capacidades de un CMS y por lo tanto de gestionar también los contenidos de los recursos.
18. **LGPL (*Lesser General Public License*)** - Licencia que permite el uso y modificación de librerías de código para desarrollar software libre o propietario. Antes conocida como *Library GPL*.
19. **LMS (*Learning Management System*)** - se diferencia de los LCMS en que no hay gestión de los contenidos, sino simplemente administración del curso, pero acostumbra a utilizarse como sinónimo. También conocido como *Course Management System (CMS)* o *Virtual Learning Environment (VLE)*
20. **Metadatos** - datos sobre los datos. Información que describe el contenido de los datos. Por ejemplo de un documento serían metadatos, entre de otros, su título, el nombre del autor, la fecha de creación y modificación, y un conjunto de palabras clave que identifiquen su contenido.
21. **MySQL** - base de datos relacional multiplataforma de código abierto, muy popular en aplicaciones web.
22. **NFS (*Sistema de Ficheros en Red*)** - Permite acceder a los ficheros remotos exactamente igual que si fueran locales. El acceso a los ficheros es totalmente transparente al cliente, funcionando con muchas arquitecturas de servidores.
23. **Open source** - Código abierto o código libre. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como *free software*.

24. **OSI (*Interconexión de Sistemas Abiertos*)** - Es un modelo o referente creado por la ISO para la interconexión en un contexto de sistemas abiertos. Se trata de un modelo de comunicaciones estándar entre los diferentes terminales y host. Las comunicaciones siguen unas pautas de siete niveles preestablecidos que son Físico, Enlace, Red, Transporte, Sesión, Presentación y Aplicación.
25. **Perl** - lenguaje de programación de alto nivel que hereda de diversos lenguajes, muy utilizado para el desarrollo de webs dinámicas.
26. **PHP (*PHP Hypertext Preprocessor*)** - Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a la C. Originalmente se conocía como *Personal Hombre Page tools*, herramientas para páginas personales (en Internet).
27. **Python** - lenguaje interpretado de alto nivel orientado a objetos.
28. **TCP/IP (*Protocolo de Control de Transmisión/Protocolo de Internet*)** - Un sistema de protocolos que hacen posibles servicios Telnet, FTP, E-mail, y otros entre ordenadores que no pertenecen a la misma red.
29. **URL (*Localizador de Recurso Uniforme*)** - Dirección de un recurso en la web. Tiene el formato `protocol://maquina.domini:port/ruta/recurs`. Por ejemplo `http://www.uoc.edu/dt/20396/index.html` donde no se indica el puerto porque el protocolo HTTP tiene uno por defecto (80).
30. **WAI (*Iniciativa de Accesibilidad Web*)** - Iniciativa del Consorcio de la *World Wide Web* para asegurar que las webs están diseñadas pensando en el acceso de personas con discapacidades.
31. **WAN (*Red de Área Amplia*)** - una red de ordenadores que abarca un área geográfica relativamente grande. Normalmente, un WAN consiste en dos o más redes de área local (LANs).
32. **Web** - sistema para presentar información en Internet basado en hipertexto. Cuando se utiliza en masculino (*la web, la web*) se refiere a un sitio web entero, en cambio si se utiliza en femenino (*la web, una web*) se refiere a una página web concreta dentro del sitio web.
33. **WebDAV (*Web-based Distributed Authoring and Versioning*)** - es una extensión del protocolo HTTP que permite a los usuarios editar y administrar ficheros de forma colaborativa en servidores web remotos (definición <http://www.webdav.org/de>)
34. **Web log** - Ver 'Blog'.
35. **WYSIWYG (*What You See Is What You Get*)** - Traducido: lo que ves es lo que obtienes, que aplicado a la edición significa trabajar con un documento con el aspecto real que tendrá. Editar una página de HTML en un editor que no sea WYSIWYG, implica trabajar con los códigos que indican el formato que tendrá el texto, sin ver el resultado final.

**AVAL DEL TRABAJO DE DIPLOMA**

El Trabajo de Diploma, titulado "Sistema para el control del funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios de la UCI.", fue realizado en la "Universidad de las Ciencias Informáticas para la dirección de laboratorios". Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un \_\_\_\_ %

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

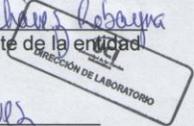
Se creó un sistema de indicadores para el proceso de evaluación de los miembros del METLab; puesto que se detectaron deficiencias en este proceso, así como la ausencia de indicadores para evaluar a los estudiantes del Movimiento.

Se automatizó todo el funcionamiento del Movimiento de Estudiantes Técnicos de Laboratorios de la UCI, creándose una aplicación web con este fin; a la vez que se escogieron las tecnologías propicias teniendo en cuenta las tendencias actuales y escogiendo la más idónea para la solución del problema.

Es posible visualizar en todo momento lo referente a la asistencia, las evaluaciones diarias y mensuales, el cronograma de las inspecciones, así como el resultado de estas, los resultados de los chequeos de emulación; además de servir como enlace para dar a conocer informaciones relacionadas con el movimiento a cada uno de sus miembros así como a la dirección de laboratorios.

Es importante señalar que este sistema facilita y contribuye a mejorar el trabajo del Movimiento de Estudiantes Técnicos de Laboratorios y de la dirección de laboratorios de la UCI.

Y para que así conste, se firma la presente a los 17 días del mes de junio del año 2010.

Ornelas Chávez Robayo  
Representante de la entidad  
  
Colares  
Firma

Especialista Superior  
cargo  
  
\_\_\_\_\_  
cuño