



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

**Título:** Sistema de Gestión de la Información del Trabajo Docente Metodológico en los Departamentos Docentes de la Facultad 6 de la Universidad de las Ciencias Informáticas.

**Autora:** Irasai Sánchez Sánchez

**Tutores:** Ing. Yosvany Marco Pérez  
Ing. Luis Enrique Lorenzo Aguilar

La Habana, Julio, de 2016  
“Año 58 de la Revolución”



*“Inteligencia más carácter,  
el objetivo de una verdadera  
educación”*

---

*Dr. Martin Luther King, Jr.*

---



## *Declaración de Autoría*

---

---

Declaro ser autora del trabajo “**Sistema de Gestión de la Información del Trabajo Docente Metodológico en los Departamentos Docentes de la Facultad 6 de la Universidad de las Ciencias Informáticas**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Irasai Sánchez Sánchez

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

Ing. Luis Enrique Lorenzo Aguilar

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

Ing. Yosvany Marco Pérez

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

**Tutor:** Ing. Yosvany Marco Pérez

**E-mail:** ymarco@uci.cu

- ✓ Ingeniero en Ciencias Informáticas
- ✓ Graduado en el 8va graduación de la UCI en el 2014
- ✓ Profesor del Departamento de Programación y Señales, Digitales Fac 6, UCI

**Tutor:** Ing. Luis Enrique Lorenzo Aguilar

**E-mail:** leaguilar@uci.cu

- ✓ Ingeniero en Ciencias Informáticas
- ✓ Graduado en el 8va graduación de la UCI en el 2014
- ✓ Profesor del Departamento de Programación y Señales, Digitales Fac 6, UCI.
- ✓

*Gracias a todas las personas que de una forma u otra me ayudaron y confiaron en mí, aún en los momentos más difíciles.*

*Gracias a mis tutores por apoyarme, Yosvany y Lorenzo, gracias de todo corazón...*

*A mi mamá por apoyarme siempre, aun cuando las decisiones tomadas no fueran las más sabias, por ser la mejor madre del mundo.*

*A quien me ha hecho feliz durante los últimos 4 años de mi vida, a quien me ha apoyado durante toda mi carrera y amo con todo mi corazón. Mi esposo.*

*A mi hermano que, a pesar de estar lejos, me ha apoyado, ayudado, confiado y sobre todo siempre ha estado en mi mente.*

*Gracias a todas las personas que me han apoyado, ayudado, aconsejado, animado durante toda mi carrera.*

## Índice

Resumen.....	IX
Abstract .....	X
Introducción .....	1
Capítulo 1 Fundamentos teóricos sobre la gestión de la información .....	5
1.1 Elementos asociados al dominio del problema .....	5
1.1.1 <i>Departamentos Docentes</i> .....	5
1.1.2 <i>Trabajo Docente Metodológico</i> .....	5
1.1.3 <i>Actividades Docentes Metodológicas</i> .....	6
1.1.4 <i>Información</i> .....	7
1.1.5 <i>Sistemas de gestión</i> .....	8
1.2 Análisis de soluciones similares.....	8
1.2.1 <i>Conclusiones del estudio realizado</i> .....	10
1.3 Metodología de desarrollo.....	10
1.3.1 <i>Metodologías robustas o pesadas</i> .....	10
1.3.2 <i>Metodologías ligeras o ágiles</i> .....	11
1.3.3 <i>Resultado de análisis de las metodologías</i> .....	12
1.4 Herramientas y Tecnologías. ....	12
1.4.1 <i>Marco de trabajo Symfony 2.3.7</i> .....	12
1.4.2 <i>Lenguajes de programación</i> .....	14
1.4.3 <i>Entorno Integrado de Desarrollo NetBeans 7.3</i> .....	15
1.4.5 <i>Lenguaje de Modelado Unificado (UML)</i> .....	15
1.4.6 <i>Herramienta CASE Visual Paradigm para UML 8.0</i> .....	16
1.4.7 <i>Sistema Gestor de Bases de Datos PostgreSQL 9.3</i> .....	17
1.4.8 <i>PGAdmin III 1.22.1 para el manejo de PostgreSQL</i> .....	17
1.4.9 <i>Apache 2.2.22 como servidor web a utilizar</i> .....	18
1.5 Conclusiones Parciales .....	18
Capítulo 2: Análisis y diseño del Sistema de Gestión de la Información del Trabajo Docente Metodológico .....	19
2.1 Modelo del negocio .....	19
2.1.2 <i>Actores y trabajadores del negocio</i> .....	20
2.1.3 <i>Diagrama de Casos de Uso del Negocio</i> .....	21
2.1.4 <i>Descripción textual de los Casos de Uso del Negocio</i> .....	22
2.1.5 <i>Diagrama de actividades del Caso de Uso del Negocio: Elaborar plan de trabajo</i>	

---

<i>docente metodológico</i> .....	23
2.2 Requisitos de Software .....	23
2.2.1 <i>Requisitos Funcionales</i> .....	23
2.2.2 <i>Requisitos No Funcionales</i> .....	24
2.3 Modelo del Sistema .....	27
2.3.1 <i>Actores del Sistema</i> .....	27
2.3.2 <i>Diagrama de Casos de Uso del Sistema</i> .....	28
2.3.3 <i>Patrones de Casos de Uso utilizados</i> .....	32
2.4 Elementos fundamentales de la arquitectura .....	33
2.4.1 <i>Patrones arquitectónicos y de diseño</i> .....	33
2.5 Modelo de Diseño .....	37
2.5.1 <i>Diagrama de clases del diseño</i> .....	38
2.6 Diagrama Entidad-Relación .....	38
2.7 Conclusiones parciales .....	39
Capítulo 3: Implementación y Pruebas del Sistema de gestión de la información generada por el trabajo docente metodológico .....	40
3.1 Modelo de Implementación .....	40
3.1.1 <i>Representación de dependencias entre componentes de software</i> .....	40
3.2 Modelo de Despliegue .....	42
3.3 Estándares y estilos de codificación .....	43
3.4 El proceso de pruebas .....	45
3.4.1 <i>Tipos de pruebas</i> .....	45
3.4.2 <i>Diseño de casos de prueba</i> .....	47
3.5 Ejecución de las pruebas .....	53
3.6 Aportes a considerar .....	55
3.7 Conclusiones parciales .....	56
Conclusiones generales .....	57
Referencias Bibliográficas .....	59

## Índice de Figuras

<i>Fig. 1 Diagrama de Casos de Uso del Negocio. Sistema de Gestión de la Información del Trabajo Docente Metodológico .....</i>	<i>21</i>
<i>Fig. 2 Diagrama de actividades del Caso de Uso del Negocio: Elaborar PTDM .....</i>	<i>23</i>
<i>Fig. 3 Diagrama de Casos de Uso del Sistema. Sistema de Gestión de la Información del Trabajo Docente Metodológico .....</i>	<i>28</i>
<i>Fig. 4 Diagrama de paquetes de diseño del Caso de Uso: Gestionar PTDM .....</i>	<i>35</i>
<i>Fig. 5 Diagrama de clases del diseño. Caso de Uso: Gestionar PTDM.....</i>	<i>38</i>
<i>Fig. 6 Diagrama Entidad-Relación del Sistema de Gestión de la Información del Trabajo Docente Metodológico .....</i>	<i>39</i>
<i>Fig. 7 Diagrama de paquetes. Caso de Uso: Gestionar PTDM .....</i>	<i>41</i>
<i>Fig. 8 Diagrama de componentes. Caso de Uso: Gestionar PTDM.....</i>	<i>41</i>
<i>Fig. 9 Diagrama de despliegue del Sistema para la Gestión de la Información del Trabajo Docente Metodológico .....</i>	<i>42</i>
<i>Fig. 10 Fragmento de código de la clase PtdmController .....</i>	<i>44</i>
<i>Fig. 11 Ejecución de las pruebas de Caja Negra. Resultados de las iteraciones realizadas.....</i>	<i>53</i>

## Índice de Tablas

<i>Tabla. 1 Actores del Negocio.....</i>	<i>20</i>
<i>Tabla. 2 Trabajadores del Negocio.....</i>	<i>21</i>
<i>Tabla. 3 Descripción textual del Caso de Uso del Negocio: Elaborar PTDM .....</i>	<i>22</i>
<i>Tabla. 4 Interfaces de Hardware. Características .....</i>	<i>26</i>
<i>Tabla. 5 Actores del Sistema .....</i>	<i>27</i>
<i>Tabla. 6 Descripción textual del Caso de Uso del sistema: Gestionar PTDM .....</i>	<i>29</i>
<i>Tabla. 7 Secciones a probar en el Caso de Uso del Sistema: Gestionar PTDM .....</i>	<i>47</i>
<i>Tabla. 8 Descripción de las Variables.....</i>	<i>48</i>
<i>Tabla. 9 Caso de Uso: Crear plan de Trabajo Docente Metodológico.....</i>	<i>49</i>
<i>Tabla. 10 Caso de Uso: Modificar plan de Trabajo Docente Metodológico .....</i>	<i>51</i>
<i>Tabla. 11Caso de uso: Eliminar plan de Trabajo Docente Metodológico .....</i>	<i>52</i>
<i>Tabla. 12 Caso de Uso: Listar plan de Trabajo Docente Metodológico.....</i>	<i>53</i>
<i>Tabla. 13 Resultados de las pruebas de carga sobre la propuesta de solución.....</i>	<i>54</i>
<i>Tabla. 14 Resultados de las pruebas de tensión sobre la propuesta de solución .....</i>	<i>54</i>

### Resumen

Para toda empresa, centro laboral o institución es de vital importancia la gestión de los documentos que esta genera. Si este proceso es asistido por un sistema informático, teniendo en cuenta los nuevos avances de las Tecnologías de la Información y las Comunicaciones (TICs) se garantiza un impacto positivo provocando que la información generada esté más organizada y disponible. La presente investigación presenta el desarrollo del Sistema de Gestión de la Información del Trabajo Docente Metodológico en los Departamentos Docentes de la Facultad 6 de la Universidad de las Ciencias Informáticas, el cual permite agilizar e informatizar dicho proceso de gestión que hasta el momento de la investigación se realiza de forma manual. Para guiar el desarrollo del sistema se utilizó como metodología OpenUp; PHP 5.3.3 y Java Script como lenguajes de programación, Symfony 2.3.7 como marco de trabajo y PostgreSQL 9.3 como gestor de base de datos, entre otras herramientas y tecnologías de código abierto acorde a la política de soberanía tecnológica que defiende el país. El sistema obtenido permite la gestión de la información generada por el trabajo docente-metodológico, dándole a este proceso una mayor rapidez, organización y control, garantizando además una mayor disponibilidad de la información, lo que contribuye a la toma de decisiones por parte de los directivos de la institución.

**Palabras claves:** sistema de gestión, información, documentos, trabajo docente-metodológico, actividades metodológicas

## **Abstract**

For any enterprise, work place or institution it is of vital importance the management of the documents any of the aforementioned generates. If this management process is assisted by a management system, taking into consideration new advancements in Information and Communication Technologies, a greater chance of success in the management of this process are assured and the information that is produced as such will be better organized and available. This investigation has as an objective the development of the Methodological Teaching Work Information Management System in which the teaching departments of faculty 6 of the University of Information Sciences, has permitted to speed up and inform the aforementioned management process which up until this point was done by hand. To guide the development process of the system the OpenUp methodology was used, PHP 5.3.3 and Java Script were used as programming languages, Symfony 2.3.7 as a development framework, NetBeans 7.3 as the integrated development environment (IDE) and PostgreSQL 9.3 as a data base manager. The development of this system will allow the management of information generated, giving this process a greater advancement in speed, organization and control. This process will hence forth guarantee a greater availability in managed information contributory to administrative decision making.

**Keywords:** management system, information, documents, Methodological Teaching Work, Methodological activities.

## Introducción

La industria del software ha alcanzado una posición relevante, teniendo un papel importante en la traducción de problemas de la vida real, en soluciones objetivas que enriquezcan el razonamiento lógico y abstracto de todos los implicados. Estas soluciones surgen del análisis crítico y profundo de diversas problemáticas que tienen como trasfondo alguna insuficiencia en la ejecución de un proceso o en la gestión realizada por el hombre. Dichas premisas, son sustentadas con el hecho de que algunos procesos son realmente complejos cuando involucran grandes cantidades de información estadística, actualización de datos sensibles para facilitar la toma de decisiones y una amplia retroalimentación entre diferentes áreas; donde cualquier retraso o imprecisión puede afectar seriamente a la institución o negocio en cuestión.

En toda área de trabajo es de vital importancia la gestión de la información y los documentos que se generan con el objetivo de lograr una buena organización, planificación y control. También para garantizar el estado de conservación de dichos documentos y al mismo tiempo facilitar el trabajo de las personas que necesitan acceder y consultar la información archivada. Con la aparición de los sistemas para gestionar la información se ha contribuido a que se gestionen mejor los procesos y que la información que se genere esté más organizada y disponible.

Las universidades generalmente no realizan los mismos procesos que una empresa, pero en ellas se trabaja con un gran número de información generada por cientos de procesos académicos o no, que en ellas se realizan. Se hace necesaria la informatización de muchos de estos procesos que se desarrollan para así poder mantener un mayor control sobre los documentos y poder tomar ciertas decisiones en el menor tiempo posible.

Cuba no está ajena a este fenómeno tan necesario de informatizar la sociedad, por esa razón se crea la Universidad de las Ciencias Informáticas (UCI), surgida en el año 2002 bajo la idea del Comandante en Jefe Fidel Castro Ruz. Dentro del proceso de enseñanza- aprendizaje que se desarrolla en cualquier centro educacional del país, es de vital importancia informatizar la gestión de la información generada por las actividades docentes metodológicas. Estas actividades no son más que las formas de organizar el trabajo docente-metodológico y la función de las mismas es orientar a los docentes acerca de los métodos, procedimientos y medios de enseñanza que se deben utilizar en el proceso de enseñanza-aprendizaje (Torres, y otros, 2010). Entre las actividades docentes metodológicas más utilizadas se encuentran las reuniones metodológicas y los controles a clase. Cada actividad de estas genera un documento como evidencia de que la misma fue desarrollada.

En la UCI existen diversos sistemas de gestión que involucran el proceso docente educativo, un ejemplo de esto es el Sistema de Gestión Universitaria. Este sistema se encarga de gestionar todo lo referente al control del desarrollo del proceso docente, notas, asistencia a clase, entre otros aspectos, manteniendo a estudiantes y profesores al tanto del desempeño académico y profesional. En la actualidad se proyecta implementar un módulo que gestione los documentos generados por las actividades docentes metodológicas, pero hasta el momento del desarrollo de esta investigación solo se ha puesto a prueba la gestión de los controles a clase.

En la Universidad desde sus inicios, se mantiene el control de la documentación generada por las actividades docentes metodológicas mediante el llenado de planillas manuscritas o digitalmente para luego ser impresas y archivadas; como consecuencia en ocasiones la información no se entrega en tiempo y con la calidad que requiere este tipo de proceso. Debido a que la información es almacenada en formato duro por los jefes de departamentos o los responsables de las áreas docentes, se hace engorroso emitir reportes estadísticos y agilizar la toma de decisiones por parte de los directivos con respecto al cumplimiento de los planes docentes metodológicos y demás actividades. Esta pérdida de tiempo para conocer los datos almacenados sobre las actividades docentes metodológicas trae consigo una gran insatisfacción en todos los involucrados, provocando molestias innecesarias y sobrecarga de trabajo en aquellos que tienen que contabilizar toda la información para emitir un reporte determinado. Por otra parte, se dificulta la evaluación del cumplimiento de los planes individuales de cada docente respecto a las actividades previstas.

Por lo anteriormente planteado surge como **problema de la investigación**: ¿Cómo contribuir a la gestión de la información que se genera en el trabajo docente metodológico en los departamentos docentes de la Universidad de las Ciencias Informáticas?

A partir del problema planteado se define como **objeto de estudio**: El proceso de gestión de la información. Enmarcado en el **campo de acción**: El proceso de gestión de la información que genera el trabajo docente metodológico en los departamentos docentes de la Universidad de las Ciencias Informáticas.

Para dar solución al problema planteado se define como **objetivo general**: Desarrollar una aplicación que contribuya a la gestión de la información que genera el trabajo docente metodológico en los departamentos docentes de la Facultad 6 de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general se definen las siguientes **tareas de la investigación**:

1. Análisis de los conceptos fundamentales implicados en el proceso de gestión de documentos y de la información relacionada con el trabajo docente metodológico.
2. Selección de la metodología y herramientas a utilizar como guía del proceso de construcción de la solución propuesta.
3. Identificación de las necesidades funcionales y tecnológicas de la propuesta de solución.
4. Identificación de los principios arquitectónicos del sistema.
5. Identificación de los principios de diseño y funcionamiento del sistema.
6. Implementación de los principios de diseño y funcionamiento con el objetivo de satisfacer las necesidades funcionales y tecnológicas del sistema.
7. Realización de las pruebas de software para comprobar el correcto funcionamiento del sistema propuesto.

Para guiar la investigación se definen las siguientes **preguntas científicas**:

- ¿Cuáles son los referentes teóricos de la gestión de la información que genera el trabajo docente metodológico en los departamentos docentes de la Facultad 6 de la Universidad de las Ciencias Informáticas?
- ¿Cuáles son las características que deben cumplir los procesos que se desarrollan en el trabajo docente metodológico para su futura informatización?
- ¿Cómo diseñar una aplicación web que gestione la información que generan los procesos que se desarrollan en el trabajo docente metodológico?
- ¿Cómo implementar una aplicación web que gestione de la información que generan los procesos que se desarrollan en el trabajo docente metodológico?
- ¿La aplicación desarrollada contribuye en la gestión de la información que generan los procesos del trabajo docente metodológico?

Para la investigación se han utilizado diferentes tipos de **métodos de investigación**, tanto teóricos como empíricos, los cuales se describen a continuación:

### **Métodos Teóricos**

- **Analítico-Sintético:** Utilizado al estudiar la documentación especializada que permitió la extracción de los elementos más importantes relacionados con el objeto de estudio permitiendo

arribar a las conclusiones de la investigación, así como determinar las características de los requisitos a implementar.

## Métodos Empíricos

- **Monitoreo de proyectos:** Este método es utilizado para analizar las aplicaciones existentes relacionadas con la gestión del proceso docente-educativo para utilizarlas como referencia a la hora de realizar la solución propuesta.
- **Tormenta de ideas:** Utilizado para obtener las necesidades del cliente y realizar el levantamiento de los requisitos con los que debe cumplir la aplicación.

## Estructura del documento

El Trabajo de Diploma está estructurado de la siguiente manera: introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas.

### Capítulo 1: “Fundamentos teóricos sobre la gestión de la información”

En este capítulo se realiza la elaboración del marco teórico donde se exponen los conceptos asociados a la solución de la problemática y las diferentes soluciones existentes a nivel mundial. De igual manera se describen y caracterizan las herramientas y el lenguaje de programación para el diseño e implementación de la solución, así como la metodología de desarrollo más recomendable.

### Capítulo 2: “Análisis y diseño del Sistema de Gestión de la Información del Trabajo Docente Metodológico”

Se exponen las principales características y cualidades de la solución a implementar, además se identifican los requisitos, se escoge la arquitectura y los patrones de diseño, se diseñan las clases y se detallan los pasos de la metodología propuesta, incluyendo varios de sus artefactos y diagramas.

### Capítulo 3: “Implementación y pruebas del Sistema de Gestión de la Información del Trabajo Docente Metodológico”

Se muestra la situación física de los distintos componentes lógicos desarrollados a través del modelo de despliegue y la organización del sistema mediante el modelo de componentes. Y por último se analiza el funcionamiento del sistema desarrollado aplicándole las pruebas necesarias para demostrar que la solución es correcta.

## Capítulo 1 Fundamentos teóricos sobre la gestión de la información

Este capítulo contiene todos los aspectos teóricos sobre la gestión de la información y el trabajo metodológico, así como de la integración entre ellos para dar cumplimiento a los objetivos planteados para solucionar la problemática existente. También se abordan aspectos importantes de soluciones similares tanto a nivel nacional como internacional que se pudieran tener en cuenta para un mejor desarrollo de la solución. Por último se describen y caracterizan la metodología, las herramientas y los lenguajes utilizados en la investigación.

### 1.1 Elementos asociados al dominio del problema

A continuación se relacionan los principales conceptos o temáticas que están asociados al desarrollo de la investigación.

#### 1.1.1 Departamentos Docentes

Un Departamento docente es una estructura académica básica en la educación superior que está constituida por un colectivo de profesores y trabajadores, cuyo alcance se identifica en que tienen la responsabilidad de dirigir de manera integrada y bajo criterios de calidad, pertinencia, eficiencia y eficacia las funciones de docencia, investigación, posgrado y extensión propias del área de conocimiento en cuestión, en correspondencia con las necesidades, demandas y expectativas de la universidad. Esta estructura de dirección se subordina a una facultad o directamente al jefe máximo de la institución puede ser el rector o director (León, 2015).

#### 1.1.2 Trabajo Docente Metodológico

El reglamento del Ministerio de Educación Superior define como Trabajo Docente Metodológico a la actividad que se realiza con el fin de mejorar de forma continua el proceso educativo; basándose fundamentalmente en la preparación didáctica que poseen los educadores, cuadros y funcionarios en el dominio de los objetivos del año de vida, grado y nivel, del contenido de los programas, de los métodos y medios con que cuenta, así como el análisis crítico y la experiencia acumulada (Valdés, 2007).

El Trabajo Docente-Metodológico contribuye a lograr la eficaz preparación del profesor, como control sistemático de su auto preparación y, en consecuencia, la elevación de la calidad de la actividad docente, la adquisición de un sistema de trabajo que propicie que desde el aula aprenda a resolver, de manera científica, los problemas que surgen. Por ello la preparación metodológica en los profesores universitarios

propicia la integración de los aspectos políticos, ideológicos, metodológicos, pedagógicos y científicos, mediante los tipos fundamentales de actividades docente-metodológicas más empleadas en la Educación Superior.

### **1.1.3 Actividades Docentes Metodológicas**

Las Actividades Docentes Metodológicas (en lo adelante ADM) son las formas de organizar el Trabajo Docente-Metodológico. Su función es orientar a los docentes acerca de los métodos, procedimientos y medios de enseñanza que se deben utilizar en el proceso de enseñanza-aprendizaje, así como la estructura metodológica y las formas de control del aprendizaje por parte de los estudiantes, en el desarrollo de las principales formas de organización de la enseñanza, de los temas y de las asignaturas (Torres, y otros, 2010).

Las Actividades Docentes Metodológicas en los diferentes niveles organizativos de la educación en nuestro país se estructuran de la siguiente manera:

- Reunión metodológica
- Clase metodológica demostrativa
- Clase metodológica instructiva
- Clase abierta
- Taller metodológico
- Seminario científico metodológico

Todas estas actividades cumplen la función de orientar a los docentes mediante la demostración, la organización y el análisis dependiendo de la actividad que sea desarrollada, con el objetivo elevar la maestría pedagógica de los docentes para lograr así una mayor eficiencia del proceso de Enseñanza - Aprendizaje y una calidad superior en la enseñanza del egresado (Valdés, 2007).

### **Control a clase**

El control a clase según (Valdés, 2007), se refiere al control del proceso docente educativo como la parte del contenido del Trabajo Docente Metodológico y el medio fundamental para conocer la calidad de dicho proceso, evaluar sus resultados y dirigirlo hacia el cumplimiento de sus objetivos. Por otra (Castillo, y otros, 2010) establece además que el control a clases del proceso docente educativo comprende el control a la planificación y organización, el control a la calidad de la ejecución, el control a la actividad docente, el cumplimiento de los programas de estudio, el aseguramiento material y humano de las

actividades docentes y el control a la preparación adquirida por los docentes. Para ello, los controladores deben tener una categoría docente igual o mayor que la del controlado y la cantidad de controles dependerá de la maestría pedagógica y del dominio del contenido que tenga el docente de la asignatura que imparte.

## **Documentos que generan las actividades metodológicas**

En un departamento docente se genera una gran cantidad de documentación, debido a que cada actividad que es desarrollada en el mismo deja una evidencia o acta de que dicha actividad fue efectuada; como ejemplo de esta documentación podemos citar al plan de trabajo docente metodológico, acta de control a clase, acta de reunión metodológica, entre otras. Cada uno de estos documentos plasma los resultados de los controles realizados a las diferentes actividades docentes. Estas actas también reflejan logros, deficiencias, recomendaciones, evaluaciones alcanzadas, también quedan evidenciados los participantes y responsables, entre otros aspectos.

El plan de Trabajo Docente Metodológico es el documento que será confeccionado a partir de los problemas detectados durante el proceso docente, los cuales serán el punto de partida para constituir los objetivos a alcanzar. Luego, dichos objetivos se concretan en acciones específicas a desarrollar para cada curso académico, utilizando las formas y tipos que se establecen en el Reglamento Docente y Metodológico del Ministerio de Educación Superior de Cuba. Dicho plan deberá ser confeccionado por cada uno de los colectivos y niveles de dirección al inicio de cada curso y podrán adecuarse para cada período escolar. A nivel de facultad todos los documentos deberán ser conservados en el departamento docente para avalar el Trabajo Docente Metodológico después de ser aprobado por el decano, a su vez se conservará en las facultades el Plan de Trabajo Docente Metodológico de la Facultad aprobado por la dirección del centro de educación superior (Valdés, 2007).

### **1.1.4 Información**

La información no es más que un recurso que se puede comparar con un conjunto de bienes que desempeñan un papel creciente e importante en la vida económica, social, cultural y política de toda sociedad. Existen escenarios complejos en los que la información se utiliza como un recurso económico y se aprecia un mayor uso de ella por el público. Su contribución se enmarca en la toma de decisiones; así como en su papel de mejorar la gestión del conocimiento (Pérez, y otros, 2009).

## **1.1.5 Sistemas de gestión**

Gestión es la acción de gestionar y administrar una actividad profesional destinada a establecer los objetivos y medios para su realización, a precisar la organización con el fin de elaborar la estrategia del desarrollo y a ejecutar la gestión del personal. Asimismo en la gestión es muy importante la acción, porque es la expresión de interés capaz de influir en una situación dada (Ruiz, 2013)

Un Sistema de Gestión es aquel que sirve de ayuda para lograr las metas y objetivos de una organización, a través de una serie de estrategias, entre las cuales se encuentra la optimización de los procesos y el enfoque basado en la gestión y la disciplina. Es decir un Sistema de Gestión, involucra a un conjunto de etapas integradas en un proceso continuo, que funcionan hasta lograr su mejora definitiva (SIG, 2015)

## **1.1.6 Gestión de la información**

La gestión de la información se puede definir como el conjunto de actividades realizadas con el fin de controlar, almacenar y posteriormente, recuperar adecuadamente la información producida, recibida o retenida por cualquier organización en el desarrollo de sus actividades. En el centro de la gestión de la información se encuentra la gestión de la documentación (la información que queda plasmada en documentos) (Ruesta, 2001).

## **1.1.7 Sistema de Gestión de la Información**

Un Sistema de Gestión de Información es el encargado de seleccionar, procesar y distribuir la información procedente de los ámbitos interno, externo y corporativo. La información interna no es más que la producida en la actividad cotidiana de la institución, la información externa es la adquirida por la institución para disponer de información sobre los temas de su interés y la información corporativa o pública es la que la institución emite al exterior (Arévalo, 2007).

## **1.2 Análisis de soluciones similares**

En el mundo actual existe un gran número de sistemas que gestionan información, a continuación se hace una caracterización de diferentes sistemas nacionales e internacionales que incluyen algún tipo de gestión relacionada con el trabajo docente metodológico.

**Sistema de la Facultad de Ingeniería de la Universidad Javeriana de Bogotá:** Brinda algunas funcionalidades docentes como son: realizar evaluaciones, lleva el control de asistencia de los alumnos, además del registro de entradas/salidas de documentos del centro. Permite exportar varios reportes a diversos formatos (xls, pdf, html, entre otros). Brinda información pertinente de las estructuras docentes de la institución. Se caracteriza por ser un sistema soportado por la web, con una arquitectura de despliegue

cliente-servidor, fue desarrollado con HTML, CSS y JavaScript, como lenguajes de programación del lado del cliente. Los procesos anteriormente mencionados, no se asemejan a los que se necesitan para gestionar los documentos generados en el trabajo docente metodológico en los departamentos docentes de la Facultad 6, siendo estos documentos diferentes a los que se manipulan por el sistema estudiado; el cual por las razones mencionadas no es viable para dar solución a la problemática planteada en esta investigación (Pontificia universidad javeriana de bogotá, 2016).

**Sistema de la Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la provincia de Buenos Aires:** El sistema se centra en las caracterizaciones de las investigaciones de las ciencias exactas, donde se incluyen las diferentes informaciones referidas a los estudiantes y graduados. Está desarrollado con la combinación de los lenguajes del lado del cliente: HTML, JavaScript y CSS, aunque algunas de sus páginas están hechas en XML. Presenta una funcionalidad referida a la gestión de los planes de trabajo. En la misma se muestra información sobre las acciones y actividades que se realizan en la universidad durante un período determinado, pero a dichas acciones no se le puede dar seguimiento, ni se pueden generar reportes de las mismas para ser almacenados en formato duro (UNICEN , 2015).

**Sistema de la Universidad de Ciencias Médicas de La Habana:** El sistema presenta en su portada una buena fundamentación de las tareas, informaciones y convocatorias de actividades que realiza la universidad. En su diseño gráfico, utiliza colores agradables a la vista del usuario. Trabaja de manera centralizada las informaciones de sus centros académicos y de sus eventos universitarios. El mismo se relaciona con diferentes sitios del mundo que trabajan la misma rama de las ciencias médicas. Proporciona información de los cursos de posgrado, de las especialidades, de la síntesis de los planes de estudios por carreras, entre otros. Está desarrollado con HTML, JavaScript y CSS, como lenguajes del lado del cliente. El sistema en estudio no gestiona información relacionada con la documentación generada por las actividades docentes-metodológicas, difiriendo estas últimas de las definidas por el MES, por ende no se puede obtener estadísticas ni monitorear el cumplimiento de las mismas resultando una solución que no resulta viable para dar cumplimiento al objetivo de esta investigación (Zelada, 1999).

**Sistema de Gestión Académica de la UCI:** Este sistema utilizado en la UCI, tiene numerosas funcionalidades que permiten la gestión de la información académica, almacena los datos de los resultados obtenidos en las pruebas realizadas, la matrícula con que cuenta de la universidad, asistencia a clases y causas que describen la situación del que se ausente, resultados de los cursos optativos, entre otras. En la actualidad el sistema no cuenta con un módulo que gestione la información generada por las actividades docentes metodológicas, hasta el momento solo se ha puesto a prueba la gestión de los controles a clase (Maso, y otros, 2008).

## **1.2.1 Conclusiones del estudio realizado**

Con el estudio de estas aplicaciones se concluye que todas ellas de una forma u otra trabajan con información característica de la entidad que representan y acorde al sistema educacional por el que se rigen, el cual en ninguno de los casos resulta ajustable a las especificidades de la problemática en cuestión. Las soluciones estudiadas no gestionan directamente la información que generan las actividades docentes metodológicas ni sus evidencias, por lo que no resultan factibles para dar solución a la problemática de esta investigación. Es necesario implementar un nuevo sistema informático capaz de gestionar la información generada por el Trabajo Docente Metodológico de los departamentos docentes de la facultad 6 de la UCI.

## **1.3 Metodología de desarrollo**

Las metodologías de desarrollo de software son específicamente quienes permiten a los desarrolladores, a partir de un conjunto de técnicas, procedimientos y herramientas, elaborar el nuevo producto; detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla, o sea, describen paso a paso el proceso de conformación del producto informático deseado (Romero, 2009).

### **1.3.1 Metodologías robustas o pesadas**

Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más fuertes y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida.

**Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés):** (Sommerville, 2005) define a RUP como un modelo de proceso de software genérico, que presenta los desarrollos de software como una actividad iterativa de cuatro fases: creación, elaboración, construcción y transición. Creación establece los casos de uso del negocio para el sistema, elaboración define la arquitectura, la construcción implementa el sistema y la transición despliega el sistema en el entorno del cliente.

RUP provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. Se basa en componentes. Usa el Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) en la preparación de todos los planos del sistema. Los aspectos distintivos de RUP están capturados en tres

conceptos clave: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura (Sulca, 2012).

**Microsoft Solutions Framework (MSF):** Es una metodología desarrollada por Microsoft consulting Services que define un marco de trabajo de referencia para construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft para cualquier plataforma (Linux, Citrix, Microsoft, Unix). Provee un conjunto de principios, modelos, disciplinas, conceptos y lineamientos para la entrega de tecnología de la información utilizando soluciones Microsoft. No se limita sólo al desarrollo de aplicaciones, también es aplicable a otros proyectos tales como proyectos de implementación de redes o infraestructura. El modelo de proceso de MSF combina el concepto de la administración de proyectos tradicional (cascada), con los modelos en espiral (mejora continua) para capitalizar en las fortalezas de cada uno de estos enfoques. MSF combina los beneficios de la planeación en cascada basado en el alcance de hitos, con los entregables iterativos e incrementales del modelo en espiral (Ruiz, y otros, 2012).

### **1.3.2 Metodologías ligeras o ágiles**

Las metodologías ágiles han despertado gran interés en el desarrollo de software ya que proponen simplicidad y velocidad para crear sistemas.

**XP (eXtreme Programming):** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre (Letelier, y otros, 2006).

**SCRUM:** Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprints es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (Letelier, y otros, 2006).

**Proceso Unificado Abierto (OpenUP):** Es un proceso modelo y extensible, dirigido a gestión y desarrollo de proyectos de software basados en desarrollo iterativo, ágil e incremental. Es apropiado para proyectos pequeños y de bajos recursos; y es aplicable a un conjunto amplio de plataformas y aplicaciones de desarrollo. Está basado en RUP (Rational Unified Process, en español Proceso Unificado Racional), fue desarrollado por IBM (International Business Machines) y es reconocido mundialmente como uno de los procesos de desarrollo de software de mayor calidad. Se basa en los principios de adaptación, importancia a los involucrados e interesados en los resultados del proyecto; colaboración, valor a la iteración; y calidad continua (Balduino, 2007).

### **1.3.3 Resultado de análisis de las metodologías**

En función del estudio realizado y las características del equipo de desarrollo y de la solución propuesta en cuestión se determina el empleo de una metodología ágil, ya que el tiempo de desarrollo es limitado, el proyecto que se pretende es relativamente pequeño, así como el equipo responsable del desarrollo del mismo. Se necesita solo la documentación imprescindible para dar soporte al sistema y su proceso de construcción.

Dentro de las metodologías ágiles se optó por la selección de OpenUp basado en sus características distintivas: apropiada para proyectos pequeños y de bajos recursos, basada en casos de uso, centrada en la arquitectura, además permite disminuir las probabilidades de fracaso e incrementar las probabilidades de éxito. Detecta errores tempranos a través de un ciclo iterativo y evita la elaboración de documentación, diagramas e iteraciones innecesarias requeridas por otras metodologías. Posee un enfoque centrado al cliente y con iteraciones cortas ajustándose plenamente a las necesidades para el desarrollo de la propuesta de solución.

## **1.4 Herramientas y Tecnologías.**

Seguidamente se realiza una descripción de las herramientas y tecnologías utilizadas para el desarrollo del sistema.

### **1.4.1 Marco de trabajo Symfony 2.3.7**

Un marco de trabajo es un software que se puede personalizar e intercambiar para favorecer en tiempo el desarrollo de una aplicación. Se puede considerar como otra aplicación incompleta y configurable a la que se le añaden las últimas piezas para construir una aplicación final. Ayudan a desarrollar aplicaciones con mayor rapidez, pues poseen una estructura definida y una organización para el desarrollo y mantenimiento del software desarrollado.

Symfony 2.3.7 es un marco de trabajo PHP caracterizado por su madurez, estabilidad y por el grado de

documentación con la que cuenta. Explota al máximo todas las nuevas características de PHP 5.3 y por eso es uno de los marcos de trabajo PHP con mejor rendimiento. Su arquitectura interna está completamente desacoplada, lo que permite reemplazar o eliminar fácilmente aquellas partes que no encajen en los proyectos (Pérez, 2006).

Entre las características de Symfony2 que se pueden destacar están las siguientes (Zaninneto, 2008):

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente del Sistema Gestor de Bases de Datos. Su capa de abstracción y el uso de Doctrine2, permiten cambiar con facilidad de Sistema Gestor de Bases de Datos en cualquier fase del proyecto.
- Utiliza programación orientada a objetos.
- Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Posee una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

## **JQuery 1.9 para JavaScript**

JQuery es una biblioteca JavaScript que ofrece una sólida infraestructura que brinda un grupo importante de ventajas y facilidades para la creación de aplicaciones de mediana y alta complejidad del lado del cliente. Está orientada a facilitar la labor en cuanto a la creación de interfaces de usuario, efectos dinámicos y otros aspectos. Es un producto estable, bien documentado y con un gran equipo de desarrolladores a cargo de su mejora y actualización. Posee una dilatada comunidad de creadores de “plugins” o componentes, lo que hace fácil encontrar soluciones ya creadas en JQuery para implementar asuntos como interfaces de usuario, galerías, votaciones, efectos diversos, y otros (Foundation, 2010b).

## 1.4.2 Lenguajes de programación.

Un lenguaje de programación consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos. Es diseñado para describir el grupo de acciones consecutivas que un equipo debe ejecutar, lo que permite crear programas informáticos posibilitando al desarrollador comunicarse con los dispositivos de hardware y software existentes (Kioskea, 2011).

### PHP 5.3.3 como lenguaje de programación del lado del servidor

PHP, acrónimo de (Hypertext Pre Processor) constituye un lenguaje de programación interpretado, de propósito general y ampliamente difundido en el mundo. Diseñado originalmente para la creación de páginas web dinámicas, principalmente en interpretación del lado del servidor, aunque puede ser incrustado dentro de código HTML.

Está especialmente diseñado para incrementar el dinamismo de las páginas web, aprovechando los recursos de las redes informáticas y los escasos requisitos de hardware que solicita el lenguaje para que sus aplicaciones funcionen correctamente. Además, permite aplicar técnicas de programación orientada a objetos. Es un lenguaje completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos. El código fuente escrito en este lenguaje es invisible al navegador ya que es el servidor el encargado de ejecutar el código y enviar su resultado HTML al cliente, logrando una programación segura y confiable (Bakken, y otros, 2005).

- Dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas (Team, 2011):
- Soporte para una gran cantidad de Bases de Datos: MySQL, PostgreSQL, Oracle, MS SQL Server, SybasemSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en pdf (documentos de Acrobat Reader) e incluso analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas de la Web de fácil programación.

## **JavaScript 1.9 como lenguaje de programación del lado del cliente**

JavaScript es uno de los recursos que han surgido para incorporar dinamismo y capacidades al lenguaje HTML. Actualmente es la tecnología más extendida en el enriquecimiento de páginas web del lado del cliente. JavaScript no es un lenguaje orientado a objetos es más bien un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y plantillas de cálculo. Entre las acciones típicas que se pueden realizar en JavaScript se tienen dos vertientes. Por un lado los efectos especiales sobre páginas web y por el otro, JavaScript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que se pueden crear páginas interactivas con programas como calculadoras, agendas o tablas de cálculo (Flanagan, 2002).

### **1.4.3 Entorno Integrado de Desarrollo NetBeans 7.3**

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), es una aplicación de software compuesto por un conjunto de herramientas de programación, consiste en un editor de código, un compilador, un depurador y un constructor de Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés). Los IDE's pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (Belmonte, y otros, 2010).

NetBeans v7.3 es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación debido a un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso (Parra, 2011).

NetBeans tiene una amplia comunidad en constante crecimiento. Provee una estructura para los proyectos y propone un esqueleto para organizar código fuente. El editor conjuntamente integra los lenguajes como PHP, HTML, JavaScript y CSS. Cabe señalar que el referido entorno integrado de desarrollo brinda soporte a Symfony2, obteniendo un grupo de ventajas importantes para el desarrollador frente al marco de trabajo (Comunity, 2011).

### **1.4.5 Lenguaje de Modelado Unificado (UML)**

Para el desarrollo de la aplicación se utilizó UML 2.0 como el lenguaje de soporte a la metodología seleccionada.

El lenguaje de modelado unificado (Unified Modeling Language, por sus siglas en inglés UML) es un lenguaje estándar para el modelado de software, que permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. Este se puede aplicar en el desarrollo

de software, entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, como es el caso de Open UP, aunque no especifica en sí mismo qué metodología o proceso usar (Rumbaugh, 2000).

Entre sus principales características se encuentran (Fornaris, y otros, 2012):

- Es independiente del proceso, para utilizarlo de forma óptima se debe usar en un proceso que sea dirigido por Casos de Uso, centrado en la arquitectura e iterativo e incremental.
- Permite modelar sistemas utilizando técnicas de programación orientadas a objetos (POO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

#### **1.4.6 Herramienta CASE Visual Paradigm para UML 8.0**

CASE es un acrónimo para (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora). Se considera una herramienta CASE a un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de software, completamente o en alguna de sus fases (UCOL, 2006).

La herramienta CASE que define utilizar la presente investigación es Visual Paradigm para UML 8.0 la cual soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue y contribuye a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Se caracteriza por el uso de un lenguaje estándar común al equipo de trabajo, que facilita la comunicación entre sus integrantes, es una herramienta fácil de instalar y actualizar, genera código para varios lenguajes de programación y exporta en formato HTML, es una tecnología libre y está disponible en varios idiomas. Uno de los elementos importantes que le aportan distinción es la posibilidad que brinda de soportar aplicaciones web (Pressman, 2002).

Esta herramienta permite aumentar la calidad del software, a través de la mejora en el desarrollo y mantenimiento del mismo, de igual forma potencia la reutilización del software y estandarización de la

documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software. Entre sus principales características se encuentran:

- Entorno de creación de diagramas para UML 2.0.
- Diseño centrado en Casos de Uso y enfocado al negocio que genera un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Modelamiento de Base de Datos.
- Ingeniería de Código.
- Disponibilidad en múltiples plataformas

### **1.4.7 Sistema Gestor de Bases de Datos PostgreSQL 9.3**

Es un Sistema Gestor de Bases de Datos relacional orientado a objetos, de carácter libre. Ofrece control de concurrencia multi-versión, soportando casi todas las sintaxis SQL como subconsultas, transacciones y funciones definidas por el usuario, cuenta además con un amplio conjunto de enlaces con lenguajes de programación. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Guerrero, 2009). Entre sus características fundamentales se destacan su alto grado de escalabilidad y la capacidad para el manejo de grandes volúmenes de datos (Group, 2011).

### **1.4.8 PGAdmin III 1.22.1 para el manejo de PostgreSQL**

PgAdmin III 1.22.1 es una plataforma de administración y desarrollo de código abierto para PostgreSQL. La aplicación puede utilizarse en Linux, FreeBSD, Solaris, Mac OSX y Windows para administrar PostgreSQL 7.3 y superiores y funciona en cualquier plataforma. Está diseñada para responder a las necesidades de todos los usuarios, desde escribir simples consultas SQL hasta crear Bases de Datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita su administración.

La conexión con el servidor es posible realizarla a través de TCP/IP o Unix Domain Sockets (en plataformas \*Unix), y puede utilizar encriptado SSL (Security Sockets Liars) para la seguridad. No se

requieren controladores adicionales para comunicarse con el servidor de Base de Datos. Desarrollado por una comunidad de expertos de PostgreSQL en todo el mundo, está disponible en más de una docena de idiomas (PostgreSQL., 2012).

#### **1.4.9 Apache 2.2.22 como servidor web a utilizar**

Apache 2.2.22 constituye una tecnología gratuita de código abierto. Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Permite la creación de sitios web dinámicos mediante el uso de lenguajes de scripting como PHP, JavaScript, Python, entre otros. Se ejecuta en varios sistemas operativos. Posee una arquitectura modular que admite ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos. Tiene una alta configurabilidad en la creación y gestión de logs; y soporta personalizar la respuesta ante los posibles errores que se puedan generar en el servidor (Foundation, 2010a).

### **1.5 Conclusiones Parciales**

El control del trabajo docente metodológico en los departamentos de la UCI es fundamental para garantizar la calidad del proceso docente-educativo en la institución. A partir del estudio de las aplicaciones similares se evidenció que ninguna resuelve el problema planteado en esta investigación, por lo que es necesario la realización de una aplicación informática que contribuya a la gestión de la documentación generada por el trabajo docente metodológico. El empleo de las herramientas y tecnologías seleccionadas aporta eficacia y agilidad durante el proceso de desarrollo, de igual manera impulsan la soberanía tecnológica propuesta por el país y la Universidad de las Ciencias Informáticas. La escalabilidad y mantenimiento del resultado que se pretende se ven beneficiadas por la selección antes descrita.

### Capítulo 2: Análisis y diseño del Sistema de Gestión de la Información del Trabajo Docente Metodológico

En el capítulo que se presenta a continuación se exponen las principales características y cualidades de la solución a implementar, se describe y detalla todo lo referente al modelo del negocio, se identifican los Requisitos Funcionales y No Funcionales; también se obtiene el diagrama de los Casos de Uso del Sistema conjunto a su descripción textual, así como los diagramas correspondientes a la representación del diseño. Como parte de la definición de los elementos arquitectónicos fundamentales se determina el estilo y el patrón arquitectónico a emplear además se relacionan los patrones de diseño que guiarán la construcción del sistema propuesto. Al concluir este capítulo se contará con una comprensión de los problemas actuales del negocio y una idea pertinente del sistema a desarrollar.

#### 2.1 Modelo del negocio

El modelo de negocio es una técnica para comprender los procesos de negocio de la organización objeto de estudio. Este modelo representa un sistema desde la perspectiva de su uso, y esquematiza cómo proporciona valor a sus usuarios, el mismo tiene como objetivo describir los procesos, existentes u observados, con el propósito de comprenderlos. Se especifican aquí qué procesos del negocio que soportará el sistema y se identifican los objetos del dominio o del negocio que estén implicados, este modelo establece las competencias que se requieren de cada proceso: sus trabajadores, sus responsabilidades y las operaciones que llevan a cabo (Jacobson, y otros, 2007).

##### 2.1.1 Reglas del negocio a considerar

1. Una facultad tiene un grupo de departamentos que a su vez tiene un grupo de disciplinas las cuales incluyen un conjunto de asignaturas.
2. Existe una jerarquía de dirección para el trabajo metodológico en la facultad, la cual se encabeza por el **vicedecano de formación**, el cual responde por el trabajo de **los jefes de departamento**, los cuales tienen bajo su responsabilidad a los **jefes de disciplina** y estos últimos a los **jefes de asignatura**, dando lugar a niveles de trabajo.
3. Un jefe de alguna de las instancias docentes puede comportarse simultáneamente como responsable de una instancia inferior a la cual él dirige.
4. El Plan de trabajo docente metodológico (en lo adelante PTDM) a cada nivel solo puede ser creado por el responsable del nivel en cuestión.
5. Los controles a clase solo podrán ser realizados por profesores que tengan mayor categoría

docente que el profesor que está siendo controlado.

6. Los controles a clase se adjuntan solo al plan de trabajo docente metodológico del departamento.
7. Las evidencias de las actividades docentes metodológicas solo pueden ser creadas y entregadas por sus responsables.
8. El cumplimiento del plan docente metodológico a cualquier instancia está dado por el cumplimiento de las actividades planificadas en él.
9. Solo se puede crear un plan de trabajo docente metodológico por curso a cualquier instancia.
10. El claustro del departamento está compuesto por el claustro de las asignaturas y disciplinas que le pertenecen.

### 2.1.2 Actores y trabajadores del negocio

En el modelo de Casos de Uso del Negocio (CUN) se describen los procesos de un negocio y su interacción con elementos externos (actores). En él se reflejan las funciones que el negocio va a realizar, así mismo un (trabajador) del negocio es una abstracción de una persona o grupo de personas, una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades. A continuación en la tabla 1 y 2 se determinan y justifican los actores y trabajadores que intervienen en el negocio a modelar.

Tabla. 1 Actores del Negocio

Actores del Negocio	Justificación
Vicedecano de Formación	Es el principal responsable del trabajo docente metodológico en la facultad. Directivo que debe recibir toda la información de la actividad metodológica en la facultad.
Jefe de Departamento	Es el principal responsable del trabajo docente metodológico en el departamento. Directivo que debe recibir toda la información de la actividad metodológica en el departamento.
Jefe de Disciplina	Es el principal responsable del trabajo docente metodológico en la disciplina. Directivo que debe recibir toda la información de la actividad metodológica de la disciplina.

Tabla. 2 Trabajadores del Negocio

Trabajadores del Negocio	Justificación
Vicedecano de Formación	Responsable de elaborar el PTDM de la facultad.
Jefe de Departamento	Responsable de elaborar el PTDM del Departamento y tributar todo tipo de información sobre las actividades metodológicas realizadas.
Jefe de Disciplina	Responsable de elaborar el PTDM de la Disciplina y tributar todo tipo de información sobre las actividades metodológicas realizadas.
Jefe de Asignatura	Responsable de elaborar el PTDM de la Asignatura y tributar todo tipo de información sobre las actividades metodológicas realizadas.

### 2.1.3 Diagrama de Casos de Uso del Negocio

El Diagrama de Casos de Uso del Negocio representa gráficamente a todos los procesos del negocio y su interacción con los actores del negocio. En la fig. 1 se muestra el diagrama de Casos de Uso del Sistema de Gestión de la Información del Trabajo Docente Metodológico enfocado en el análisis de la solución actual.

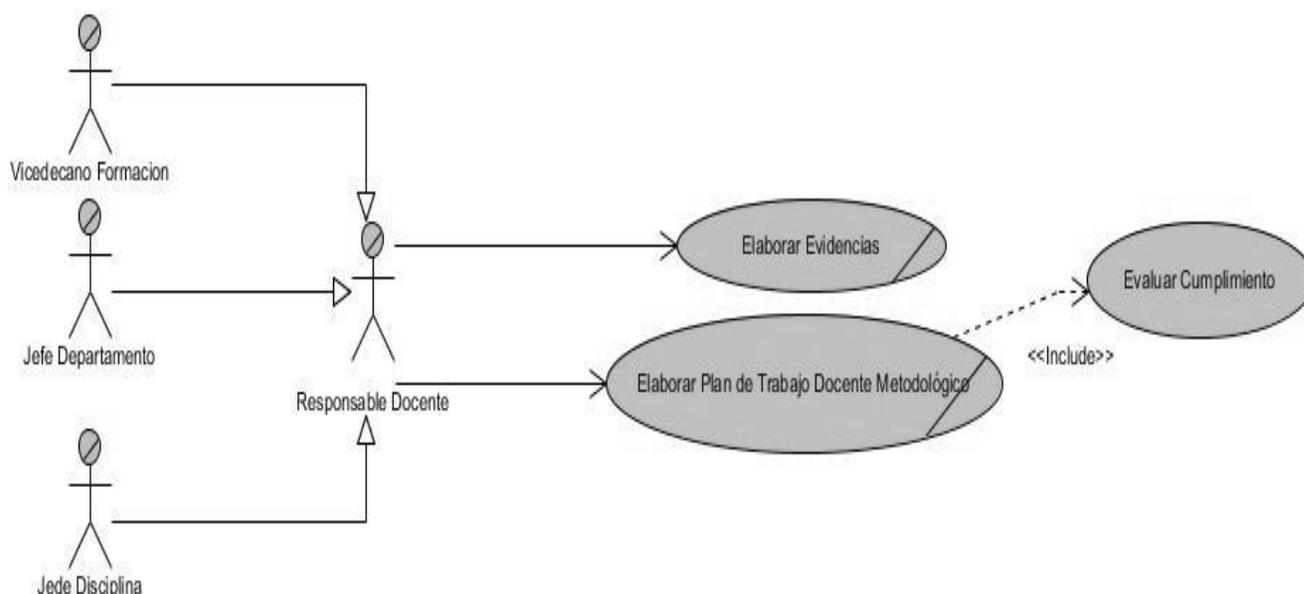


Fig. 1 Diagrama de Casos de Uso del Negocio. Sistema de Gestión de la Información del Trabajo Docente Metodológico

### 2.1.4 Descripción textual de los Casos de Uso del Negocio

A continuación en la tabla 3 se realiza la descripción textual del Caso de Uso del Negocio: Elaborar PTDM

Tabla. 3 Descripción textual del Caso de Uso del Negocio: Elaborar PTDM

Nombre del Caso de Uso del Negocio:	Elaborar Plan de Trabajo Docente Metodológico.	
Actores del negocio:	Responsable docente (inicia).	
Propósito:	Elaborar el plan de trabajo docente metodológico	
Resumen:	El Caso de Uso inicia cuando un responsable docente solicita la elaboración del PTDM del curso actual, continúa con la elaboración de los acápites de dicho plan y agregando al mismo un grupo de actividades y sus responsables para luego ser entregado a la instancia docente superior (IDS).	
Casos de Uso asociados:	Evaluar Cumplimiento, Elaborar Evidencia.	
Flujo normal de los eventos		
Acción del actor:	Respuesta del negocio:	
1 El responsable docente solicita a su <i>instancia inmediata inferior</i> la elaboración del PTDM.	2 Elabora los acápites del PTDM. 3 Agrega actividades al plan conteniendo sus respectivos responsables. 4 Entrega a la IDS el PTDM elaborado.	
5 El responsable docente recibe el PTDM revisando su pertinencia. Finaliza el Caso de Uso.		
Prioridad:	Alta.	
Mejoras:	<ul style="list-style-type: none"> <li>• La elaboración de los acápites se realizará con agilidad y eficacia.</li> <li>• El seguimiento respecto al cumplimiento de las actividades incluidas en el PTDM facilitará la obtención de estadísticas respecto a su cumplimiento.</li> </ul>	
Cursos alternos:		
5 El responsable docente revisa el PTDM determinando que la información reflejada en el mismo no es pertinente y solicitando la reelaboración del mismo.		
Casos de Uso Incluidos:	Evaluar cumplimiento	

### 2.1.5 Diagrama de actividades del Caso de Uso del Negocio: Elaborar plan de trabajo docente metodológico

A continuación en la fig. 2 se detalla el Caso de Uso Elaborar PTDM a través del diagrama de actividades. Un diagrama de actividad describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio (Larman, 2005).

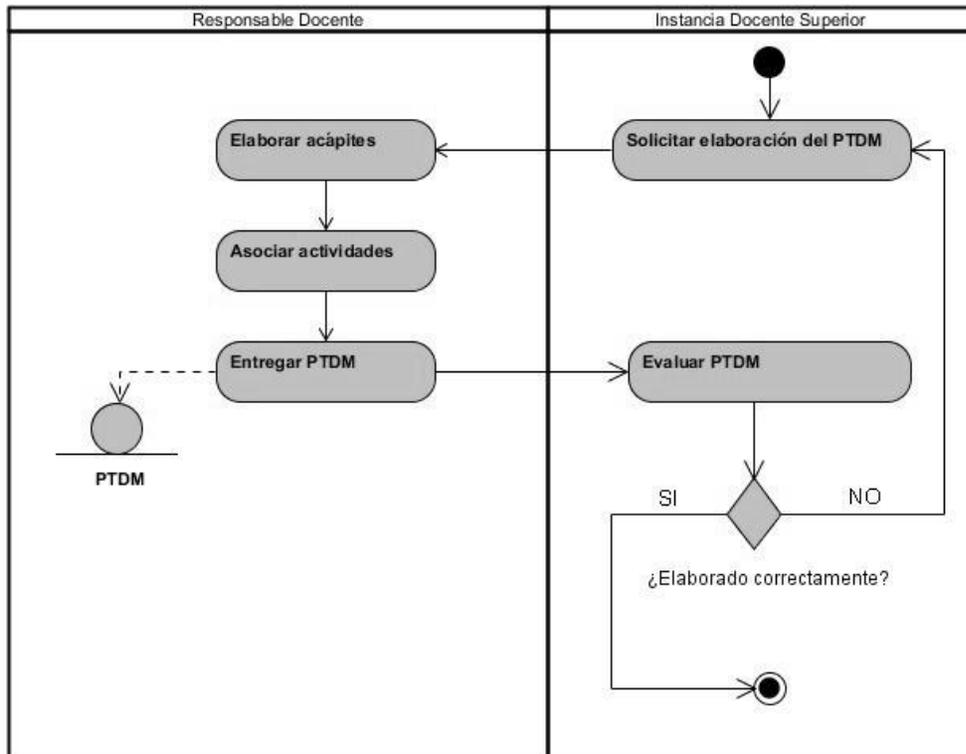


Fig. 2 Diagrama de actividades del Caso de Uso del Negocio: Elaborar PTDM

## 2.2 Requisitos de Software

Según (Pressman, 2008) los requisitos de software son las necesidades de los clientes, los servicios que los usuarios desean que proporcione el sistema de desarrollo y las restricciones en las que debe operar. Los requisitos se dividen en Funcionales y No Funcionales y muestran las capacidades o condiciones que el sistema debe cumplir y las propiedades o cualidades que el producto debe tener.

### 2.2.1 Requisitos Funcionales

Los requisitos funcionales (RF) son declaraciones de los servicios que deben proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requisitos funcionales de los sistemas también pueden

declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005).

El sistema debe ser capaz de:

- |  |  |
|--|--|
| RF 1. Crear PTDM.                        | RF 26. Crear disciplina.                         |
| RF 2. Modificar PTDM.                    | RF 27. Modificar disciplina.                     |
| RF 3. Eliminar PTDM.                     | RF 28. Eliminar disciplina.                      |
| RF 4. Listar PTDM.                       | RF 29. Listar disciplina.                        |
| RF 5. Crear ADM.                         | RF 30. Crear asignatura.                         |
| RF 6. Modificar ADM.                     | RF 31. Modificar asignatura.                     |
| RF 7. Eliminar ADM.                      | RF 32. Eliminar asignatura.                      |
| RF 8. Listar ADM.                        | RF 33. Listar asignatura.                        |
| RF 9. Crear otras actividades.           | RF 34. Crear profesor.                           |
| RF 10. Modificar otras actividades.      | RF 35. Modificar profesor.                       |
| RF 11. Eliminar otras actividades.       | RF 36. Eliminar profesor.                        |
| RF 12. Listar otras actividades.         | RF 37. Listar profesor.                          |
| RF 13. Filtrar actividad.                | RF 38. Autenticar usuario.                       |
| RF 14. Evaluar actividad.                | RF 39. Crear Evidencia.                          |
| RF 15. Evaluar Cumplimiento del PTDM.    | RF 40. Modificar Evidencia.                      |
| RF 16. Notificar actividades pendientes. | RF 41. Eliminar Evidencia.                       |
| RF 17. Exportar documentos a pdf.        | RF 42. Listar Evidencia.                         |
| RF 18. Crear facultad.                   | RF 43. Asociar evidencia.                        |
| RF 19. Modificar facultad.               | RF 44. Crear plan de controles a clase.          |
| RF 20. Eliminar facultad.                | RF 45. Modificar plan de controles a clase.      |
| RF 21. Listar facultad.                  | RF 46. Eliminar plan de controles a clase.       |
| RF 22. Crear departamento.               | RF 47. Listar plan de controles a clase.         |
| RF 23. Modificar departamento.           | RF 48. Crear informe de resultado de examen.     |
| RF 24. Eliminar departamento.            | RF 49. Modificar informe de resultado de examen. |
| RF 25. Listar departamento.              | RF 50. Eliminar informe de resultado de examen.  |
|  | RF 51. Listar informe de resultado de examen.    |

### **2.2.2 Requisitos No Funcionales**

Los requisitos no funcionales (RNF) son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requisitos no

funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (Sommerville, 2005).

Luego de analizado los aspectos fundamentales del negocio y el ámbito de la organización, se detallan a continuación los siguientes RNF bajo la nomenclatura propuesta por (Sommerville, 2005):

**Requisitos de usabilidad:** El sistema está orientado a usuarios con un dominio en el manejo de tecnologías, pero el mismo se caracterizará por tener un diseño sencillo e intuitivo, de acuerdo con las políticas de identidad marcaría definidas por la UCI, fundamentalmente bajo la marca XABAL, dirigida a sistemas para la administración pública.

**Requisitos de fiabilidad:** El sistema se encargará de controlar los diferentes accesos y privilegios de los usuarios, además de identificar al usuario antes de que pueda realizar cualquier acción. La información y funcionalidades del sistema estarán disponibles de forma permanente.

**Requisitos de eficiencia:** El tiempo de respuesta de la aplicación estará dado por la cantidad de información a procesar de acuerdo a la funcionalidad solicitada. Al igual que el tiempo de respuesta la velocidad de procesamiento de la información y su disposición al usuario estarán condicionadas por la cantidad de datos a manejar. La aplicación debe desempeñarse de manera correcta para un mínimo de 40 usuarios concurrentes con un tiempo de respuesta no superior a los 5 segundos.

Ante condiciones de sobreexplotación del sistema el mismo debe mantenerse estable, mostrando parámetros de respuesta cercanos a los 15 segundos para cifras de usuarios concurrentes no mayores de 500.

**Portabilidad:** El sistema debe desarrollarse de forma tal que cuente con la posibilidad de ser multiplataforma.

### Requisitos de interfaz

#### Interfaces de hardware

Tabla. 4 Interfaces de Hardware. Características

Para las computadoras cliente:	Para los servidores:
<ul style="list-style-type: none"><li>• 512 MB de RAM.</li><li>• Al menos 40 GB de disco duro.</li><li>• Procesador 512 MHz</li></ul>	<ul style="list-style-type: none"><li>• 4 GB de RAM.</li><li>• Al menos 40 GB de disco duro para el servidor de Base de Datos.</li><li>• Procesador con al menos 2 núcleos.</li></ul>

#### Interfaces de software

La construcción de la aplicación funcionará bajo los conceptos de arquitectura de despliegue: cliente/servidor. Por tanto, el servidor del usuario final debe tener como requisitos mínimos de software:

Para las Computadoras cliente: Navegador: Cualquiera que cumpla con los estándares de la w3c<sup>1</sup>, soporte html5 y la biblioteca JavaScript en su versión 1.9, además de un visor de pdf. Preferiblemente Mozilla Firefox 17.0, Chrome 20.0, Zafari 4.0, Opera 25.0 o versiones superiores a estas.

Para los Servidores: Sistema operativo GNU/Linux Ubuntu Server 14.04. Servidor Web Apache 2.2.22, con módulo PHP 5.5.9 configurado con la extensión “pgsql” incluida. PostgreSQL 7.3 o superior como Sistema Gestor de Bases de Datos. La biblioteca “wkhtmltox” debe ser incluida para la obtención de documentos en formato pdf.

**Requisitos de seguridad:** El sistema debe ser accedido solo mediante comunicación segura a través de la red, haciendo uso del protocolo https. Las credenciales de acceso de los usuarios deben ser comprobadas mediante el directorio LDAP de la UCI y las mismas no deben persistirse de ninguna forma dentro de la aplicación. El acceso a las funcionalidades de la solución propuesta debe estar mediado por una jerarquía de roles cumpliendo el principio de mínimo privilegio.

**Requisitos legislativos:** El sistema debe ajustarse y regirse por la ley, decretos leyes, decretos, resoluciones y manuales (órdenes) establecidos, que norman los procesos que serán automatizados. La totalidad de las herramientas de desarrollo deben ser libres. Como producto, el sistema se distribuye amparado bajo las normativas legales establecidas en el registro comercial emitido por las entidades

<sup>1</sup> El W3C es una comunidad internacional que incluye un personal de tiempo completo, expertos de la industria, y varias organizaciones miembros. Estos grupos trabajan conjuntamente para desarrollar estándares para la World Wide Web.

jurídicas de la Universidad de las Ciencias Informáticas.

### 2.3 Modelo del Sistema

Según (Larman, 2005) el modelo de Casos de Uso permite describir los RF del sistema, dando lugar a un acuerdo entre el cliente y los desarrolladores de la aplicación. Proporciona una explicación clara y consistente de lo que debería hacer el software, de modo que el modelo se use a lo largo del proceso de desarrollo. Posibilita que se obtenga una base para realizar verificaciones del producto informático, siendo la entrada fundamental para el análisis, el diseño y las pruebas.

Los Casos de Uso son descripciones funcionales del sistema que permiten definir los límites del software y las relaciones entre la aplicación y el entorno; describen cómo los actores pueden usar un sistema. Especifican una secuencia de acciones que debe devolver algún resultado de valor a un actor (Schmuller, 2010).

#### 2.3.1 Actores del Sistema

Un actor es una entidad externa al sistema representada por un ser humano, una máquina o un software que interactúa con el sistema. Representa un tipo particular de usuario del negocio más que un usuario físico, debido a que varios usuarios físicos pueden realizar el mismo papel en relación al negocio (Schmuller, 2010). En la tabla 5 se detallan los actores del sistema:

Tabla. 5 Actores del Sistema

Actor	Descripción
Administrador	Es el usuario del sistema, con permisos para gestionar la información referente a la estructura docente de la facultad.
Vicedecano	Responsable dentro del sistema, de la gestión del claustro de profesores, tiene acceso a todas las funcionalidades acorde al nivel que el mismo representa exceptuando las relacionadas con el administrador.
Jefe de Departamento	Responsable dentro del sistema, de la gestión del plan de controles a clase, tiene acceso a todas las funcionalidades acorde al nivel que el mismo representa exceptuando las relacionadas con el administrador y la gestión del claustro.
Jefe de Disciplina	Usuario que asume todas las funcionalidades del Jefe de Asignatura y Profesor acorde al nivel que el mismo representa.
Jefe de Asignatura	Responsable dentro del sistema acorde al nivel que representa de: la gestión de informes de resultado de exámenes, la gestión de actividades y la gestión de

	PTDM, así mismo puede gestionar evidencias.
Profesor	Usuario con acceso a la gestión de evidencias.

## 2.3.2 Diagrama de Casos de Uso del Sistema

El diagrama de Casos de Uso del Sistema (en lo adelante CUS) explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos y los casos de uso. Estos últimos se muestran en óvalos y los actores son figuras estilizadas. Hay líneas de comunicaciones entre los casos y los actores; las flechas indican el flujo de la información o el estímulo (Larman, 2005).

Seguidamente se presenta en la fig. 3 el diagrama de Casos de Uso del Sistema de Gestión de la Información del Trabajo Docente Metodológico.

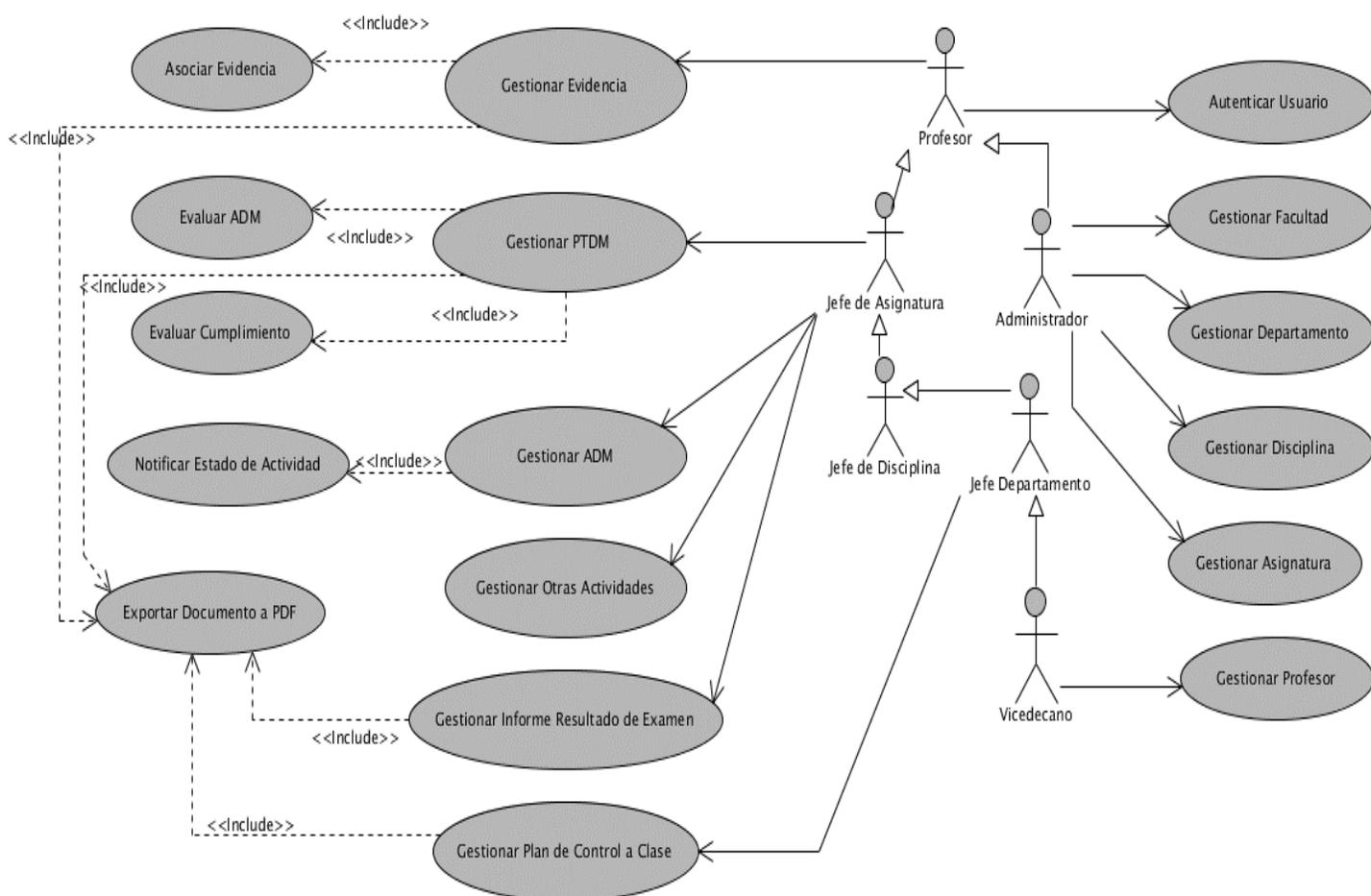


Fig. 3 Diagrama de Casos de Uso del Sistema. Sistema de Gestión de la Información del Trabajo Docente Metodológico

A continuación en la tabla 6 se especifica la descripción textual del Caso de Uso del Sistema: “Gestionar PTDM”.

Tabla. 6 Descripción textual del Caso de Uso del sistema: Gestionar PTDM

<b>Caso de Uso:</b>	Gestionar plan de trabajo docente metodológico.	
<b>Actores:</b>	Vicedecano, Jefe de Departamento, Jefe de Disciplina, Jefe de Asignatura (inician).	
<b>Propósito:</b>	Este Caso de Uso se lleva a cabo con el objetivo de Gestionar planes de trabajo docente metodológico.	
<b>Resumen:</b>	Este Caso de Uso se inicia cuando el usuario selecciona la opción de crear un plan de trabajo docente metodológico y culmina con la realización de una de las siguientes operaciones sobre un plan de trabajo docente metodológico: <i>Crear, Editar, Eliminar o Listar</i> .	
<b>Precondiciones:</b>	El actor debe estar autenticado en el sistema y debe tener permisos para realizar las mencionadas operaciones. Debe seleccionarse el plan de trabajo docente metodológico sobre el cual se pretende realizar una de las siguientes acciones: <i>Modificar, Eliminar o Listar</i> . Para realizar una de estas acciones debe existir en el sistema al menos un plan.	
<b>Referencias:</b>	RF 1, RF 2, RF 3, RF 4	
<b>Prioridad:</b>	Alta	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
1 El Caso de Uso inicia con la selección de la sección gestionar PTDM al nivel correspondiente.	2 Muestra una interfaz con las opciones crear, editar y mostrar En el caso de seleccionar: <ul style="list-style-type: none"> <li>✓ “Crear plan de trabajo docente metodológico” ver sección <i>Crear plan de trabajo docente metodológico</i>.</li> <li>✓ “Editar plan de trabajo docente metodológico” ver sección <i>editar plan de trabajo docente metodológico</i>.</li> <li>✓ “Mostrar plan de trabajo docente metodológico” ver sección <i>Mostrar plan de trabajo docente metodológico</i>.</li> </ul>	
3 El usuario selecciona la acción que desea realizar.		
<b>Prototipo de Interfaz</b>		
<b>Sección Crear plan de trabajo docente metodológico</b>		
<b>Flujos Básicos</b>		
<b>Acción del Actor:</b>	<b>Respuesta del Sistema:</b>	
	1 Se muestra una interfaz para introducir los datos	

	referentes a la gestión del PTDM.
2 Se introducen los datos necesarios y selecciona el botón <i>crear</i> .	<p>3 Se verifica que no existan campos vacíos.</p> <p>3.1 Verifica que los datos introducidos sean válidos.</p> <p>3.2 Verifica la existencia de archivo en formato documento.</p>
	4 Se crea un nuevo PTDM, muestra el mensaje: “Acción realizada con éxito”.

### Prototipo de Interfaz

### Flujos Alternos

Acción del Actor:	Respuesta del Sistema:
	3 Muestra un mensaje informando que existen campos vacíos.
	<p>3.1 Muestra un mensaje informando que existe algún campo inválido.</p> <p>3.2 Si existe un archivo en formato documento pueden existir campos vacíos y si no existe, los campos deben estar correctamente llenados.</p>

### Sección Editar plan de trabajo docente metodológico

#### Flujos Básicos

Acción del Actor:	Respuesta del Sistema:
	<p>1 Muestra una interfaz con los campos editables del plan de trabajo docente metodológico y la opción eliminar plan de trabajo docente metodológico.</p> <p>✓ En caso de seleccionar eliminar ver sección eliminar</p>
2 El usuario determina los datos que desea editar y selecciona el botón <i>editar</i> .	<p>3 Verifica que no existan campos vacíos.</p> <p>3.1 Verifica que los datos introducidos sean válidos.</p> <p>3.2 Si existe un archivo en formato digital pueden existir campos vacíos y si no existe, los campos deben estar correctamente llenados.</p>
	4 Edita el plan de trabajo docente metodológico.

### Prototipo de Interfaz

**Editar PT-DM-Departamento [Programacion y Sistemas Digitales]**

---

Datos del PT-DM-Departamento [Programacion y Sistemas Digitales]:

Nombre :  Fecha :

Fundamentacion :

File Edit View Format

Formats **B** *I* [List of icons]

En el acto de apertura la especialista de la Dirección de Extensión Universitaria en la institución, Gladys Martínez Iglesias, recorrió la Atenas del primer cuarto de siglo de nuestra era. Invitó a los presentes, desde el sabio Diógenes y su tinaja de la virtud, a la contemplación de más de 170 obras de diversos géneros en representación de nueve centros de Educación Superior de La Habana que se exhiben en el Complejo Recreativo-Cultural del Docente 6.

Flujos Alternos	
Acción del Actor:	Respuesta del Sistema:
	<b>3</b> Muestra un mensaje informando que existen campos vacíos.
	<b>3.1</b> Muestra un mensaje informando que existe algún campo inválido. <b>3.2</b> Si existe un archivo en formato digital pueden existir campos vacíos y si no existe, los campos deben estar correctamente llenados.
Sección mostrar plan de trabajo docente metodológico	
Flujos Básicos	
Acción del Actor:	Respuesta del Sistema:
	<b>1</b> Muestra la interfaz que contiene todos los detalles asociados a un plan de trabajo docente metodológico. La posibilidad de descargar el archivo en formato digital en caso de ser asociado. Además muestra la opción eliminar plan de trabajo docente metodológico.  ✓ En caso de seleccionar eliminar ver sección eliminar
Prototipo de Interfaz	

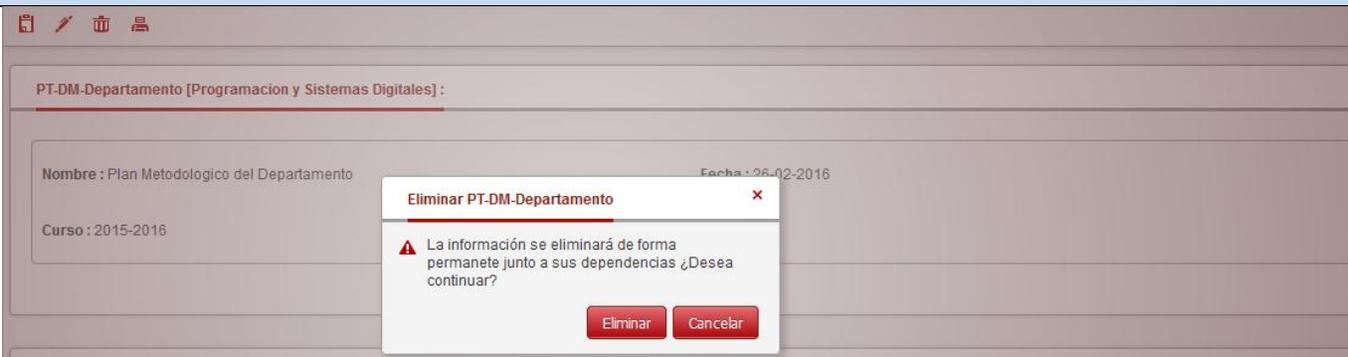
**PT-DM-Departamento [Programacion y Sistemas Digitales]:**

Nombre : Plan Metodologico del Departamento Fecha : 26-02-2016

Curso : 2015-2016

Fundamentación :

En el acto de apertura la especialista de la Dirección de Extensión Universitaria en la institución, Gladys Martínez Iglesias, recorrió la Atenas del primer cuarto de siglo de nuestra era. Invitó a los presentes, desde el sabio Diógenes y su tinaja de la virtud, a la contemplación de más de 170 obras de diversos géneros en representación de nueve centros de Educación Superior de La Habana que se exhiben en el Complejo Recreativo-Cultural del Docente 6.

Flujos Alternos	
Acción del Actor:	Respuesta del Sistema:
<b>Sección eliminar plan de trabajo docente metodológico</b>	
Flujos Básicos	
Acción del Actor:	Respuesta del Sistema:
	<b>1</b> Muestra una petición de confirmación y una advertencia de que se eliminaran de forma permanente los datos.
<b>2</b> El actor confirma mediante la selección del botón aceptar.	<b>3</b> Elimina el plan de trabajo docente metodológico.
Prototipo de Interfaz	
	
Flujos Alternos	
Acción del Actor:	Respuesta del Sistema:
<b>2</b> El actor selecciona la opción cancelar	<b>3</b> Mantiene la interfaz mostrar plan de trabajo docente metodológico.
Poscondiciones:	Queda creado, modificado o eliminado el plan de trabajo docente metodológico.
Casos de Uso Incluidos:	Evaluar cumplimiento, Evaluar ADM, Exportar documento a pdf.

### 2.3.3 Patrones de Casos de Uso utilizados

Los patrones de Casos de Uso aplicados en la construcción del diagrama de Casos de Uso del Sistema fueron los siguientes:

#### CRUD Completo

Este patrón consiste en un Caso de Uso para administrar información, nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, Modificar, eliminar y listar. El mismo se manifiesta en los Casos de Uso: Gestionar plan de trabajo docente metodológico (Schmuller, 2010).

#### Extensión Concreta o Inclusión

Una relación de inclusión es una relación desde un Caso de Uso base a un Caso de Uso de inclusión, que

especifica cómo el comportamiento definido para el Caso de Uso de inclusión se incarta explícitamente dentro del comportamiento definido para el Caso de Uso base. Se utiliza para dividir partes de un flujo de trabajo de cuyos resultados, y no del método para obtenerlo, depende el Caso de Uso base. Se puede hacer esta partición si simplifica la comprensión del Caso de Uso base o si el comportamiento separado puede reutilizarse en otros Casos de Uso. Gestionar PTDM contiene a los Casos de Uso: Evaluar cumplimiento, Evaluar ADM y Exportar documentos a pdf (Schmuller, 2010).

### **Generalización/Especialización entre actores**

Una relación de generalización de una clase hija de actor a otra clase padre de actor indica que el hijo hereda el rol que la clase padre puede jugar respecto a un Caso de Uso. Varios actores pueden jugar el mismo rol en un Caso de Uso particular, como se pone de manifiesto en el diagrama de Casos de Uso del Sistema con la generalización/especialización establecida entre los actores del sistema “administrador” y el “Profesor” ya que el administrador realiza sus funciones específicas pero a ellas se le agregan otras funcionalidades como profesor (Schmuller, 2010).

## **2.4 Elementos fundamentales de la arquitectura**

La arquitectura de software es la estructura o estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Debemos tener en cuenta que la arquitectura no es el sistema operativo es una representación que permite que un ingeniero del software analice la efectividad del diseño para cumplir con los requisitos establecidos, considere opciones arquitectónica en una etapa en que aún resulta relativamente fácil hacer cambios al diseño, reduzca los riesgos asociados con la construcción del software (Pressman, 2002).

### **2.4.1 Patrones arquitectónicos y de diseño**

#### **Estilo Arquitectónico**

Un Estilo Arquitectónico es una transformación impuesta al diseño de todo un sistema. El objetivo es establecer una arquitectura para todos los componentes del sistema. En caso de que una arquitectura existente se valla a someter a la reingeniería, la imposición de un estilo arquitectónico desembocará en cambios fundamentales en la estructura del software, incluida una reasignación de la funcionalidad de los componentes (Pressman, 2002).

### **Patrón Arquitectónico**

Un patrón arquitectónico al igual que un estilo, impone una transformación en el diseño de una arquitectura. Sin embargo, un patrón difiere de un estilo de varios elementos fundamentales: el alcance de un patrón es menor, ya que se concentra en un aspecto en lugar de hacerlo en toda la arquitectura, un patrón impone una regla sobre la arquitectura, pues describe la manera en que el software maneja algún aspecto de su funcionalidad al nivel de la infraestructura. Los patrones arquitectónicos tienden a abarcar aspectos específicos del comportamiento dentro del contexto de la arquitectura. Los patrones se usan junto con un estilo arquitectónico para determinar la forma de la estructura general de un sistema (Pressman, 2002).

### **Patrón Modelo-Vista-Controlador**

El patrón de arquitectura de software separa los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello propone la construcción de tres componentes distintos: el modelo, la vista y el controlador. Este patrón basa sus principios en las ideas de reutilización de código y la separación de conceptos, con el objetivo de facilitar las tareas de desarrollo y mantenimiento de las aplicaciones.

El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requisitos de información sobre su estado y responde a instrucciones de cambiar el estado. La vista maneja la visualización de la información. Mientras que el controlador responde a eventos (usualmente acciones del usuario) e invoca peticiones al “modelo” cuando se hace alguna solicitud sobre la información (Garlan, y otros, 1996).

La arquitectura del sistema está sustentada en el patrón de diseño antes explicado, propiciando mantener un código limpio, fácil de implementar y de probar. La fig. 4 Diagrama de paquetes de diseño del Caso de Uso Gestionar PTDM muestra la arquitectura que implementa la propuesta de solución:

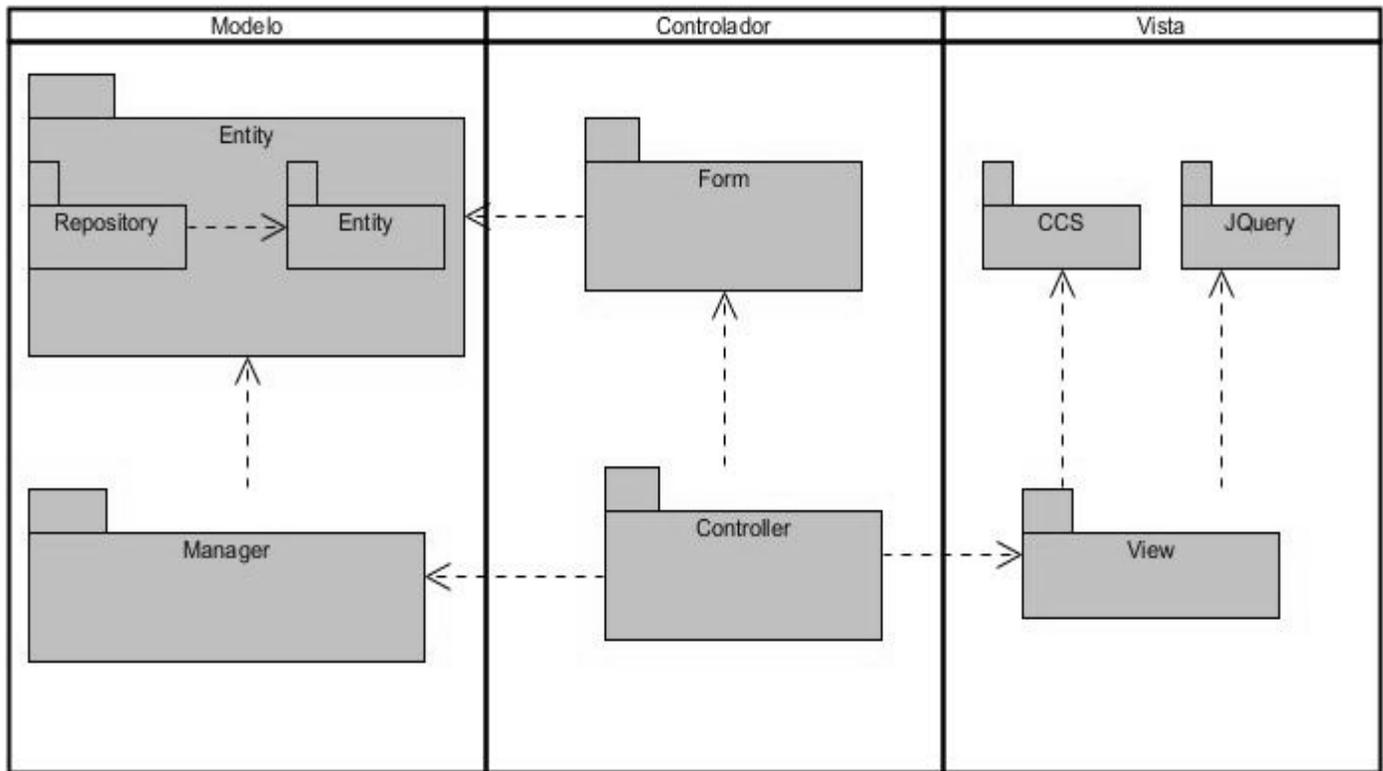


Fig. 4 Diagrama de paquetes de diseño del Caso de Uso: Gestionar PTDM

La capa correspondiente a la Vista contiene dentro del paquete “View” los archivos de extensión “.twig” donde se especifica el título de la página que se gestiona. Contiene las páginas y plantillas de la aplicación, además, los paquetes “Jquery” y “CSS” engloban los archivos JavaScript y las hojas de estilo respectivamente. La capa Controlador engloba los controladores del sistema, dentro del paquete “Controller” se localizan las clases encargadas de manejar la lógica de control y se definen en ellas las funcionalidades que son invocadas, cuando el usuario hace alguna solicitud al sistema a través de acciones sobre los componentes de la interfaz; dentro del paquete “Form” se definen las clases que corresponden a los formularios. El paquete “Entity” por su parte contiene clases que modelan las Base de Datos, análogamente el paquete “Repository” incluye las clases responsables de los mecanismos de selección de datos.

### Patrones de Diseño

Los patrones de diseño proveen un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular. Constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos.

### Patrones generales de software para asignación de responsabilidades utilizados (GRASP)

**Experto:** Este patrón se encarga de asignar las responsabilidades a las clases de acuerdo a la información que contienen las mismas cumpliendo así un principio básico e intuitivo de la programación orientada a objetos (Larman, 2005). En la presente solución se utiliza el mencionado patrón en la clase entidad un ejemplo de esto es cuando se elabora un plan de trabajo docente metodológico en los diferentes niveles.

**Creador:** Este patrón guía la asignación de responsabilidades relacionada con la creación de objetos. (Larman, 2005). Se observa en el sistema propuesto, dentro de la capa controlador correspondiente a cada uno de los casos de uso la cual distribuye las responsabilidades de comunicación con las instancias de la capa de presentación, asignándole a un grupo de clases la responsabilidad de crear instancias de las clases que sirven la información necesaria para su manejo, identificando así quien debe ser el responsable de esta actividad en cada caso.

**Controlador:** El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, manejando los eventos de entrada de dicha interfaz (Larman, 2005). En la aplicación se manifiesta el uso de este patrón en la creación de controladores responsables del manejo de eventos para cada una de las principales áreas en las que se dividen dichas entradas, de acuerdo al marco del negocio y las funcionalidades que el sistema debe implementar, como es el caso del controlador: “ProfesionalController”

**Alta Cohesión:** Este patrón persigue la idea de que los elementos de un componente colaboran para producir algún comportamiento bien definido, preferentemente único en el sistema (Larman, 2005). Cada uno de los casos de uso contiene los componentes modelo, vista, controlador, los mismos trabajan en alta cohesión colaborando entre sí, al mismo tiempo tratan de inmiscuirse lo menos posible en el comportamiento de los componentes de otros casos de uso.

**Bajo Acoplamiento:** Este patrón propone la asignación de responsabilidades de manera tal que la dependencia entre una clase y otra sea la menor posible, de tal forma que se potencie la reutilización y se mitiguen los efectos que puedan producir en una, la realización de cambios en la otra (Larman, 2005). Las clases que se encuentran en la capa Modelo no presentan fuertes dependencias o relaciones directas con las clases implícitas dentro de la capa Controlador o Vista, de esta manera es posible realizar transformaciones en estas independientemente al resto del sistema; asimismo, la forma en la que se maneja la relación entre las clases que pertenecen a la capa Controlador y Vista posibilitan agregar o

retirar funcionalidades sin provocar cambios traumáticos para el sistema, favoreciendo la reutilización y el mantenimiento del mismo.

### Patrones GOF utilizados

**Agente Remoto:** Utilizado en el caso de que el sistema en cuestión deba comunicarse con un servicio externo (Larman, 2005). Su utilización en el sistema tiene como objetivo establecer la comunicación con el servicio “Ldap” brindado por la Universidad de las Ciencias Informáticas, para la obtención de datos referentes a los usuarios de la misma y sus credenciales, en busca de complementar la autenticación de usuarios en el sistema, evidenciándose el patrón en la creación de la clase: “Ldap” responsable del chequeo de credenciales y la autenticación en el sistema.

**Singleton:** Este patrón es puesto en práctica con el fin de garantizar que una clase solo tenga una única instancia, proporcionando un punto de acceso local a la misma (Larman, 2005). El uso de este patrón viene dado por el acceso dentro de la aplicación del “EntityManager”, instancia de la clase Manager, la cual se encarga del manejo de acceso a los métodos relacionados con las clases entidades del sistema, es utilizado siempre desde la capa Controlador.

**Decorador:** Este patrón es de tipo estructural y permite añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender funcionalidad (Larman, 2005). En el trabajo con plantillas “twig” se evidencia el uso de este patrón, ya que la mayoría de las plantillas en el sistema heredan de la plantilla “base.html.twig”, pero ellas se redefinen mediante elementos que constituyen otros objetos. Se aplica con la intención de proporcionar una forma flexible de introducir o eliminar funcionalidades a las páginas de la aplicación, solo estableciendo los bloques que pueden ser redefinidos sin modificar su apariencia externa o su función.

### 2.5 Modelo de Diseño

El modelo del diseño se encarga de describir la realización física de los Casos de Uso y se corresponde directamente con los elementos físicos del ambiente de implementación. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y/o subsistemas (Jacobson, y otros, 2007).

### 2.5.1 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación y permiten modelar la vista de diseño del sistema. A continuación se representa en la fig. 5 el diagrama de clases del diseño para el Caso de Uso Gestionar PTDM.

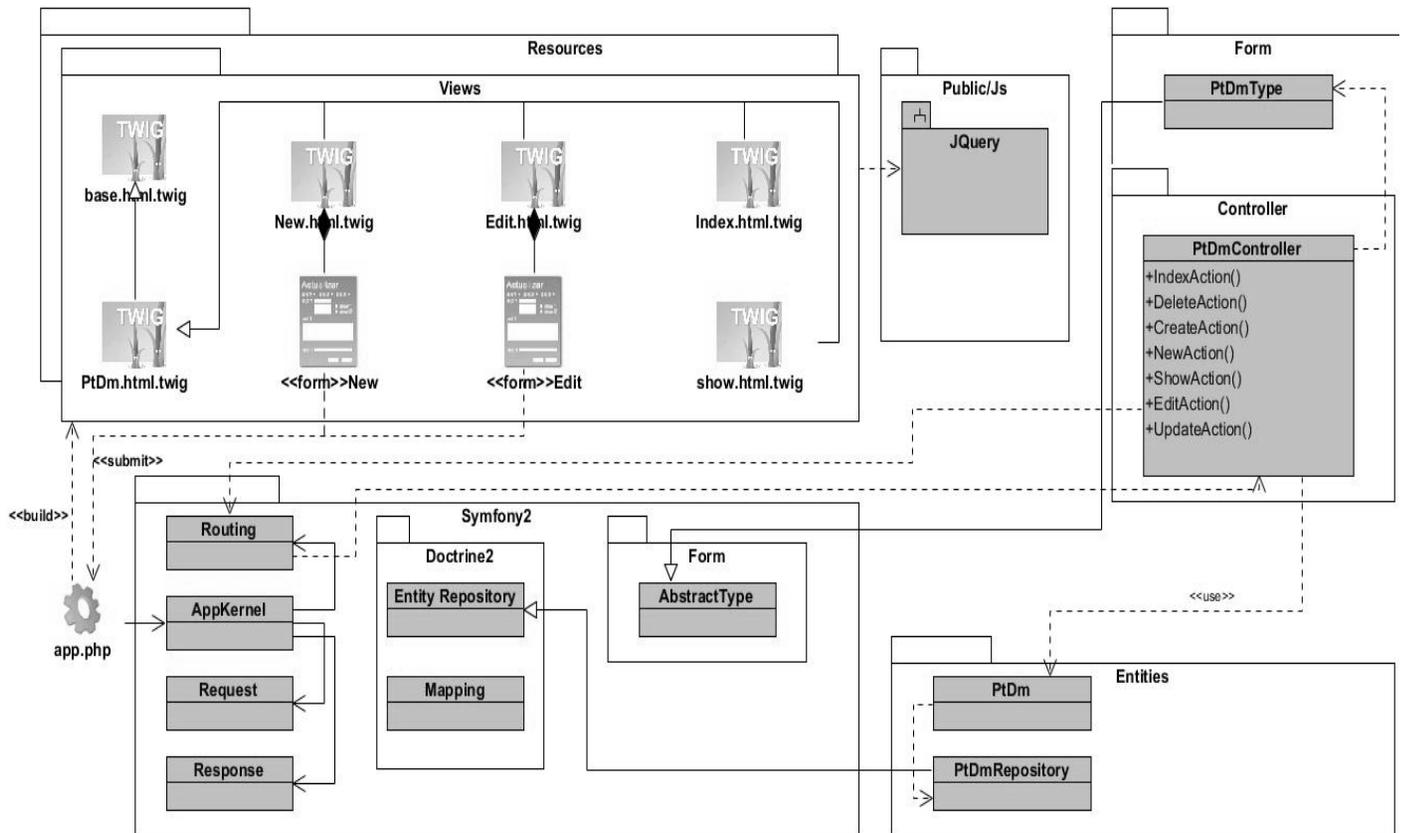


Fig. 5 Diagrama de clases del diseño. Caso de Uso: Gestionar PTDM

### 2.6 Diagrama Entidad-Relación

El diagrama Entidad-Relación (DER) tiene como objetivo representar mediante una notación gráfica los objetos de datos y sus relaciones. Este define todos los datos que se introducen, se almacenan, se transforman y se producen dentro de una aplicación (Pressman, 2002). A continuación en la fig. 6 diagrama Entidad-Relación del Sistema de Gestión de la Información del Trabajo Docente Metodológico se muestra la estructura física de la Base de Datos del sistema:

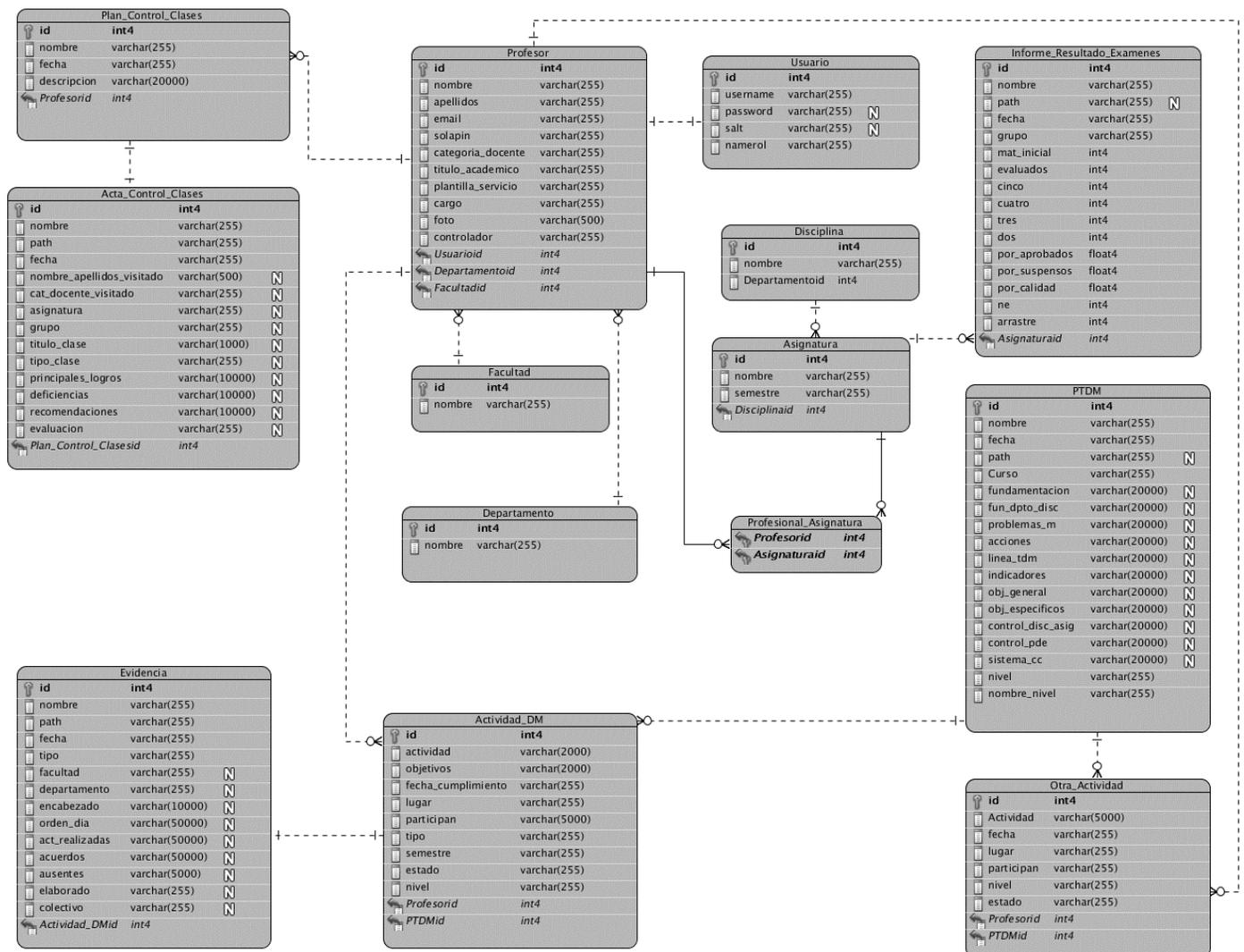


Fig. 6 Diagrama Entidad-Relación del Sistema de Gestión de la Información del Trabajo Docente Metodológico

## 2.7 Conclusiones parciales

Con el modelado del negocio, la obtención de los Requisitos Funcionales y No Funcionales y el modelado del sistema, se propicia el entendimiento entre el equipo de desarrollo y el cliente en función de lo que el sistema debe realizar y las características que debe poseer. Como resultado del presente capítulo se obtuvieron además los artefactos correspondientes al flujo de trabajo de diseño, el cual sirve como material de referencia para futuras ampliaciones y modificaciones del sistema y como guía certera para comenzar su implementación.

### Capítulo 3: Implementación y Pruebas del Sistema de gestión de la información generada por el trabajo docente metodológico

En este capítulo se detallan las fases finales de la construcción del sistema, abordando artefactos fundamentales dentro de la implementación, con el objetivo de obtener un producto que cumpla con los requisitos y que su aplicación sea de manera satisfactoria. Se hace necesaria la comprobación para asegurar que satisface todas las expectativas y cumple con las funcionalidades esperadas, con tal objetivo se introducen, un grupo de ideas fundamentales sobre el proceso de pruebas de la propuesta de solución, abordando las diferentes tipologías de las mismas, centradas en su aplicación sobre sistemas Web.

#### 3.1 Modelo de Implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. También muestra cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros. El modelo brinda una mejor visualización del camino a seguir por los desarrolladores del sistema en el momento de enfrentar la implementación del mismo (Jacobson, y otros, 2007).

##### 3.1.1 Representación de dependencias entre componentes de software

Un componente de software es una parte física de un sistema, pudiendo considerarse la personificación en software de una clase. La representación de dependencias entre componentes de software puede considerarse como representación tácita del modelo de implementación, en ella se pueden incluir componentes de código fuente, componentes del código binario, y componentes ejecutables (Jacobson, y otros, 2007).

La fig. 7 Diagrama de paquetes Caso de Uso Gestionar PTDM ilustra un diagrama de paquetes, el mismo constituye una representación de las principales dependencias entre los elementos fundamentales que conforman la estructura del sistema. Los paquetes “jQuery” y “CSS” agrupan los archivos de extensión “js” y “css” respectivamente, responsables de enriquecer las vistas con las que deben interactuar los usuarios de la aplicación. Estas vistas están constituidas por archivos de extensión “twig” agrupados en el paquete “View”. Los paquetes “Controller” y “Model” contienen los componentes encargados de la lógica de control y negocio respectivamente dentro del sistema. El paquete denominado “Núcleo de Symfony2”

representa los componentes provistos por el framework con el fin de posibilitar y facilitar la comunicación y el funcionamiento de los componentes propiamente construidos y el sistema de forma general.

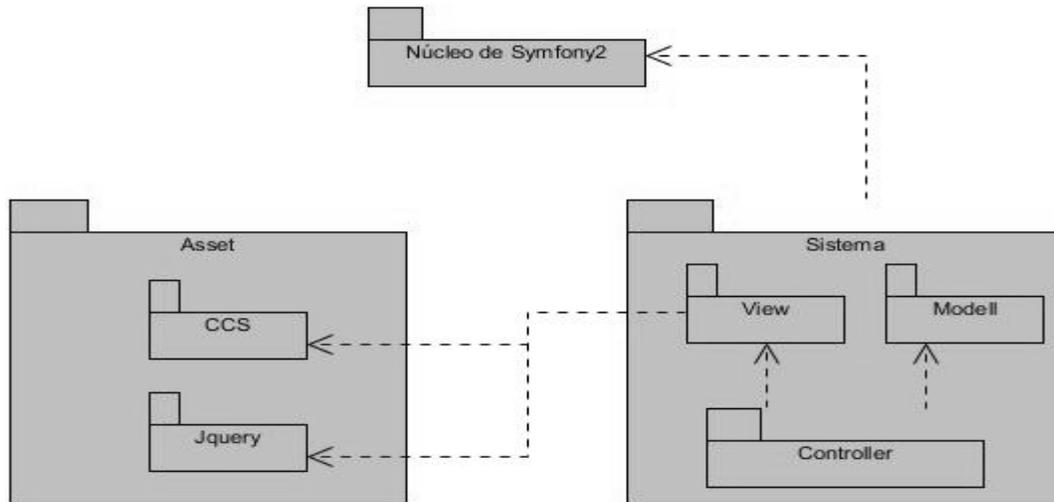


Fig. 7 Diagrama de paquetes. Caso de Uso: Gestionar PTDM

A continuación se representa en la fig. 8 Diagrama de componentes Caso de Uso Gestionar PTDM, incluye los componentes fundamentales y sus dependencias.

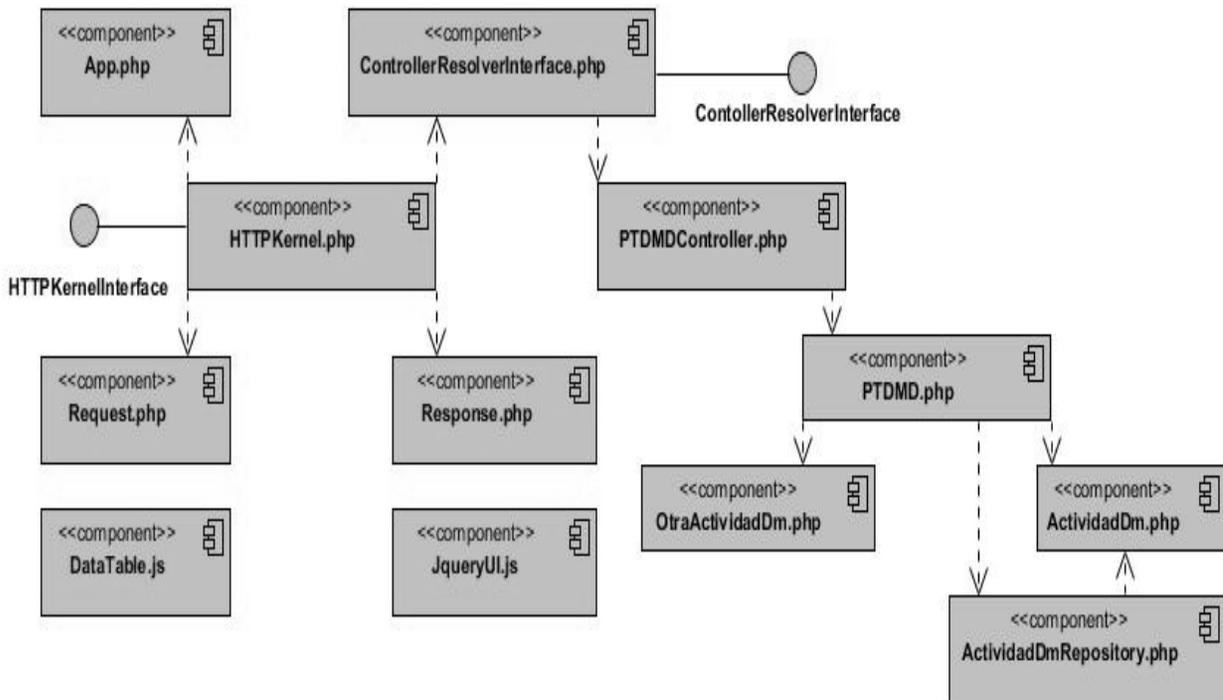


Fig. 8 Diagrama de componentes. Caso de Uso: Gestionar PTDM

## 3.2 Modelo de Despliegue

El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos que son representados (Pressman, 2002). Se presenta a continuación la fig. 9 Diagrama de despliegue del Sistema para la Gestión de la Información del Trabajo Docente Metodológico.

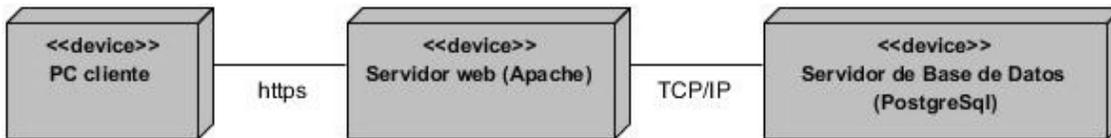


Fig. 9 Diagrama de despliegue del Sistema para la Gestión de la Información del Trabajo Docente Metodológico

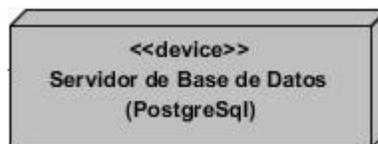
### Descripción de sus nodos.



**Pc Cliente:** Nodo que representa las PC que pueden ser utilizadas por los usuarios para acceder a la aplicación.



**Servidor Web (Apache):** Nodo que representa el servidor donde está alojada la aplicación destinada a la automatización de la información generada por el trabajo docente metodológico en los departamentos docentes de la facultad 6.



**Servidor de Base de Datos (PostgreSQL):** Nodo que representa el servidor donde se encuentra la Base de Datos que contiene los datos persistentes de la aplicación.

### 3.3 Estándares y estilos de codificación

Un estilo de programación no es más que la forma en que se da formato al código fuente (Calcifer, 2012). A diferencia de la sintaxis de un programa que son reglas estáticas que obligatoriamente hay que seguir, un estilo de programación está constituido por directrices que ayudan a obtener programas más legibles. Si bien no existen estilos de programación correctos o incorrectos es aconsejable la adopción de un conjunto de normas para la escritura de programas. Estas normas, básicamente, se refieren a la forma en que se colocan las llaves, a como se indenta el código y como se ubican los paréntesis (Viñolo, y otros, 2012).

Es muy importante dentro de un mismo proyecto mantener siempre las mismas reglas o el mismo estilo. Incluso, cuando se trabaja sobre un proyecto escrito por otro, es mejor adaptarse al estilo en que está escrito en vez de mezclarlos. Un estándar de codificación comprende todos los aspectos de la generación de código. La utilización de un estándar permite reflejar un estilo armonioso en el código, como si un único programador lo hubiera escrito todo de una sola vez. Es necesario que este se establezca al comenzar un proyecto de software para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada (Viñolo, y otros, 2012).

Con el fin de garantizar las características descritas en este acápite y con ello potenciar las ventajas que ofrece la adopción de estándares y estilos de codificación se emplearon las convenciones seguidas por Symfony2 de los estándares PSR-0, PSR-1 y PSR-2 del lenguaje PHP (Potencier, 2016).

Algunas características del estilo utilizado basado en los estándares PSR-0, PSR-1 y PSR-2 de PHP:

- El nombre de las clases se realiza en **UpperCamelCase**, es decir, que comienza por mayúscula.
- Se usa notación **camelCase** sin guiones bajos en variables, funciones, métodos y argumentos.
- Se usa el guion bajo para los nombres de opciones y nombres de parámetros.
- Todas las clases llevan **namespace**.
- Las clases abstractas llevan el prefijo **Abstract**.
- Las interfaces llevan el sufijo **Interface**.
- Los traits (PHP >= 5.4) llevan el sufijo **Trait**.
- Las excepciones tienen el sufijo **Exception**.
- Los nombres de fichero usan caracteres alfanuméricos y guiones bajos.
- Se añade un solo espacio después de cada delimitador coma.
- Se añade un solo espacio alrededor de los operadores (==, &&, ...).
- Se añade una coma después de cada elemento del arreglo en un arreglo multilínea, incluso

después del último.

- Se añade una línea en blanco antes de las declaraciones **return**, a menos que el valor devuelto solo sea dentro de un grupo de declaraciones (tal como una declaración **if**).
- Se usan llaves para indicar la estructura del cuerpo de control, independientemente del número de declaraciones que contenga.
- Se declaran las propiedades de la clase antes que los métodos.

A continuación, mediante la fig. 10 se muestra un fragmento de código de la clase PtdmController donde se evidencia la utilización de las convenciones propuestas por el framework de desarrollo empleado.

```
<?php

namespace uci\sgptmBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;
use uci\sgptmBundle\Entity\PtDmD;
use uci\sgptmBundle\Form\PtDmDType;
use Symfony\Component\HttpFoundation\Response;

/**
 * PtdmController.
 *
 */
class PtdmController extends Controller {

    public function planAction($id) {

        $em = $this->getDoctrine()->getManager();
        $controles = $em->getRepository('sgptmBundle:PlanControlClasses')->findBy(array('ptDmDid' => $id));
        $entity = $em->getRepository('sgptmBundle:PtDmD')->find($id);
        return $this->render('sgptmBundle:PtDmD:show_plan.html.twig', array(
            'controles' => $controles,
            'entity' => $entity,
        ));
    }
}
```

Fig. 10 Fragmento de código de la clase PtdmController

### 3.4 El proceso de pruebas

Los procesos de prueba para las aplicaciones web comienzan con pruebas que ejercitan el contenido y la funcionalidad de la interfaz que es inmediatamente visible para los usuarios finales. Conforme se realizan las pruebas, se ejercitan los aspectos de la arquitectura de diseño y la navegación. Finalmente la atención se centra en las pruebas que ejercitan las capacidades tecnológicas referentes a la infraestructura de la aplicación y cuestiones de instalación e implementación (Pressman, 2002).

Según (Pressman, 2002) se establecen una serie de reglas que pudieran entenderse como los objetivos fundamentales del proceso de pruebas:

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

#### 3.4.1 Tipos de pruebas

Según lo planteado por (Pressman, 2002) existen siete tipos fundamentales de pruebas recomendadas sobre aplicaciones web:

- La prueba de contenido, intenta descubrir errores en el contenido del sistema.
- La prueba de interfaz, ejercita los mecanismos de interacción y valida los aspectos estéticos de la interfaz del usuario.
- La prueba de navegación, aplica Casos de Uso, derivados del análisis, en el diseño de casos de prueba que ejerciten cada escenario de uso contra el diseño de la navegación.
- La prueba de componentes, ejercita el contenido y las unidades funcionales dentro de la aplicación.
- Las pruebas de configuración, intentan descubrir los errores que son específicos respecto de un cliente o ambiente de servidor particulares.
- La prueba de seguridad, incorpora una serie de pruebas para explotar las vulnerabilidades de la aplicación y su ambiente.
- La prueba de desempeño, abarca una serie de pruebas para valorar como afecta el aumento de tráfico de usuarios la respuesta en tiempo y confiabilidad de la web.

Con el objetivo de llevar a cabo la evaluación de la propuesta de solución actual, teniendo en cuenta las características y los propósitos del sistema, se seleccionó la aplicación de dos tipos de prueba correspondientes a: las pruebas de componentes y pruebas de desempeño:

**Pruebas de componentes**, también llamadas pruebas de función, estas se enfocan sobre un conjunto de pruebas que intentan descubrir errores en la aplicación. Cada función es un módulo de software el cual se determina probar empleando el método de prueba de Caja Negra. Según (Pressman, 2008), las pruebas de Caja Negra se centran en los RF del software. Estas se emplean cuando se conoce la función específica para la que se diseñó el producto; se aplican a la interfaz del software intentando demostrar que cada función es plenamente operacional, mientras se buscan los errores de cada función.

El método mencionado anteriormente utiliza además técnicas para su realización, en la presente investigación se determina el uso de la técnica de Partición Equivalente, la que divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba (Pressman, 2005). Según (Jacobson, y otros, 2007), un Caso de Prueba representa una forma de probar el sistema, incluyendo la entrada o resultado con la que se ha de probar y las condiciones bajo las que ha de probarse.

**Pruebas de desempeño**, estas pruebas se aplican para descubrir problemas de desempeño que se presentan debido a falta de recursos en el lado del servidor, ancho de banda de red inapropiado, capacidades del sistema operativo, funcionalidades mal diseñadas y otros conflictos. Dichas pruebas se diseñan con el fin de simular situaciones de carga del mundo real, conforme crece el número de usuarios simultáneos de la aplicación, o aumenta el número de transacciones en línea, o se incrementa la cantidad de datos, las pruebas de desempeño ayudaran a elaborar una idea del nivel de aceptabilidad con el que responde el sistema a estos factores.

### **Se realizan dos pruebas de desempeño diferentes:**

**Pruebas de carga:** Son utilizadas para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

**Pruebas de tensión:** Están enfocadas a evaluar cómo el sistema responde bajo condiciones anormales (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible). Este tipo de prueba se realiza para determinar la solidez de un sistema en los momentos de carga extrema y ayuda a los administradores para determinar el rendimiento en caso de que la carga real supere a la carga esperada (Pressman, 2002).

### 3.4.2 Diseño de casos de prueba

Según (Peña, 2006) “el conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular como, por ejemplo, ejercitar un camino concreto de un programa o verificar el cumplimiento de un determinado requisito” es a lo que podríamos llamar o determinar cómo caso de prueba.

Seguidamente las fig. 7, 8, 9, 10, 11, 12 se muestra todo lo referente al diseño del caso de prueba (en lo adelante DCP) correspondiente al Caso de Uso: Gestionar PTDM, acorde al método y técnica de evaluación descritos en el epígrafe anterior.

#### Diseño de Casos de Prueba del Caso de Uso: Gestionar PTDM.

##### Descripción general

Este Caso de Uso se inicia cuando el usuario selecciona la opción de crear un nuevo plan de trabajo docente metodológico y culmina con la realización de una de las siguientes operaciones sobre un plan: *Crear, Editar, Eliminar o Mostrar*.

##### Condiciones de ejecución

El actor debe estar autenticado en el sistema y debe tener permisos para realizar las mencionadas operaciones. Debe seleccionarse el PTDM sobre la cual se pretende realizar una de las siguientes acciones: *Modificar, Eliminar o Listar*. Para realizar una de estas acciones debe existir en el sistema al menos un PTDM.

#### Secciones a probar en el Caso de Uso: Gestionar PTDM.

Tabla. 7 Secciones a probar en el Caso de Uso del Sistema: Gestionar PTDM

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
SC 1: “Crear PTDM”.	EC 1.1: El usuario inserta y selecciona datos correctos en los campos de entrada.	El sistema almacena un nuevo PTDM. Muestra un mensaje confirmando el éxito de la operación.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Crear PTDM”.</li> <li>▪ Inserta datos en los campos.</li> <li>▪ Clic en el botón “Crear”.</li> </ul>
	EC 1.2: El usuario deja campos vacíos.	El sistema muestra un mensaje indicando que se deben especificar todos los campos para crear un PTDM.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Crear PTDM”.</li> <li>▪ Inserta datos en los campos.</li> <li>▪ Clic en el botón “Crear”.</li> </ul>
	EC 1.3: El usuario	El sistema muestra un	<ul style="list-style-type: none"> <li>▪ Clic en “PTDM”.</li> </ul>

	introduce datos incorrectos.	mensaje indicando que existen datos incorrectos.	<ul style="list-style-type: none"> <li>▪ Clic en “Crear PTDM”.</li> <li>▪ Inserta datos en los campos.</li> <li>▪ Clic en el botón “Crear”.</li> </ul>
<b>SC 2:</b> “Editar PTDM”.	<b>EC 2.1:</b> El usuario modifica correctamente los datos en los campos.	El sistema modifica el PTDM. Muestra un mensaje indicando el éxito de la operación.	<ul style="list-style-type: none"> <li>▪ Clic en “PTDM”.</li> <li>▪ Clic en “Listar PTDM”.</li> <li>▪ Clic en el icono editar del PTDM correspondiente.</li> <li>▪ Insertar los datos a editar.</li> <li>▪ Clic en el botón “Editar”.</li> </ul>
	<b>EC 2.2:</b> El usuario deja campos vacíos.	El sistema muestra un mensaje indicando que todos los campos son requeridos.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Listar PTDM”.</li> <li>▪ Clic en el icono editar del PTDM correspondiente.</li> <li>▪ Insertar datos a editar.</li> <li>▪ Clic en el botón “Editar”.</li> </ul>
	<b>EC 2.3:</b> El usuario introduce datos incorrectos.	El sistema muestra un mensaje indicando que existen datos incorrectos.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Listar PTDM”.</li> <li>▪ Clic en el icono editar del PTDM correspondiente.</li> <li>▪ Insertar datos a editar.</li> <li>▪ Clic en el botón “Editar”.</li> </ul>
<b>SC 3:</b> “Mostrar PTDM”.	<b>EC 3.1:</b> Mostrar PTDM.	El sistema muestra los datos de un PTDM dado.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Listar PTDM”.</li> <li>▪ Clic en el icono Mostrar para ver los detalles del PTDM correspondiente.</li> </ul>
<b>SC 4:</b> “Eliminar PTDM”.	<b>EC 4.1:</b> Eliminar un PTDM.	El sistema elimina el plan de trabajo docente metodológico.	<ul style="list-style-type: none"> <li>▪ Clic en “Gestionar PTDM”.</li> <li>▪ Clic en “Listar PTDM”.</li> <li>▪ Clic en el icono eliminar del PTDM correspondiente.</li> </ul>

### Descripción de las Variables

Tabla. 8 Descripción de las Variables

No.	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	nombre	campo de texto	No	Debe ser un título de documento válido, solo caracteres alfanuméricos.
2	fecha	campo de selección	No	Debe ser una fecha válida, no anterior a la actual, bajo el formato dd/mm/YY.
3	curso	campo de selección	No	Debe coincidir con el formato de un curso en el conjunto de los dos años que este comprende: YY/YY.
4	fundamentación	campo de texto	Si	Debe ser un texto que en caso de que exista valor para el campo <b>archivo</b> puede ser nulo.

5	problemas metodológicos	campo de texto	Si	Debe ser un texto que en caso de que exista valor para el campo <b>archivo</b> puede ser nulo.
6	indicadores	campo de texto	Si	Debe ser un texto que en caso de que exista valor para el campo <b>archivo</b> puede ser nulo.
7	objetivo general	campo de texto	Si	Debe ser un texto que en caso de que exista valor para el campo <b>archivo</b> puede ser nulo.
8	objetivos específicos	campo de texto	Si	Debe ser un texto que en caso de que exista valor para el campo <b>archivo</b> puede ser nulo.
9	archivo	campo de archivo	Si	Debe ser un archivo en formato documento.

### Matriz de Datos

Tabla. 9 Caso de Uso: Crear plan de Trabajo Docente Metodológico

Escenario	Nombre	fecha	curso	fundamentación	problemas metodológicos	indicadores	objetivo general	objetivos específicos	archivo	Respuesta del sistema
El usuario inserta y selecciona datos correctos en los campos de entrada.	plan de trabajo docente metodológico	14-05-2016	2015-2016	Para planificar las actividades a desarrollar...	Baja calidad en las actividades ...	El cumplimiento del Plan ...	Elevar la calidad de la preparación ...	Eliminar las principales insuficiencias ...	Documento (*.pdf)	El sistema almacena un nuevo PTDM, muestra un mensaje indicando el éxito de la operación.

El usuario deja campos vacíos.	plan de trabajo docente metodológico	14-05-2016	2015-2016	Para planificar las actividades a desarrollar...	vacío	vacío	vacío	vacío	vacío	El sistema muestra un mensaje indicando que se deben especificar todos los campos para crear un PTDM
El usuario introduce datos incorrectos.	plan de trabajo docente metodológico	14444	2015-tttt	Para planificar las actividades a desarrollar...	Baja calidad en las actividades ...	El cumplimiento del Plan ...	Elevar la calidad de la preparación ...	Eliminar las principales insuficiencias ...	Archivo (*.mkv)	El sistema muestra un mensaje indicando que existen datos incorrectos.

Tabla. 10 Caso de Uso: Modificar plan de Trabajo Docente Metodológico

Escenario	Nombre	fecha	curso	fundamentación	problemas metodológicos	indicadores	objetivo general	objetivos específicos	archivo	Respuesta del sistema
El usuario modifica correctamente los datos en los campos	plan de trabajo docente metodológico	14-05-2016	2015-2016	Para planificar las actividades a desarrollar...	Baja calidad en las actividades ...	El cumplimiento del Plan ...	Elevar la calidad de la preparación ...	Eliminar las principales insuficiencias ...	Documento (*.pdf)	El sistema modifica el PTDM. Muestra un mensaje indicando el éxito de la operación.
El usuario deja campos vacíos.	plan de trabajo docente metodológico	14-05-2016	2015-2016	Para planificar las actividades a desarrollar...	vacío	vacío	vacío	vacío	vacío	El sistema muestra un mensaje indicando que se deben especificar todos los campos para crear un PTDM.

El usuario introduce datos incorrectos.	plan de trabajo docente metodológico	14444	2015-tttt	Para planificar las actividades a desarrollar...	Baja calidad en las actividades ...	El cumplimiento del Plan ...	Elevar la calidad de la preparación ...	Eliminar las principales insuficiencias ...	Archivo (*.mkv)	El sistema muestra un mensaje indicando que existen datos incorrectos.
---	--------------------------------------	-------	-----------	--	-------------------------------------	------------------------------	---	---	-----------------	--

Tabla. 11 Caso de uso: Eliminar plan de Trabajo Docente Metodológico

Escenario	Nombre	fecha	curso	fundamentación	problemas metodológicos	indicadores	Objetivo general	Objetivos específicos	Archivo	Respuesta del sistema
Eliminar plan de trabajo.	NA	NA	NA	NA	NA		NA	NA	NA	El sistema elimina un PTDM seleccionado, muestra un mensaje confirmando el éxito de la operación.

Tabla. 12 Caso de Uso: Listar plan de Trabajo Docente Metodológico

Escenario	Nombre	fecha	curso	fundamentación	problemas metodológicos	Objetivo general	Objetivos específicos	archivo	Respuesta del sistema
El usuario modifica correctamente los datos en los campos	NA	NA	NA	NA	NA	NA	NA	NA	El sistema muestra un listado con los PTDM existentes.

### 3.5 Ejecución de las pruebas

Como parte de la ejecución de las pruebas de Caja Negra, con el objetivo de identificar en qué medida satisface la aplicación las funcionalidades implementadas se realizaron tres iteraciones de pruebas como muestra en la fig. 11, las mismas fueron corregidas progresivamente hasta obtenerse un sistema que garantizara el cumplimiento de todos los requisitos definidos, durante esta ejecución fueron aplicados los diseños de casos de pruebas realizados a partir de las descripciones textuales de los Casos de Uso del Sistema.

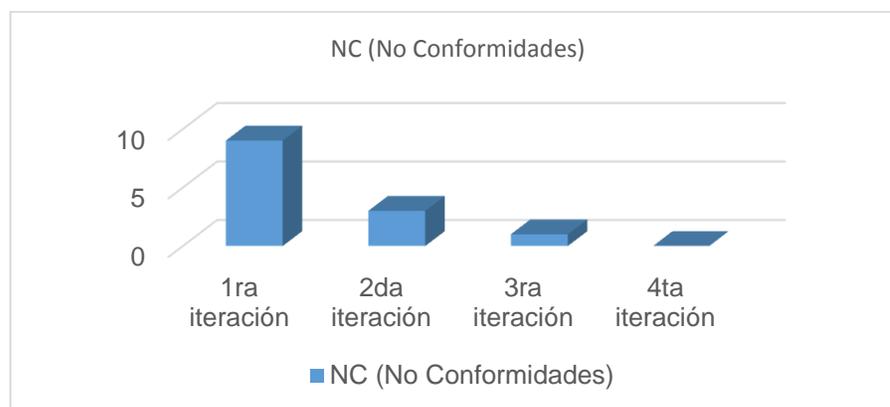


Fig. 11 Ejecución de las pruebas de Caja Negra. Resultados de las iteraciones realizadas

Al realizarse la ejecución de las pruebas de carga, se obtuvieron los resultados que a continuación se reflejan en la tabla 13, los mismos obtenidos a partir del uso de la herramienta Apache JMeter 2.3.4.

Tabla. 13 Resultados de las pruebas de carga sobre la propuesta de solución

No.	Cantidad de usuarios concurrentes	Tiempo medio de respuesta en segundos	Errores detectados
1	20	1.3	0
2	35	2.5	0
3	50	4	0

Resulta relevante destacar que el tiempo de respuesta del sistema para una cantidad de usuarios concurrentes semejante a lo que debe constituir su ambiente común, no sobrepasa los 5 segundos conforme a lo definido en el Requisito No Funcional de eficiencia.

Luego de la ejecución de las pruebas de tensión, se obtuvieron los resultados que a continuación se reflejan en la tabla 14, estos fueron arrojados a partir del uso de la herramienta Apache JMeter 2.3.4 quedando en evidencia el cumplimiento del requisito no funcional de eficiencia en el caso de condiciones de estrés del sistema.

Tabla. 14 Resultados de las pruebas de tensión sobre la propuesta de solución

No.	Cantidad de usuarios concurrentes	Tiempo medio de respuesta en segundos	Errores detectados
1	200	8.5	0
2	350	10	0
3	450	12.5	0
4	500	14	0

### 3.6 Aportes a considerar

La obtención del sistema propuesto como solución a la problemática de la presente investigación, va encaminada a contar con una serie de ventajas sobre el trabajo metodológico en la facultad de los profesionales de la educación superior.

La solución permite el manejo del claustro de acuerdo a las estructuras definidas en la facultad, la cual es flexible y escalable, permitiendo obtener estadísticas sobre los profesionales.

La creación de un Plan de Trabajo Docente Metodológico se puede realizar a partir del sistema, ofreciendo un modelo estandarizado para ello, con características avanzadas para la edición del texto y la incorporación de un grupo de datos de forma automática a partir del usuario que realiza la acción. Se permite adjuntar al plan un documento ya elaborado en formato digital, el cual puede representar el propio plan, buscando obtener estadísticas retrospectivas sin necesidad de transcribir la información hacia el sistema. La creación de las actividades, tanto metodológicas como no, ocurre a manera de proceso de asignación, asociándose al PTDM correspondiente y notificándole mediante correo electrónico a su responsable la generación de la misma.

Las evidencias asociadas a las diversas ADM pueden ser generadas a partir del sistema, ofreciendo una propuesta de modelo estandarizado que aporta claridad y consistencia en cuanto a la información que se recoge. De igual manera un grupo de datos se generan de manera automática facilitando el proceso y aportándole agilidad. Las ADM sobre el sistema, vinculadas a sus evidencias son evaluadas por los responsables pertinentes y se determina su cumplimiento, permitiendo la obtención de estadísticas mediante informes y gráficas que aportan datos relevantes para la toma de decisiones. Las actividades incumplidas o pendientes son notificadas a sus responsables de forma automática.

Se permite la generación a partir de la solución de otros documentos relacionados con el trabajo metodológico, como el plan de controles a clase y su evidencia respectiva, cuidando aspectos como las categorías docentes de los profesionales controlados y a controlar conforme a las reglas del negocio. Así mismo los informes de resultado de examen pueden ser creados sobre el sistema, ofreciendo en todos los casos un estándar en la documentación. Todos los documentos generados a partir de la aplicación pueden obtenerse además en formato pdf.

El acceso al sistema se puede realizar desde cualquier punto de la red de la UCI y en cualquier horario, mediante las credenciales del dominio que provee la universidad para sus servicios telemáticos.

### **3.7 Conclusiones parciales**

La obtención del modelo de componentes aporta claridad ante el proceso de implementación del sistema, así mismo el modelo de despliegue permite conocer el ambiente en el que debe desempeñarse la presente propuesta de solución. Mediante el proceso de pruebas se pudo identificar y resolver los errores, garantizando un producto de alta calidad el cual cumple los requisitos planteados, tanto funcionales como no funcionales.

### Conclusiones generales

Una vez culminada la investigación se puede afirmar que se les dio cumplimiento a los objetivos planteados, arribando a las siguientes conclusiones:

- ✓ El estudio de los principales conceptos relacionados con la gestión de la información permitió sentar las bases para el desarrollo del Sistema de Gestión de la Información del Trabajo Docente Metodológico.
- ✓ A partir de la realización del análisis y diseño del Sistema de Gestión de la Información del Trabajo Docente Metodológico se obtuvo como resultado los diagramas y artefactos necesarios para guiar el desarrollo del mismo.
- ✓ La implementación del Sistema de Gestión de la Información del Trabajo Docente Metodológico dio cumplimiento a los 51 requisitos funcionales identificados en las fases de análisis y diseño.
- ✓ El diseño y ejecución de las pruebas a nivel de componentes y de desempeño permitió comprobar el correcto funcionamiento del Sistema de Gestión de la Información del Trabajo Docente Metodológico y el cumplimiento de las funcionalidades incorporadas.
- ✓ Como resultado se obtuvo el Sistema de Gestión de la Información del Trabajo Docente Metodológico que permite gestionar toda la documentación referente al Trabajo Docente Metodológico en los departamentos docente de la facultad 6 de la UCI.

1. Elaborar un manual de usuario que sirva de guía para la explotación óptima del sistema.

### Referencias Bibliográficas

- Arévalo. (2007). *Gestión de la información, gestión del contenido y del conocimiento*.
- Bakken, S. S., & A. AULBACH, e. a. (2005). *Manual de PHP. MARTINEZ, R.*
- Balduino, R. (2007). *Introduction to OpenUP (Open Unified Process)*.
- Belmonte, & Oscar, G. C. (2010). *Desarrollo de proyectos informáticos con tecnología Java*. Obtenido de <http://www3.uji.es/~belfern/pdf/libroJavaConTapa.pdf>.
- Calcifer. (2012). *Guía de programación de GNOME*. Obtenido de <http://calcifer.org/documentos/programming-guidelines/code-style.html>.
- Castillo, & González, L. J. (2010). *La observación a clases: Vía fundamental para la superación pedagógica de los profesores*. Cuba.
- CENDITEL. (2011). *Estilos de programación*. Obtenido de [fsl.cenditel.gob.ve/docman/view.php/7/73/estilos.odt](http://fsl.cenditel.gob.ve/docman/view.php/7/73/estilos.odt).
- Comunity, O. U. (2011).
- Flanagan, D. (2002). *JavaScript: The Definitive Guide*.
- Fornaris, Y. D., & D. R. ROSALES, e. a. (2012). *Herramienta para la transmisión de audiovisuales basada en Listas de Edición para el Departamento de Señales Digitales*. La Habana, Universidad de las Ciencias Informáticas. Señales Digitales.
- Foundation. (2010a). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, A. S. (2010 ). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, A. S. (2010 a). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, A. S. (2010). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, A. S. (2010a). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, A. S. (2010a). *Overview of new features in Apache 2.2*. Obtenido de [http://httpd.apache.org/docs/2.2/new\\_features\\_2\\_2.html](http://httpd.apache.org/docs/2.2/new_features_2_2.html)
- Foundation, J. (2010b). *What is jQuery?* Obtenido de <http://www.jquery.com>
- Garlan, D., & M. SHAW, e. a. (1996). *An Introduction to Software Architecture*.
- Group, P. G. (2011). *Nota Original de Lanzamiento, Equipo Global de Desarrollo de PostgreSQL*. Obtenido de <http://www.postgresql.org/about/press/presskit91/es/>

- Guerrero, M. R. (2009). *Sobre PostgreSQL*. Obtenido de [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Jacobson, I., & G. BOOCH, e. a. (2007). *Rational Unified Proces*.
- Kioskea. (2011). *Lenguaje de programación*. Obtenido de <http://es.kioskea.net/contents/304-lenguajes-de-programacion>
- Larman, C. (1999). *UML y patrones introduccion al analisis y diseño orientado a objetos*.
- Larman, C. (2005). *UML y patrones introduccion al analisis y diseño orientado a objetos* (2da Edición ed.). California.
- León, I. B. (2015).
- Letelier, & P. a. (2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
- Maso, I. H., & A. P. (2008). *Análisis y diseño del Módulo profesor del Sistema de Gestión Académica*. La Habana.
- MINED. (2010). *Reglamento del trabajo metodológico del ministerio de educación*.
- Parra. (2011). *portal Ubuntu*. Obtenido de <http://www.portalubuntu.com/2011/04/instalar-netbeans-70-en-español-en.html>
- Peña, J. M. (2006). *Pruebas de Software*. Obtenido de <http://www.uv.mx/personal>
- Pérez, J. E. (2006). *Así serán las novedades de Symfony 2.0*. Obtenido de <http://www.symfony.es/2009/03/06/asi-serán-las-novedades-de-symfony-20/>
- Pérez, M. D., & Yimian de Lis Contreras, e. a. (2009). *Características de los sistemas de información que permiten la gestión oportuna de la información y el conocimiento institucional* (Vol. Vol 20).
- Pérez, M. Z. (1999). La Habana. Obtenido de <http://instituciones.sld.cu/ucmh/>
- Pichardo, R. R., & Roberto Verrier Rodríguez, e. a. (2010). *Algunas actividades presenciales y metodologicas del profesor universitario en Cuba*.
- Pontificia universidad javeriana de bogotá. (16 de 02 de 2016). Obtenido de <http://ingenieria.javeriana.edu.co/inicio>
- PostgreSQL. (2012). *pgadmin*. Obtenido de <http://www.pgadmin.org>
- Potencier, F. (2016). *The Symfony book* (Vol. 1). Paris. Obtenido de [http://symfony.com/pdf/Symfony\\_book\\_3.1.pdf?v=4](http://symfony.com/pdf/Symfony_book_3.1.pdf?v=4)
- Pressman, R. S. (2002). *Ingeniería del Software* (5ta ed.). La Habana.
- Pressman, R. S. (2005). *Ingeniería del Software. Un enfoque práctico*.
- Pressman, R. S. (2008). *Ingeniería del Software*.
- Romero, H. (2009). *Metodologías de Desarrollo*.
- Ruesta, C. B. (2001). *Gestion del conocimiento y gestion de la informacion*.
- Ruiz, B. J., & S. P. Pérez, e. a. (2012). *Gestión de Desarrollo de Proyecto de software*.
- Ruiz, R. V. (2013). *La Gestión en la producción*.

- Rumbaugh, J. (2000). *El Lenguaje de Modelado. Manual de Referencia*.
- SCHMULLER, J. (2010). *Aprendiendo UML en 24*.
- Schmuller, J. (2010). *Aprendiendo UML en 24 horas*.
- SIG, I. (2015).
- Sommerville, I. (2005). *Ingeniería del Software*.
- Sulca, J. A. (2012). *El Proceso Unificado de Desarrollo de Software (RUP)*. Perú : Universidad Cesar Vallejo.
- Team, S. T. (2011). *Características de PHP*.
- Torres, E. O., & SÁNCHEZ, a. M. (2010). *La clase metodologica Instructiva en La Educacion Superior Cubana*.
- UCOL. (2006). *Las herramientas CASE. [En línea]* . Universidad de Colima. Obtenido de [http://docente.ucol.mx/al961223/public\\_html/centro6.htm](http://docente.ucol.mx/al961223/public_html/centro6.htm).
- UNICEN . (10 de 12 de 2015). Obtenido de Universidad nacional del centro de la provincia de Buenos Aires: <http://www.unicen.edu.ar/>
- Valdés, J. V. (2007). *Reglamento el trabajo metodológico del ministerio de educación Superior*.
- Viñolo, & Raydel Raúl y Roquero Figueroa, A. (2012). *Sistema Gestor de Procesos de Medias v2*. La Habana : UCI.
- Zamora, A. M. (2011). *LA Clase Metodologica en el proceso de Universalizacion de la Educacion Superior* (Vol. Vol 3).
- Zaninneto, F. a. (2008). *Symfony. La Guía Definitiva*.