

Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Ingeniería

Carrera de Ingeniería Informática



“Sistema Informático para la Gestión del Trabajo Metodológico en la Universidad de Cienfuegos”.

Trabajo de diploma para optar por el título de Ingeniero Informático

Autora:

María Caridad Díaz Rojas

Tutor:

Dr. Ernesto Roberto Fuentes Garí

Cienfuegos, Cuba

Curso 2012 – 2013

Declaración de autoría

Declaro que soy el único autor de este trabajo de diploma titulado “Sistema Informático para la Gestión del Trabajo Metodológico en la Universidad de Cienfuegos”. y por este medio reconozco al Departamento de Informática de la Facultad de Ingeniería de la Universidad de Cienfuegos “Carlos Rafael Rodríguez” con todos los derechos patrimoniales del mismo. Para que así conste firmamos la presente a los _____ días del mes de _____ del 2013.

Autor: María C. Díaz Rojas

Tutor: Dr. Ernesto Roberto Fuentes Garí

Pensamiento

"Las grandes almas tienen voluntades; las débiles tan solo deseos."

Proverbio chino

Dedicatoria

A la persona más importante de mi vida, mi Mama.

A mi familia.

A mi pareja.

Resumen

La presente investigación tiene como meta la elaboración de un sistema informático capaz de gestionar y controlar la información relacionada con los procesos propios del trabajo metodológico en la Universidad de Cienfuegos, con el fin de agilizar la gestión de la información y mejorar tales procesos.

El sistema propuesto tiene como nombre “Sistema Informático para la Gestión del Trabajo Metodológico de la Universidad de Cienfuegos”. Permite gestionar el proceso de planificación del trabajo metodológico así como el control de la ejecución de las actividades planificadas y facilita el acceso, la manipulación y la compartición de los datos.

En el documento presentado quedan descritos los elementos que conforman el análisis, diseño e implementación del sistema propuesto, siguiendo lo establecido por la metodología Desarrollo Guiado por Funcionalidades (FDD) y utilizando el Lenguaje Unificado de Modelado (UML). Para la implementación del mismo se utilizó MySQL como Sistema Gestor de Bases de Datos, Apache como servidor web, PHP como lenguaje de programación y el framework CodeIgniter.

Índice

Introducción	1
Capítulo I: Fundamentación Teórica.	6
1.1- <i>Introducción.</i>	6
1.2- <i>Conceptos relacionados al dominio del Problema.</i>	6
1.3- <i>Descripción del objeto de estudio.</i>	12
1.4- <i>Descripción de los sistemas existentes asociados al Objeto de Estudio:</i>	13
1.4.1- <i>Sistemas existentes en Cuba.</i>	13
1.4.2- <i>Presentación de la Propuesta de Solución.</i>	14
1.5- <i>Tendencias, lenguajes de programación, metodologías y tecnologías actuales.</i> 14	
1.5.1- <i>Etapas de diseño y desarrollo.</i>	14
1.5.2- <i>Lenguajes y tecnologías web.</i>	17
1.5.3- <i>Selección del Framework.</i>	21
1.5.4- <i>Recursos utilizados para el desarrollo del producto informático.</i>	24
1.5.5- <i>Sistema Gestión de Base de Datos (SGBD).</i>	25
1.6- <i>Conclusiones del capítulo.</i>	27
Capítulo II: Descripción del Sistema.	28
2.1- <i>Introducción.</i>	28
2.2- <i>Desarrollo de un modelo global.</i>	28
2.2.1- <i>Flujo actual de los procesos involucrados en el campo de acción.</i>	28
2.2.2- <i>Objeto de automatización.</i>	30
2.2.3- <i>Modelo de dominio. Diagrama de objetos.</i>	31
2.2.4- <i>Reglas a considerar.</i>	32
2.3- <i>Construcción de una lista de funcionalidades.</i>	32
2.3.1- <i>Requerimientos funcionales.</i>	33
2.3.2- <i>Requerimientos no funcionales.</i>	35
2.4- <i>Estudio de la factibilidad.</i>	37
2.4.1- <i>Beneficios tangibles e intangibles.</i>	45
2.4.2- <i>Análisis de costos y beneficios.</i>	46
2.5- <i>Planeación por funcionalidad.</i>	46
2.6- <i>Diseño por funcionalidad.</i>	49

2.6.1- Principios del diseño.....	49
2.6.2- Modelo conceptual de la base de datos.....	50
2.6.3- Modelo lógico de la base de datos.....	52
2.6.4- Modelo físico de la base de datos.....	53
2.7- Conclusiones.....	54
Capítulo III: Construcción del Sistema.....	55
3.1- Introducción.....	55
3.2- Descripción de la arquitectura propuesta.....	55
3.3- Diagrama de clases.....	56
3.4- Estándar de implementación.....	57
3.5- Seguridad del sistema.....	57
3.6- Tratamiento de errores.....	58
3.7- Diagrama de despliegue.....	58
3.8- Validación de la solución propuesta.....	59
3.8.1- Prueba T para muestras pareadas.....	59
3.9- Conclusiones.....	63
Conclusiones.....	64
Recomendaciones.....	65
Referencias Bibliográficas.....	66
Bibliografía.....	68
Anexos.....	71

Índice de Tablas

Tabla 1: Comparación entre metodologías.....	15
Tabla 2: Primer bloque de funcionalidades	33
Tabla 3 : Segundo bloque de funcionalidades	34
Tabla 4: Tercer bloque de funcionalidades.....	35
Tabla 5: Cuarto bloque de funcionalidades.	35
Tabla 6: Entradas externas.	39
Tabla 7: Salidas externas.....	40
Tabla 8: Consultas externas.	41
Tabla 9: Archivos lógicos internos.....	41
Tabla 10: Puntos de función desajustados.....	41
Tabla 11: Multiplicadores de Esfuerzo	42
Tabla 12: Factor de Escala.	43
Tabla 13: Resumen de resultados.....	45
Tabla 14: Primer bloque de funcionalidades planeado.....	47
Tabla 15: Segundo bloque de funcionalidades planeado.....	47
Tabla 16: Tercer bloque de funcionalidades planeado.....	49
Tabla 17: Cuarto bloque de funcionalidades planeado.	49
Tabla 18: Diagramas de clases	57
Tabla 19: Tiempos antes y después de la realización de los procesos.	60

Índice de Figuras

Ilustración 1: Funciones del trabajo metodológico.....	8
Ilustración 2: Niveles organizativos del trabajo metodológico.....	10
Ilustración 3: Formas del trabajo metodológico.....	12
Ilustración 4: Niveles jerárquicos del trabajo metodológico	29
Ilustración 5: Tipo de actividades metodológicas.....	30
Ilustración 6: Modelo de objetos.....	31
Ilustración 7: Modelo conceptual de la base de datos.....	51
Ilustración 8 : Modelo lógico de la base de datos.....	52
Ilustración 9: Modelo físico de la base de datos.....	53
Ilustración 12: Diagrama de despliegue del sistema.....	59
Ilustración 13: Comparación del tiempo promedio de los procesos antes y después del sistema.....	61
Ilustración 14: Prueba de Kolmogorov-Smirnov para una muestra.....	62
Ilustración 15: Resultado de la Prueba T.....	62

Introducción

El Ministerio de Educación Superior de Cuba ha puesto esencial interés y preocupación en llevar a todos los Centros Universitarios, los adelantos más actualizados de las TIC, al servicio de profesores y estudiantes, con el objetivo de lograr un egresado con mente abierta y gran creatividad, capaz de enfrentar los retos que ofrece la sociedad de estos tiempos. Además, el empleo de los diferentes productos informáticos constituye un eslabón esencial pues eleva el interés en los estudiantes, profesores y trabajadores por la investigación científica y posibilita el mejoramiento de las habilidades creativas, la imaginación, habilidades comunicativas y colaborativas pudiendo acceder a mayor cantidad de información y proporcionando los medios para un mejor desarrollo integral de los individuos. [1]

Un aspecto sumamente debatido actualmente en la educación superior es el relacionado con las vías y procedimientos a emplear para perfeccionar la labor docente en las diferentes carreras que se desarrollan en una Universidad. Es bastante generalizada la tendencia a aceptar que el éxito de esta tarea depende exclusivamente del dominio que los profesores poseen de la ciencia que enseñan, subvalorando el papel que desempeña su formación pedagógica. [2]

Por supuesto que compartimos la necesidad de ser un especialista con un dominio profundo del contenido que se enseña como premisa para que el proceso de enseñanza sea efectivo y se dirija a dar respuesta a los objetivos que en la educación superior se persiguen. Si los profesores universitarios carecen de esa elevada preparación a que hicimos referencia, entonces la esencia misma de la Universidad corre peligro, toda vez que esta institución estructura sus procesos fundamentales precisamente sobre la base de la excelencia de sus recursos humanos como elemento decisivo para el desarrollo de los mismos.[2]

Sin embargo, la práctica pedagógica en las Universidades, evidencia que ese elemento, por sí solo, no siempre posibilita el éxito a que se aspira. La labor

educativa supone, en general, una doble profesión: El profesor universitario está obligado a ser un especialista en la materia de estudio que enseña y a la vez debe dominar las regularidades pedagógicas que intervienen en esa labor, lo que le permite dirigirla adecuadamente hacia el logro de los objetivos trazados.[2]

Lamentablemente este segundo aspecto antes mencionado es ignorado, o cuando menos no adecuadamente priorizado, en muchas de las Universidades latinoamericanas, en las cuales el profesor “dicta” sus conferencias y con ello siente que ha cumplido con la parte que le corresponde en el complejo proceso de lograr que el estudiante se forme como profesional, y sin que la propia organización interna de esas Universidades lo conduzca de alguna manera a un mayor nivel de compromiso con los resultados del aprendizaje de los estudiantes.[2]

En la Educación Superior cubana, desde la Reforma Universitaria de 1962, y basada en una tradición pedagógica que tiene sus orígenes en nuestros principales próceres, ha existido una comprensión mucho más completa del problema antes señalado, lo que gradualmente ha propiciado que los profesores, conjuntamente con su superación científica en la ciencia que enseñan, hayan adquirido la preparación pedagógica adecuada para garantizar la efectiva conducción de este proceso. La creación del Ministerio de Educación Superior (MES) en el año 1976 una característica distintiva importante de nuestro sistema educacional- ha contribuido de modo esencial a este propósito en estos últimos 20 años, en los que se ha venido consolidando un sistema de trabajo en toda la red de Universidades de nuestro país, que usualmente se acostumbra a denominar: “trabajo metodológico”. [2]

El desarrollo del trabajo metodológico constituye una de las exigencias inherentes a la Universidad, pues sin dudas, está dirigido a lograr la mejor preparación de los docentes y garantizar por tanto mayor eficiencia en el proceso docente-educativo.

Es el núcleo de la planificación estratégica en la educación y comprende el conjunto de acciones que realizan los órganos técnicos y de dirección sobre la

actividad profesional del personal docente y las interacciones que deben producir los diferentes estilos de enseñanza sobre cada estilo de aprendizaje, mediante una constante búsqueda de posibilidades y recursos para adoptar estrategias cada vez más óptimas. Bien orientado desde la propia dirección, el trabajo metodológico es básico para la elevación de la calidad de la educación y para alcanzar los objetivos que persigue cada institución.[3]

Situación Problemática:

Hoy en día, en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” (UCF) el trabajo metodológico juega un papel muy importante. Sin embargo, el mecanismo actual por el que se gestiona los procesos de planificación, ejecución y control del trabajo metodológico es engorroso y demora más de lo deseado en concretarse.

Existe la necesidad de elaborar y entregar informes a las diferentes instancias de dirección, en los que en ocasiones algunos datos son reiterativos, así como lo complicado que resulta consolidar la entrega de dicha información en el tiempo establecido, permite identificar el siguiente ***problema científico***, “La necesidad de mejorar la gestión la información relacionada con los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos”.

Idea a Defender

La elaboración de un sistema informático para la gestión de la información relacionada con los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos posibilitará mejorar la gestión la información relacionada con dichos procesos.

Objeto de Estudio

Los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos.

Campo de Acción

La gestión de la información relacionada con los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos.

Objetivo General

Desarrollar un sistema informático para gestionar la información relacionada con los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos.

Objetivos Específicos

- ❖ Realizar el análisis para la confección del diseño del sistema informático que dará solución a la problemática planteada.
- ❖ Diseñar el sistema informático propuesto.
- ❖ Implementar el sistema informático utilizando el enfoque de desarrollo de aplicaciones WEB.
- ❖ Validar el sistema informático.

Tareas científicas:

- ❖ Caracterización del problema y la situación actual para identificar las particularidades de los procesos relacionados con la planificación, ejecución y control del trabajo metodológico.
- ❖ Entrevistas a directivos y al personal encargado de los procesos de planificación, ejecución y control del trabajo metodológico.
- ❖ Recopilación de los principales aportes teóricos relacionados con los sistemas automatizados existentes asociados al problema.
- ❖ Selección de las tecnologías a emplear para dar solución al problema planteado.
- ❖ Diseño de la interfaz de la aplicación.
- ❖ Análisis y diseño de la base de datos que almacenará la información necesaria para la aplicación.
- ❖ Instalación del software para su etapa de prueba y validación.

Aporte Práctico

El sistema propuesto tiene **valor práctico** para la institución, ya que permite gestionar digitalmente la información relacionada con los procesos de planificación, ejecución y control del trabajo metodológico en la Universidad de Cienfuegos, facilitando y agilizando dichos procesos.

El presente documento está estructurado de la siguiente manera:

Capítulo I. Fundamentación Teórica.

En el presente capítulo se abordan los aspectos teóricos del tema en análisis, se detallan los conceptos fundamentales y los sistemas ya existentes asociados al mismo, describiendo sus características y dificultades.

Se presenta la propuesta de solución y las metodologías, lenguajes y tecnologías utilizadas, así como su importancia y comparación con otras similares.

Capítulo II. Descripción del Sistema.

En este capítulo se realiza, la descripción del flujo actual de los procesos involucrados en el campo de acción junto con el objeto de automatización, los principales conceptos asociados al dominio del sistema, la elaboración del modelo de objeto, la especificación de los requerimientos que debe tener el sistema, el estudio de factibilidad de la aplicación son los aspectos que se tomarán en cuenta para la confección del siguiente capítulo.

Capítulo III. Construcción del Sistema.

En este capítulo se realiza una descripción del diseño del sistema utilizando los diagramas de clases. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. Adicionalmente se realiza la validación de la solución propuesta.

Capítulo I: Fundamentación Teórica.

1.1- Introducción.

En el presente capítulo se abordan los aspectos teóricos del tema en análisis, se detallan los conceptos fundamentales y los sistemas ya existentes asociados al mismo, describiendo sus características y dificultades.

Se presenta la propuesta de solución y las metodologías, lenguajes y tecnologías utilizadas, así como su importancia y comparación con otras similares.

1.2- Conceptos relacionados al dominio del Problema.

¿Qué es Gestión?

Gestión: Acción y efecto de gestionar. Según la Real Academia Española, gestionar es hacer diligencias conducentes al logro de un negocio o de un deseo cualquiera. [4]

¿Qué es Información?

Información: Es un conjunto de datos organizados y que tienen un significado. De esta manera, si tomamos datos por separado no tendrían un significado mientras que si los agrupamos en forma organizada, sí. La información es un elemento fundamental en el proceso de comunicación, ya que tiene un significado para quien la recibe, que la va a comprender si comparte el mismo código de quien la envía. [4]

¿Qué es Gestión de la Información?

La gestión de la información es el proceso de analizar y utilizar la información que se ha obtenido y registrado para permitir a los administradores tomar decisiones documentadas. [4]

Para desarrollar una correcta gestión de la información es necesario tener en cuenta una serie de pasos, entre los que se encuentran los siguientes: [4]

1. Determinar la información que se precisa.
2. Recoger y analizar la información.
3. Registrarla y recuperarla cuando sea necesaria.
4. Utilizarla.
5. Divulgarla.

¿Qué es un sistema informático?

Un sistema informático resulta de la interacción entre los componentes físicos que se denominan Hardware y los lógicos que se denominan Software. A estos hay que agregarles el recurso humano, parte fundamental de un sistema informático.[4]

Trabajo Metodológico.

Generalidades

El trabajo metodológico en la Universidad Cubana está definido por la Resolución No.210 de fecha 31 de Julio de 2007 del Ministro de Educación Superior en su artículo 24 como la labor que, apoyados en la Didáctica, realizan los sujetos que intervienen en el proceso docente-educativo, con el propósito de alcanzar óptimos resultados, jerarquizando la labor educativa desde la instrucción, para satisfacer plenamente los objetivos formulados en los planes de estudio. [3]

El contenido del trabajo metodológico está dado en primer lugar, por los objetivos y el contenido, los que a su vez están interrelacionados con las formas organizativas, los métodos, los medios y la evaluación del aprendizaje. [3]

El trabajo metodológico en los centros educativos tiene funciones y formas; cada instancia tiene sus propias responsabilidades en su realización. Las funciones pueden resumirse en:

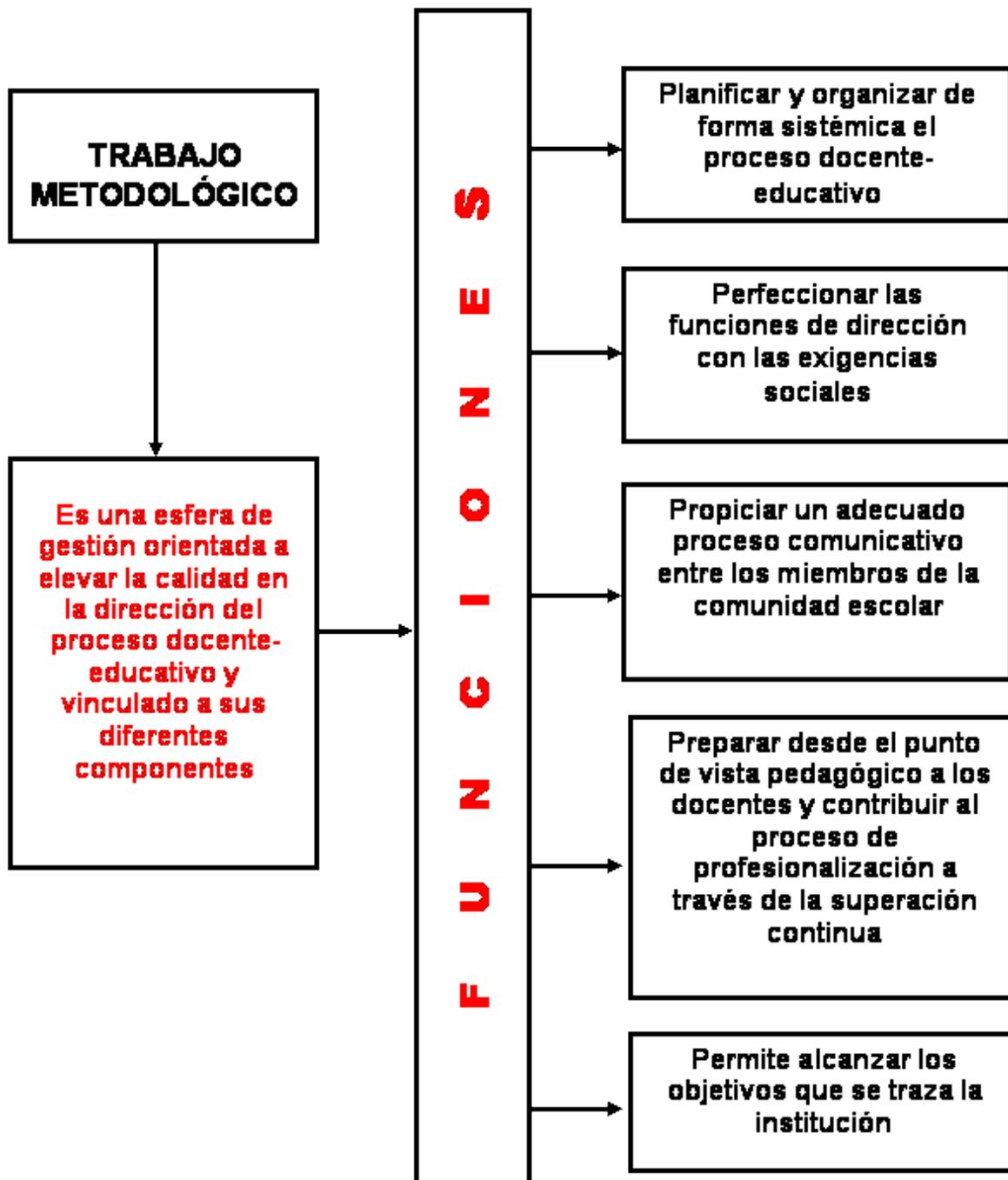


Ilustración 1: Funciones del trabajo metodológico.

El trabajo metodológico se concreta, fundamentalmente, en el desarrollo con calidad del proceso docente-educativo, logrando una adecuada integración de las clases con la actividad investigativa y laboral, así como con las tareas de alto

Capítulo I: Fundamentación Teórica

impacto social y demás tareas de carácter extracurricular que cumplen los estudiantes. [3]

Las funciones principales del trabajo metodológico son:

- ❖ La planificación
- ❖ La organización
- ❖ La regulación
- ❖ El control

El adecuado desempeño de estas funciones, que tienen como sustento esencial lo didáctico, garantiza el eficiente desarrollo del proceso docente-educativo. El profesor es el responsable fundamental de que la asignatura que imparte posea la calidad requerida, desarrollando una labor educativa desde la instrucción. Para ello debe poseer una adecuada preparación pedagógica y dominar los contenidos de la asignatura; así como orientar, controlar y evaluar a los estudiantes para lograr un adecuado dominio de dichos contenidos, en correspondencia con los objetivos generales de la asignatura, contribuyendo así a su formación integral. [3]

El trabajo metodológico se realiza tanto de forma individual como colectiva y se desarrolla tanto en las sedes centrales como en las sedes universitarias. La preparación de los profesores y tutores que laboran en las sedes universitarias es una prioridad de esta labor metodológica. [3]

El trabajo metodológico individual es la labor de autopreparación que realiza el profesor en los aspectos científico-técnico, didáctico, filosófico, político-ideológico e informáticos requeridos para el desarrollo de su labor docente. Esta autopreparación es premisa fundamental para que resulte efectivo el trabajo metodológico que realiza el profesor de forma colectiva. [3]

En los centros de educación superior, el trabajo metodológico que se realiza de forma colectiva, tiene como rasgo esencial el enfoque en sistema y se lleva a cabo en cada uno de los niveles organizativos del proceso docente-educativo, como vía

Capítulo I: Fundamentación Teórica

para su perfeccionamiento en cada nivel. Se identifican como subsistemas o niveles organizativos principales para el trabajo metodológico los siguientes:

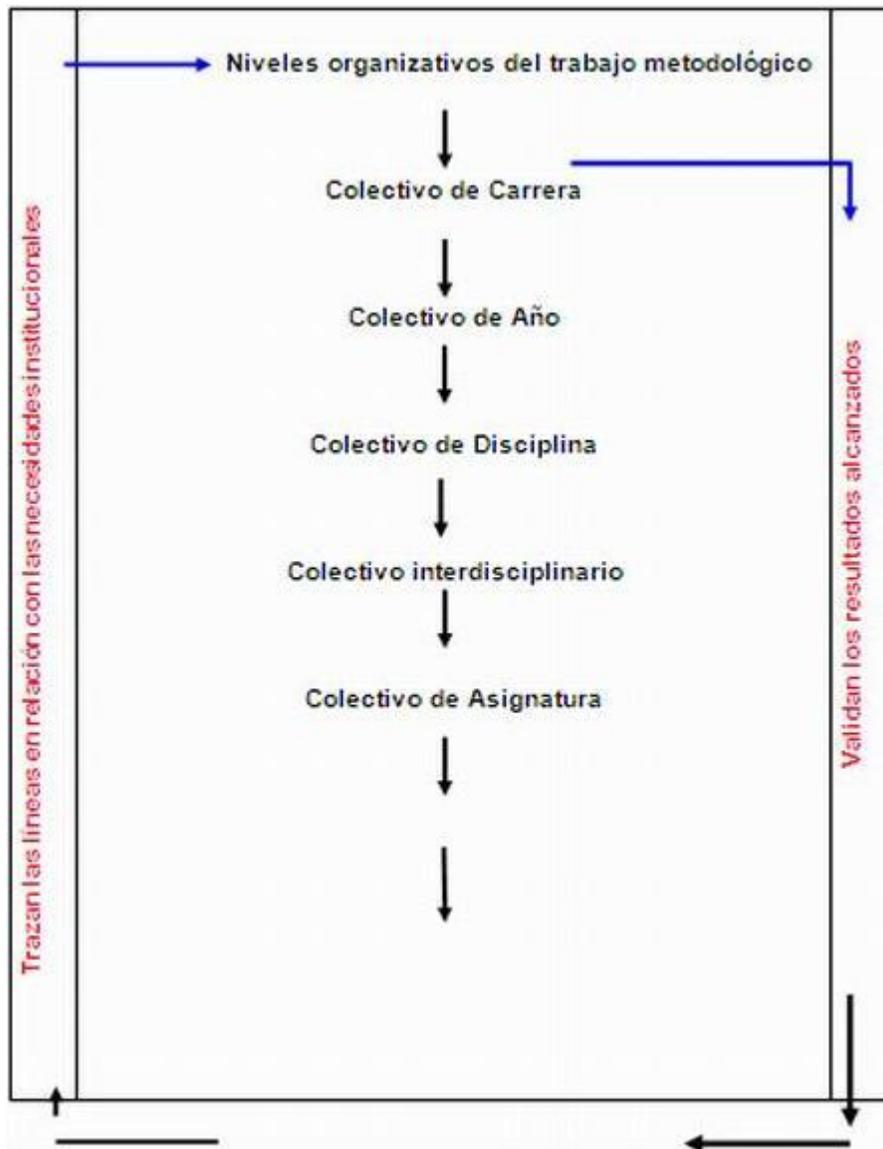


Ilustración 2: Niveles organizativos del trabajo metodológico.

Formas fundamentales del trabajo metodológico.

La propia Resolución No. 210/07 establece como formas fundamentales del trabajo metodológico: el docente-metodológico y el científico-metodológico. [3]

Capítulo I: Fundamentación Teórica

El trabajo docente- metodológico es la actividad que se realiza con el fin de mejorar de forma continua el proceso docente-educativo, basándose fundamentalmente en la preparación didáctica que posean los profesores de las diferentes disciplinas y asignaturas, así como la experiencia acumulada. [3]

El trabajo científico-metodológico es la actividad que realizan los profesores en el campo de la didáctica, con el fin de perfeccionar el proceso docente-educativo, desarrollando investigaciones o utilizando resultados de investigaciones realizadas que tributen a la formación integral de los futuros profesionales. Los resultados del trabajo científico-metodológico constituyen una de las fuentes principales que le permite al profesor el mejor desarrollo del trabajo docente metodológico. [3]

El trabajo docente-metodológico está diseñado para potenciar la superación mediante la autopreparación y el trabajo didáctico de la disciplina, y el científico-metodológico está encaminado a la investigación en la aplicación de los resultados de la práctica pedagógica como la vía más importante para complementar la formación académica y científica de los docentes, todo lo cual se traduce en una mejor preparación de los profesores y con ello una mayor calidad del proceso docente-educativo. [3]

No obstante las diferentes formas que puede adoptar el trabajo docente, y la demostración de estrategias que posibiliten un aprendizaje más significativo y que contribuyan a formar un sujeto más creativo con los valores necesarios para el desarrollo de su sociedad, son aspectos básicos a contemplar en el trabajo de dirección didáctica de un centro educativo. [3]

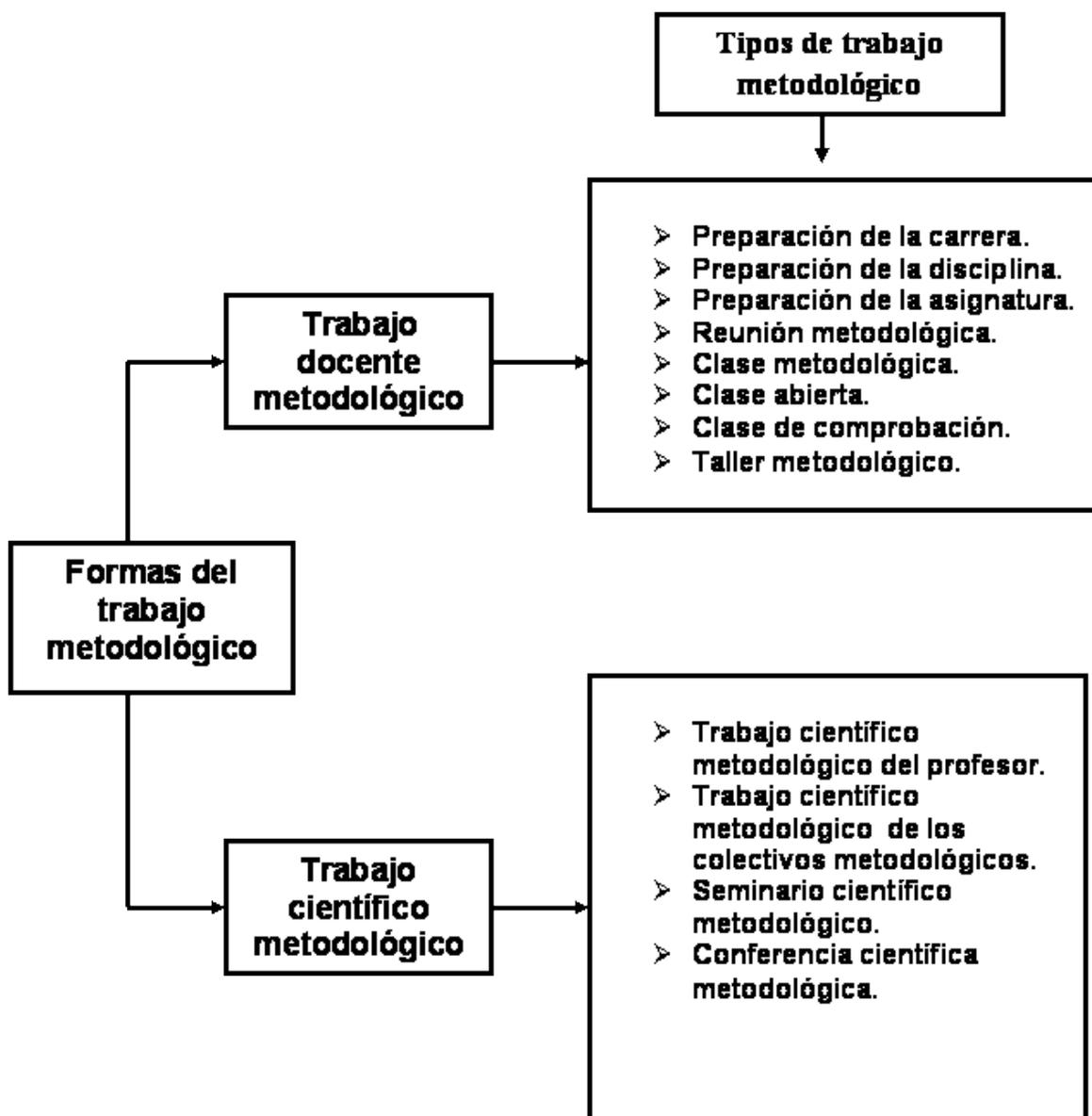


Ilustración 3: Formas del trabajo metodológico.

1.3- Descripción del objeto de estudio.

La presente investigación se desarrolló en la Universidad de Cienfuegos “Carlos Rafael Rodríguez” (UCF), institución del Ministerio de Educación Superior de Cuba encargada de formar profesionales en diversas áreas del conocimiento, para empresas y proyectos locales, nacionales e internacionales. Dicha investigación se centra en los procesos de la planificación, la ejecución y el control del trabajo metodológico en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”.

1.4- Descripción de los sistemas existentes asociados al Objeto de Estudio:

Las tecnologías de la información y la comunicación son una parte de las tecnologías emergentes que habitualmente suelen identificarse con las siglas (TIC) Tecnologías de la Información y las Comunicaciones y que hacen referencia a la utilización de medios informáticos para almacenar, procesar y difundir todo tipo de información con diferentes finalidades ya sean: formación educativa, organización y gestión empresarial o toma de decisiones en general.

La presencia de las Tecnologías de la Información y las Comunicaciones en la vida moderna es incuestionable, pues se han incorporado como parte de la cultura tecnológica que rodea al hombre y con la que debe convivir. Amplía sus capacidades físicas y mentales, y las posibilidades de su desarrollo como ser social.

Después de realizar una búsqueda sobre los sistemas relacionados la gestión y control del trabajo metodológico se encontró que los sistemas encontrados aunque presentan algunas similitudes al que se propone en el presente trabajo, no satisface las necesidades de la investigación.

1.4.1- Sistemas existentes en Cuba.

Sistema Informático para la Gestión de la Planificación Docente.

Este sistema fue desarrollado en el Departamento de Informática de la Facultad de Ingeniería Eléctrica de la Universidad de Oriente. Dicho sistema es una aplicación web que brinda la posibilidad de gestionar y planificar la docencia y las actividades metodológicas referidas al trabajo en el departamento de informática.

A pesar de los beneficios que ofrece este software, es importante destacar que está basado en la planificación docente y solo toca algo de las actividades metodológicas, además es integrado y específico de su negocio, adaptado a las necesidades del medio en que se desarrolla. [5]

1.4.2- Presentación de la Propuesta de Solución.

La propuesta de solución lleva por nombre Sistema Informático para la Gestión del trabajo metodológico en la Universidad de Cienfuegos, tomando como punto de partida los diferentes diseños informáticos de gestión previstos en su concepción.

Este Sistema constituye una vía de trabajo eficiente, pues reduce tiempo y recursos de escritorio en la gestión de la información asociada a los procesos la planificación, la ejecución y el control dentro de la Universidad, agilizando la obtención de la información, el procesamiento de los datos y por ende los reportes asociados a ellos.

A diferencia de los descritos con anterioridad, este producto de software, permite mejorar notablemente la obtención de resultados finales, facilitándole al usuario la interacción con el mismo a través de una interfaz de fácil navegación.

El sistema posibilita almacenar y exportar a formato PDF la información que obtiene.

1.5- Tendencias, lenguajes de programación, metodologías y tecnologías actuales.

A continuación se referencian las tendencias, lenguajes, metodologías y tecnologías actuales, investigadas para la realización del sistema propuesto.

1.5.1- Etapa de diseño y desarrollo.

Metodologías de desarrollo

La siguiente tabla muestra una breve comparación entre metodologías de desarrollo ágiles y metodologías de desarrollo pesadas.

Metodologías Ágiles	Metodologías Pesadas
---------------------	----------------------

Capítulo I: Fundamentación Teórica

Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Especialmente preparados para cambios durante el proyecto
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1: Comparación entre metodologías

FDD: Desarrollo Guiado en Funcionalidades

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff de Luca y se podría considerar a medio camino entre RUP y XP, aunque al seguir siendo un proceso ligero es más similar a este último.[6]

FDD está pensado para proyectos con tiempo de desarrollo relativamente corto (menos de un año). Se basa en un proceso iterativo de iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.[6]

Las iteraciones se deciden en base a las funcionalidades (de ahí el nombre del proceso) o funcionalidades, que son pequeñas partes del software con significado para el cliente. [6]

Un proyecto que sigue FDD se divide en 5 fases:

Capítulo I: Fundamentación Teórica

- ❖ Desarrollo de un modelo general.
- ❖ Construcción de una lista de funcionalidades.
- ❖ Plan de reléase en base a las funcionalidades a implementar.
- ❖ Diseñar en base a las funcionalidades.
- ❖ Implementar en base a las funcionalidades.[6]

¿Por qué se seleccionó FDD como metodología para la solución propuesta?

Como las otras metodologías adaptables, se enfoca en iteraciones cortas que entregan funcionalidad tangible. Dicho enfoque no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. Sin embargo, fue diseñado para trabajar con otras actividades de desarrollo de software y no requiere la utilización de ningún modelo de proceso específico. Además, hace énfasis en aspectos de calidad durante todo el proceso e incluye un monitoreo permanente del avance del proyecto. Al contrario de otras metodologías, FDD afirma ser conveniente para el desarrollo de sistemas críticos.

Lenguaje de modelado, UML

Es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. [7]

UML está consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. [7]

En otros términos, así como en la construcción de un edificio se realizan planos previos a su construcción, en Software se deben realizar diseños en UML previa codificación de un sistema. Ahora bien, aunque UML es un lenguaje, éste posee más características visuales que programáticas, las que facilitan a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente, estos integrantes siendo los analistas, diseñadores, especialistas de área y desde luego los programadores. [7]

1.5.2- Lenguajes y tecnologías web.

Tecnología N Capas.

La programación por capas es una arquitectura cliente-servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño [8]; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos [8].

El diseño más utilizado actualmente es el diseño en tres niveles (o en tres capas).

Capa de presentación: es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio [8].

Capa de negocio: es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él [8].

Capa de datos: es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio [8].

Arquitectura Modelo-Vista-Controlador.

La arquitectura Modelo-Vista-Controlador (MVC) separa la lógica de negocio y la presentación por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El Controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El Modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación [9]

El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos [9].

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página como un encabezado o un pie de página. También puede ser una página RSS, o cualquier otro tipo de "página" [9]

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web [9]

Tecnologías del lado del cliente.

Están insertadas en la página HTML del cliente y son interpretadas y ejecutadas por el navegador. Es decir, su correcta funcionalidad depende del soporte de la versión del navegador a ser utilizado por el usuario.

Mozilla Firefox

Mozilla Firefox es un navegador web libre y de código abierto descendiente de Mozilla Application Suite y desarrollado por la Fundación Mozilla el 9 de noviembre del 2004. Es multiplataforma, estando disponible para varios sistemas operativos como Microsoft Windows, GNU/Linux, Mac OS X, y en muchas otras plataformas.

Capítulo I: Fundamentación Teórica

Su código fuente es software libre, publicado bajo una triple licencia GNU GPL, GNU LGPL.

Internet Explorer

Windows Internet Explorer conocido comúnmente como IE, es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows desde 1995.

HTML

HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados [10]

CSS Cascading Style Sheets

Hojas de Estilo en Cascada es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos [11]

JavaScript

Es un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. Al ser un lenguaje de tipo script significa que no es un lenguaje compilado, es decir, tal cual se va leyendo se ejecuta por el cliente [11]. [12]

Tecnologías del lado del servidor

Apache

El servidor Web está elaborado para traducir lenguajes y sentencias a una interfaz entendible por el usuario. El servidor Apache es un servidor HTTP de código abierto para varias plataformas. Presenta mensajes de error altamente configurables, Base de Datos de Autenticación y negociado de contenidos. Es el servidor HTTP más usado en la actualidad. El mismo es capaz de transformar lenguaje PHP a lenguaje HTML que es el que soporta el navegador con el cual se le presenta la interfaz al usuario. Se caracteriza por su robustez, fácil de configurar y estabilidad [13]

PHP

PHP es un lenguaje de tipo script que se ejecuta en el servidor Web. Es una herramienta que permite crear páginas dinámicas y hace posible que las páginas Web sean tratadas como una página HTML común, de igual forma su creación y edición es igual a la de cualquier documento HTML. Con PHP se escribe un script incrustado en el documento HTML, cuyo resultado será visualizado directamente al desplegarse dicho documento. El código PHP está contenido en etiquetas especiales que indican el comienzo y fin del código, permitiendo conmutar entre el código PHP y HTML [14]

¿Por qué se eligió PHP como lenguaje de programación para implementar la solución propuesta?

1. Está soportado en la mayoría de las plataformas de Sistemas Operativos.
2. Soporta una gran cantidad de Bases de Datos.
3. Es el que más conocen los programadores.
4. Brinda todas las prestaciones necesarias y requeridas para el desarrollo del sistema propuesto.

5. El PHP no tiene costo oculto, cuenta con un grupo de bibliotecas importantes y en caso de necesitar alguna es posible encontrarla de forma rápida y gratis en Internet.

1.5.3- Selección del Framework.

En esta sección se realiza un análisis de los framework más utilizados en la actualidad para así facilitar la selección del framework a utilizar en el sistema propuesto.

¿Qué es un Framework?

El término framework, se refiere a una estructura software formada por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerarse como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta [15].

Por tanto un framework Web, puede definirse como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web [15].

Frameworks más utilizados en la actualidad.

Symphony

Symphony es un CMS open source creado con PHP, basado en XSLT (estándar de la organización W3C para representar documentos XML) y que utiliza base de datos MySQL, estos tres puntos dan una aplicación totalmente moldeable y perfecta para cualquier programador que quiera agregar y quitar funciones con el menor esfuerzo [16].

El potencial de este framework está en su flexibilidad además que cuenta con una amplia biblioteca de extensiones y tiene un panel de administración perfectamente

Capítulo I: Fundamentación Teórica

organizado, con un editor muy completo y un estilo minimalista, todo esto lo convierte en un entorno amigable para cualquier tipo de usuario.

Ventajas de utilizar Symphony:

1. Administración centralizada de todos los servicios.
2. Atención de solicitudes a través de múltiples canales.
3. Acceso vía web en cualquier momento.
4. Fácil de configurar y adaptable a sus necesidades.
5. Estructura altamente flexible e intuitiva.
6. Administración del catálogo de servicios.
7. Administración de niveles de servicio.
8. Autoservicio online.
9. Informes a medida del cliente.
10. Administración de la base del conocimiento.
11. Seguridad a través de perfiles.
12. Plataforma independiente.

CakePHP

CakePHP es un framework que tiene como fin ayudar a los programadores a construir aplicaciones web robustas, de fácil y rápido mantenimiento, utilizando PHP 4 o PHP 5 por su compatibilidad con estas versiones del lenguaje PHP. Para utilizar CakePHP es preciso tener conocimientos previos de PHP y HTML, además el programador deberá estar familiarizado con el MVC, modelo que hace reusable su código. [17]

CakePHP es libre, puede ser utilizado de cualquier manera pues su código de fuente es abierto, o sea, se tiene acceso al código fuente; su estructura de bibliotecas permite realizar el trabajo de manera rápida y que a su vez el resultado sea una aplicación flexible. Cake PHP por sus rasgos se convierte fácilmente en un framework deseado por una cantidad notable de programadores en el mundo [17].

Sus principales características son:

1. Compatible con PHP 4 y PHP 5.
2. Utiliza el MVC.
3. URLs personalizado.
4. Templating rápido y flexible.
5. Auxiliadores para AJAX, JavaScript, y formularios HTML.

CodeIgniter

Es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación. Contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación. Marca una manera específica decodificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener [18].

¿Por qué se eligió CodeIgniter como frameworks de PHP para la solución propuesta?

- ❖ Versatilidad: La principal característica de CodeIgniter, en comparación con otros frameworks es que es capaz de trabajar la mayoría de los entornos o servidores [9].
- ❖ Compatibilidad: es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Funciona correctamente también en PHP 5 [9].
- ❖ Facilidad de instalación: No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos [9].

- ❖ Flexibilidad: CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio [9].
- ❖ Ligereza: El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente [9].

Después de analizar los frameworks más utilizados actualmente, se concluye utilizar CodeIgniter para la realización del sistema propuesto debido a que su curva de aprendizaje es pequeña en relación con la de otros frameworks, y porque se considera suficiente para desarrollar el sistema propuesto.

1.5.4- Recursos utilizados para el desarrollo del producto informático.

Adobe Dreamweaver.

Adobe Dreamweaver es una aplicación en forma de estudio enfocada a la construcción y edición de sitios y aplicaciones Web basados en estándares. Adobe Dreamweaver permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado. Permite la conexión a un servidor, a base de datos, soporte para programación en ASP, PHP, JavaScript y cliente FTP integrado [19].

Adobe Photoshop

Adobe Photoshop se trata esencialmente de una aplicación informática que está destinada para la edición, retoque fotográfico y pintura a base de imágenes de mapa de bits. Su capacidad de retoque y modificación de fotografías le ha hecho

titularse el programa de edición de imágenes más famoso y utilizado del mundo. [20]

PhpMyAdmin

PhpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva [21].

Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones con mejor calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. [22]

1.5.5- Sistema Gestión de Base de Datos (SGBD).

MySQL

Es un sistema de gestión de bases de datos relacional. MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales [23].

Capítulo I: Fundamentación Teórica

¿Por qué se seleccionó MySQL como gestor de Base de Datos para la solución propuesta?

1. Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
2. Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
3. Facilidad de configuración e instalación.
4. Soporta gran variedad de Sistemas Operativos.
5. Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
6. Conectividad y seguridad.

MySQL es compatible con PHP, el lenguaje escogido para desarrollar el sistema propuesto por un amplio conjunto de comandos definidos para el tratamiento del mismo. [23]

PostgreSQL

PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo [24]. PostgreSQL proporciona un gran número de características que normalmente solo se encontraban en las bases de datos comerciales tales como DB2 u Oracle.

Posee como principales características:

1. Altamente extensible: soporta operadores, funcionalidades de acceso y tipos de datos definidos por el usuario.
2. Soporte SQL comprensivo: PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
3. Integridad referencial: PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

4. Cliente/Servidor: PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL [24].

1.6- Conclusiones del capítulo.

En este capítulo se realizó un estudio de los diferentes conceptos relacionados al dominio del problema, así como un análisis de las diferentes tecnologías y herramientas a utilizar en el software propuesto.

Se seleccionó la metodología FDD, como guía para la documentación del software propuesto y UML como lenguaje para modelar el análisis y diseño.

Se seleccionó además como lenguaje de programación PHP, usando el lenguaje HTML y JavaScript para generar los scripts del lado del cliente.

Se seleccionó MySQL como gestor de base de datos.

Se eligió como framework del lado del servidor para desarrollar la aplicación CodeIgniter.

Se eligió Apache como servidor Web, Visual Paradigm, PhpMyAdmin, Adobe Photoshop, Adobe Dreamweaver CS3 como herramientas de desarrollo.

Esta elección fue realizada en base a las potencialidades de cada una de las herramientas y lenguajes para realizar la implementación del sistema informático propuesto.

Capítulo II: Descripción del Sistema.

2.1- Introducción.

La descripción de las distintas fases de la metodología FDD, la descripción del flujo actual de los procesos involucrados en el campo de acción junto con el objeto de automatización, los principales conceptos asociados al dominio del sistema, la elaboración del modelo de objeto, la especificación de los requerimientos que debe tener el sistema, el estudio de factibilidad de la aplicación son los aspectos que se tomarán en cuenta para la confección del siguiente capítulo.

2.2- Desarrollo de un modelo global.

Cuando esta fase comienza, los expertos del dominio ya tienen una idea del contexto y los requerimientos del sistema. Es probable que el documento de especificación de requerimientos ya exista. Sin embargo, FDD no hace énfasis en la recolección y administración de los requerimientos. [6]

El dominio global es dividido en diferentes áreas y se realiza un informe detallado para cada una de dichas áreas por parte de los expertos del dominio. Luego de cada informe, un grupo de desarrolladores realizan un modelo de objetos para el área del dominio. Además, el equipo de desarrollo discute y decide cual es el modelo de objetos más apropiado para cada área del dominio. Simultáneamente, el modelo global del sistema es construido.[6]

2.2.1- Flujo actual de los procesos involucrados en el campo de acción.

Una de las actividades fundamentales referentes a la planificación, ejecución y control del trabajo metodológico en la universidad de Cienfuegos "Carlos Rafael Rodríguez" es la gestión del plan de trabajo metodológico y el control a las actividades metodológicas. Esta gestión es realizada por diferentes niveles de instancias.

Capítulo II: Descripción del Sistema.

La actividad de planificación del plan de trabajo metodológico funciona de la siguiente manera:

El vicerrectorado de formación de la Universidad elabora un grupo de objetivos generales los cuales llegan a los diferentes niveles.

Los niveles se encuentran organizados como se muestra en la siguiente figura:

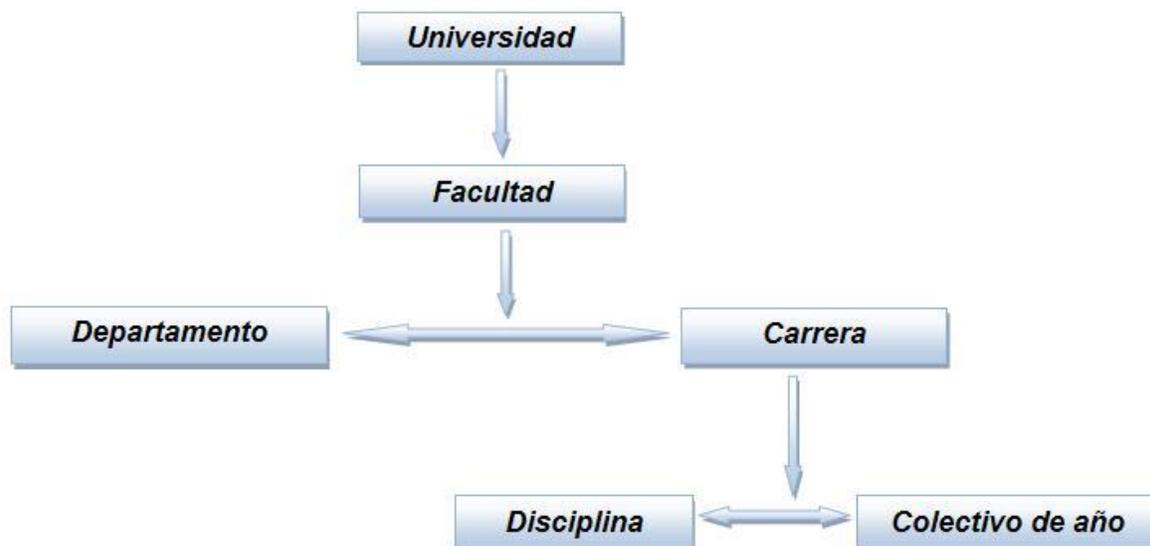


Ilustración 4: Niveles jerárquicos del trabajo metodológico

Estos son adecuados en cada uno de los niveles de acuerdo con sus condiciones particulares.

El jefe de carrera y el jefe de departamento diseñan el plan de trabajo metodológico tomando como punto de partida el balance del curso anterior, las indicaciones de vicerrectorado de formación y las exigencias del modelo del profesional y lo envían a los jefes de disciplina y jefe de colectivo de año los cuales le agregan sus objetivos específicos, respectivamente.

El control a las actividades metodológicas funciona de la siguiente manera:

El plan de trabajo metodológico contiene entonces un conjunto de actividades metodológicas que responden a los objetivos en las cuales se señalan, el nombre de la misma, los participantes, el responsable, la fecha y lugar entre otros datos.

Capítulo II: Descripción del Sistema.

Las actividades metodológicas se realizan según su nivel y estos pueden ser: Universidad, Facultad, Departamento, Carrera, Disciplinas, Colectivo de año.

Dicho responsable de la actividad es el encargado de controlar que la actividad se cumpla con los requisitos o indicadores establecidos.

El plan de control a clases solo es conocido por los responsables de cada nivel.

Las actividades metodológicas se encuentran organizadas como se muestra en la siguiente figura:

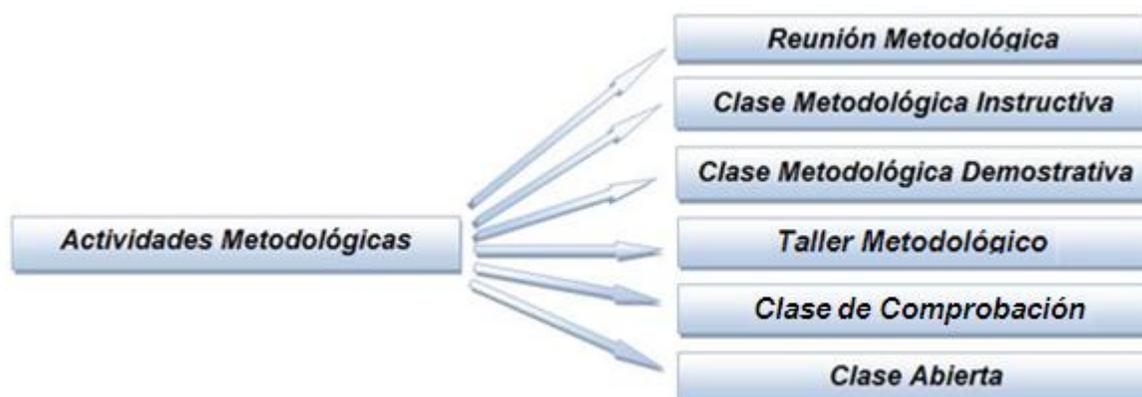


Ilustración 5: Tipo de actividades metodológicas

De la ejecución de las actividades del plan de trabajo metodológico se genera una gran cantidad de información a partir de la cual se elaboran distintos informes para entregar a los directivos que la soliciten. Los directivos también pueden solicitar información referente al claustro de profesores de una carrera.

2.2.2- Objeto de automatización.

Se desea automatizar los procesos relacionados con la planificación, ejecución y control del trabajo metodológico. Entre los cuales resalta el proceso de planificación del plan de trabajo metodológico el cual es desarrollado por diferentes instancias desde el vicerrectorado de formación hasta el colectivo de año de una carrera.

Capítulo II: Descripción del Sistema.

Además se pretende automatizar el control de las actividades del plan de trabajo metodológico de la Facultad en los diferentes niveles organizativos. Así como el cumplimiento del plan de control a clase. Generando informes, los cuales se entregan a las distintas instancias.

2.2.3- Modelo de dominio. Diagrama de objetos.

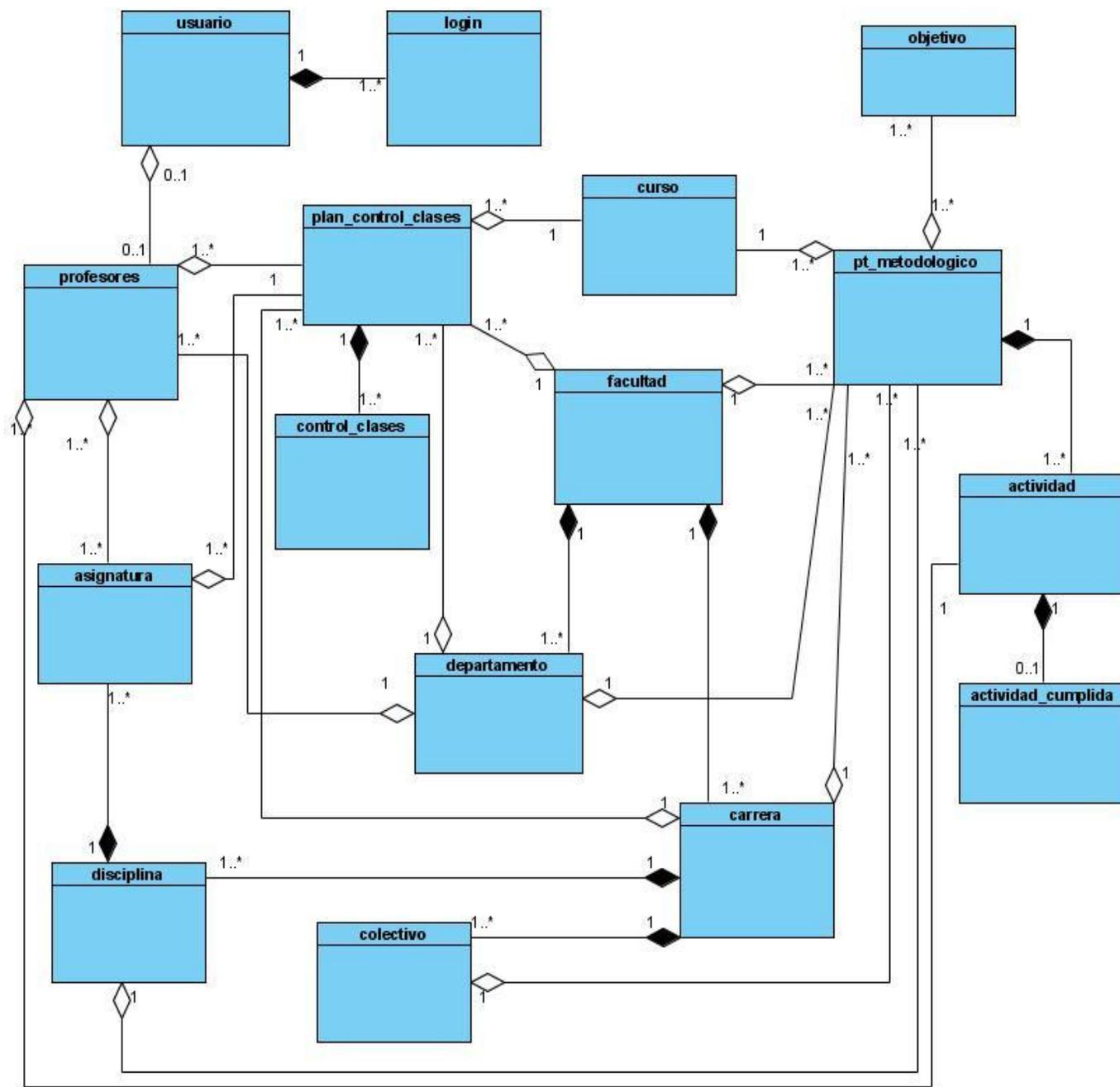


Ilustración 6: Modelo de objetos.

2.2.4- Reglas a considerar.

1. El plan de trabajo metodológico solo es gestionado según su nivel organizativo.
2. El plan de control a clases solo es gestionado según su nivel organizativo
3. Las actividades metodológicas solo pueden ser gestionadas según su nivel organizativo.
4. Los directivos son los únicos que pueden solicitar informes.
5. Solo el responsable del control a clases según su nivel organizativo puede llenar el acta del control a clases.
6. Solo el nivel organizativo Universidad puede crear una facultad.
7. Solo el nivel organizativo Facultad puede crear una carrera.
8. Solo el nivel organizativo Facultad puede crear un departamento.
9. Solo el nivel organizativo Carrera puede crear una disciplina.
10. Solo el nivel organizativo Carrera puede crear un colectivo de año.
11. Todos los niveles organizativos tendrán un solo plan de control a clases en un curso escolar.
12. Solo podrán tener un plan de control a clases en el curso escolar los niveles organizativos directivos.
13. Los profesores de un departamento solo podrán ser gestionados por el nivel organizativo Departamento.

2.3- Construcción de una lista de funcionalidades.

El modelo de objetos y los requerimientos existentes ofrecen una buena base para construir una lista de funcionalidades que resuma la funcionalidad del sistema a ser desarrollado. En dicha lista, el equipo de desarrolladores presenta cada una de las funcionalidades evaluadas por el cliente. Las funcionalidades son presentadas por cada área del dominio y éstas forman una lista de funcionalidades. Dicha lista es dividida en subconjuntos en base a la funcionalidad. Estas representan diferentes actividades con un área específica del dominio. La lista de funcionalidades es

Capítulo II: Descripción del Sistema.

revisada por los usuarios y sponsors del sistema para su validación y aprobación.
[6]

2.3.1- Requerimientos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. Son capacidades o condiciones que el sistema debe cumplir definiendo el comportamiento interno del software así como los comportamientos del sistema. [25]

Funcionalidades de planificación del plan de trabajo metodológico:

Nro.	Funcionalidad	Prioridad
1	Insertar plan de trabajo metodológico.	A
2	Gestionar plan de trabajo metodológico.	A
3	Buscar plan de trabajo metodológico.	A
4	Insertar objetivos al plan de trabajo metodológico.	A
5	Gestionar objetivos del plan de trabajo metodológico.	A
6	Mostrar plan de trabajo metodológico según su nivel organizativo.	A
7	Insertar nuevo curso escolar	A
8	Listar planes de trabajo metodológico de cursos anteriores.	A
9	Exportar a PDF el plan de trabajo metodológico.	B
10	Imprimir el plan de trabajo metodológico.	B

Tabla 2: Primer bloque de funcionalidades

Funcionalidades de los diferentes niveles organizativos:

Nro.	Funcionalidad	Prioridad
11	Insertar facultad	A
12	Gestionar facultad	A
13	Insertar departamento	A
14	Gestionar departamento	A
15	Insertar carrera	A
16	Gestionar carrera	A
17	Insertar colectivo de año	A

Capítulo II: Descripción del Sistema.

18	Gestionar colectivo de año	A
19	Insertar disciplina	A
20	Gestionar disciplina	A
21	Insertar asignatura	A
22	Gestionar asignatura	A

Tabla 3 : Segundo bloque de funcionalidades

Funcionalidades del control a las actividades metodológicas:

Nro.	Funcionalidad	Prioridad
23	Insertar actividad metodológica.	A
24	Gestionar actividad metodológica.	A
25	Mostrar actividades metodológicas según su nivel organizativo.	A
26	Insertar cumplimiento de la actividad metodológica.	A
27	Modificar cumplimiento de la actividad metodológica.	A
28	Listar cumplimiento de las actividades metodológicas.	A
29	Obtener informe sobre el cumplimiento de las actividades metodológicas.	A
30	Exportar a PDF el informe sobre el cumplimiento de las actividades metodológicas.	B
31	Ayuda del sistema.	B
32	Visualizar actividades metodológicas no cumplidas	A
33	Insertar plan de control a clases.	A
34	Gestionar plan de control a clases.	A
35	Insertar control a clases.	A
36	Gestionar control a clases.	A
37	Exportar a PDF el plan de control a clases.	B
38	Imprimir el plan de control a clases	B
39	Mostrar guía para el control a clases.	B
40	Mostrar modelo de control a clases.	B
41	Insertar profesor de un departamento.	A
42	Gestionar profesor de un departamento.	A
43	Buscar profesor.	A
44	Obtener informe sobre el claustro de profesores de una carrera.	B
45	Exportar a PDF claustro de profesores de una carrera	B
46	Imprimir claustro de profesores de una carrera	B
47	Calcular el porcentaje de actividades metodológicas	B

Capítulo II: Descripción del Sistema.

	cumplidas hasta la fecha.	
--	---------------------------	--

Tabla 4: Tercer bloque de funcionalidades.

Funcionalidades de Administración:

Nro.	Funcionalidad	Prioridad
48	Insertar usuario	A
49	Modificar usuario	A
50	Eliminar usuario	A
51	Listar usuario	A
52	Autenticarse	A
53	Cambiar contraseña	A

Tabla 5: Cuarto bloque de funcionalidades.

2.3.2- Requerimientos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. [25]

Requerimientos de apariencia o interfaz externa:

- ❖ El sistema debe tener una interfaz sencilla, legible y simple de usar.
- ❖ La interfaz debe ser diseñada respetando parámetros de diseño.

Requerimientos de usabilidad:

- ❖ La herramienta será utilizada solo por personas que sean usuarios del sistema y que previamente se les haya asignado una cuenta dentro del mismo, por parte del administrador, para posibilitar la navegación. Esta cuenta pertenece a un tipo de usuario y acorde con ello serán otorgados los privilegios de navegación.

Capítulo II: Descripción del Sistema.

Requerimientos de rendimiento:

- ❖ El sistema propuesto debe ser rápido en el procesamiento de la información así como a la hora de dar respuesta solicitudes de los usuarios. No debe tardarse más de 5 segundos para realizar acciones complejas, que no dependan del acceso a las colecciones de datos.

Requerimiento de Portabilidad:

- ❖ La plataforma seleccionada para desarrollar el sistema fue Windows, pero puede ser usado desde otras plataformas como LINUX que soporten PHP como lenguaje y MySQL como Sistema Gestor de Bases de Datos.

Requerimientos de seguridad

- ❖ Se debe garantizar un control estricto sobre la seguridad de la información teniendo en cuenta el establecimiento de niveles de acceso. Se define una política de usuarios con privilegios de acuerdo a su rol, lo que asegura que la información pueda ser consultada de acuerdo a su nivel de acceso.
- ❖ Accederán de manera rápida y operativa los usuarios sin que los requerimientos de seguridad se conviertan en un retraso para usar el sistema.
- ❖ Se utilizará un mecanismo de encriptación para las contraseñas que por cuestiones de seguridad no deben viajar al servidor en texto plano. Se guardará encriptado en la base de datos usando para ello la función hash MD5: La función hash es un tipo de algoritmo que calcula un mensaje implícito o valor hash para cualquier mensaje que se le indique. El valor generado por el algoritmo no es importante, lo importante es que el resultado sea fijo, es decir, que sea el mismo cada vez que se utiliza una entrada dada, que sea pequeño y que el algoritmo sea rápido. La función MD5 calcula y devuelve el hash MD5 que consiste en una cadena de 32 números en formato hexadecimal.[25]
- ❖ Solo tendrá acceso al control de los usuarios el administrador del sistema.

Capítulo II: Descripción del Sistema.

Requerimiento de Hardware:

- ❖ Las máquinas que utilizarán el software tienen que estar conectadas a la red.
- ❖ Los requerimientos mínimos para una estación de trabajo deben ser de 128 MB de RAM, Microprocesador a 800 MHz, 10GB de disco duro e impresora Epson LX300+ o superior.
- ❖ Los requerimientos mínimos para el servidor del software deben ser de 1GB de RAM, Microprocesador a 1.4 GHz, 40 GB de disco duro.

Requerimientos de software:

- ❖ Los requerimientos mínimos para una estación de trabajo deben ser:
 - Navegador Web.
- ❖ Se requiere además, una máquina servidor que cuente con:
 - La versión 2.2.6 o superior del servidor Web Apache.
 - Lenguaje de programación PHP 5.2.5.
 - Servidor de base de datos MySQL 5.0.45 o superior.
 - Navegador Web.

2.4- Estudio de la factibilidad.

Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de función. Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas. [26]

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Éste método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code. [26]

Capítulo II: Descripción del Sistema.

En este epígrafe se propone la utilización combinada de dos métodos (Puntos de Función y Cocomo) los cuales tienden a proporcionar una estimación más precisa de la siguiente forma:

Primero la aplicación del método de Puntos de Función para determinar las sentencias de código del proyecto software.

En segundo lugar la aplicación del método COCOMO II partiendo de la información producida por el anterior para llegar a una estimación precisa de las horas-hombre a aplicar y fundamentalmente a la estimación de la duración total del proyecto.

Después clasificar cada una de las funcionalidades previstas se arribó a los siguientes resultados:

Nombre de la salida externa	Cantidad de ficheros referenciados	Cantidad de Elementos de datos	Complejidad (Bajo, Medio y Alto)
Insertar plan de trabajo metodológico.	3	11	Medio
Gestionar plan de trabajo metodológico	4	12	Alto
Insertar objetivos al plan de trabajo metodológico.	1	3	Medio
Gestionar objetivos al plan de trabajo metodológico.	2	4	Medio
Insertar nuevo curso escolar	1	1	Bajo
Insertar facultad	1	2	Bajo
Gestionar facultad	2	3	Medio
Insertar departamento	1	3	Bajo
Gestionar departamento	2	5	Medio
Insertar carrera	2	2	Bajo
Gestionar carrera	2	5	Medio
Insertar colectivo de año	1	4	Bajo
Gestionar colectivo de año	2	6	Medio
Insertar disciplina	1	3	Bajo
Gestionar disciplina	2	4	Medio

Capítulo II: Descripción del Sistema.

Insertar asignatura	1	3	Bajo
Gestionar asignatura	2	4	Medio
Insertar profesor de un departamento.	2	5	Bajo
Gestionar profesor de un departamento.	2	5	Medio
Insertar actividad metodológica.	1	6	Bajo
Gestionar actividad metodológica.	2	8	Medio
Insertar cumplimiento de la actividad metodológica	2	5	Bajo
Modificar cumplimiento de la actividad metodológica	4	7	Medio
Insertar plan de control a clases.	1	5	Bajo
Gestionar plan de control a clases.	2	6	Medio
Insertar control a clases.	2	15	Medio
Gestionar control a clases.	3	17	Alto
Insertar usuario	1	4	Bajo
Gestionar usuario	2	3	Medio
Cambiar contraseña	1	1	Medio
Autenticarse	1	2	Bajo

Tabla 6: Entradas externas.

Nombre de la salida externa	Cantidad de ficheros referenciados	Cantidad de Elementos de datos	Complejidad (Bajo, Medio y Alto)
Exportar a PDF el plan de trabajo metodológico.	5	28	Alto
Imprimir el plan de trabajo metodológico.	5	28	Medio
Obtener informe sobre el cumplimiento de las actividades metodológicas	3	14	Medio
Ayuda del sistema.	3	14	Medio
Imprimir el informe sobre el cumplimiento de las actividades	1	14	Medio

Capítulo II: Descripción del Sistema.

metodológicas			
Exportar a PDF el plan de control a clases.	2	6	Bajo
Imprimir el plan de control a clases	1	6	Bajo
Obtener informe sobre el claustro de profesores de una carrera.	4	7	Medio
Exportar a PDF claustro de profesores de una carrera	4	7	Medio
Imprimir claustro de profesores de una carrera	1	7	Bajo

Tabla 7: Salidas externas.

Nombre de la consulta externa	Cantidad de ficheros Referenciados	Cantidad de Elementos de datos	Complejidad (Bajo, Medio y Alto)
Buscar plan de trabajo metodológico.	3	2	Medio
Mostrar plan de trabajo metodológico según su nivel organizativo.	5	28	Alto
Listar planes de trabajo metodológico de cursos anteriores.	5	7	Medio
Mostrar actividades metodológicas según su nivel organizativo	2	8	Medio
Listar cumplimiento de las actividades metodológicas	3	5	Medio
Visualizar actividades metodológicas no cumplidas	3	8	Medio
Mostrar guía para el control a clases.	1	2	Bajo
Mostrar modelo de control a clases.	1	2	Bajo
Buscar profesor.	4	3	Medio
Calcular el porcentaje de	3	6	Medio

Capítulo II: Descripción del Sistema.

actividades metodológicas cumplidas hasta la fecha.			
Listar usuario	1	4	Bajo

Tabla 8: Consultas externas.

Nombre del fichero lógico interno	Cantidad de ficheros referenciados	Cantidad de Elementos de datos	Complejidad (Bajo, Medio y Alto)
Usuario	1	5	Bajo
Login	1	5	Bajo
Curso	3	1	Bajo
Facultad	3	3	Bajo
Departamento	3	3	Bajo
Carrera	4	3	Bajo
Colectivo	2	3	Bajo
Disciplina	3	3	Bajo
Asignatura	3	3	Bajo
Profesor	4	6	Bajo
Pt_metodológico	7	7	Medio
Objetivos	1	2	Bajo
Actividad	3	5	Bajo
Acitividad_cumplida	1	4	Bajo
Plan_control_clase	5	4	Medio
Control_clase	1	14	Bajo

Tabla 9: Archivos lógicos internos.

Elementos	Baja	X Peso	Media	X Peso	Alta	X Peso	Subtotal de puntos de función
Entradas externas	13	3	16	4	2	6	115
Salidas externas	3	4	6	5	1	7	40
Consultas externas	3	3	7	3	1	6	36
Ficheros lógicos internos	14	7	2	10	0	15	118
Total							309

Tabla 10: Puntos de función desajustados.

Capítulo II: Descripción del Sistema.

Cálculo de:	Valor	Justificación
RCPX	1	BD moderada, se requiere de una documentación básica. La aplicación Web tiene una moderada complejidad y una alta confianza de software requerida. (Nominal)
RUSE	1	Se implementa código reusable para el aprovechamiento de este en toda la aplicación. (Nominal)
PDIF	1	No posee grandes restricciones en cuanto al tiempo de ejecución debido a que el software podrá estar trabajando sin límite de tiempo, ni limitación de memoria impuesta. La plataforma de aplicación tiene gran estabilidad. (Nominal)
PERS	0.63	Alta capacidad del analista, alta capacidad del programador, no existe movimiento del personal. (Muy alto)
PREX	0.87	El equipo tiene dominio y posee conocimiento del lenguaje de programación y las herramientas utilizadas. Con una experiencia de aproximadamente un año. (Alto)
FCIL	0.87	Se utilizan herramientas de programación como: Macromedia Dreamweaver, así como la herramienta Visual Paradim para la documentación, empleando como notación UML. (Alto)
SCED	1	La planificación se hace con moderada frecuencia. (Nominal)

Tabla 11: Multiplicadores de Esfuerzo

Capítulo II: Descripción del Sistema.

Cálculo de:	Valor	Justificación
PREC	2.40	El equipo de desarrollo posee una comprensión considerable de los objetivos del producto, no tiene experiencia en la realización de software de este tipo. (Alto)
FLEX	3.04	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa. (Nominal)
TEAM	2.83	El equipo que va a desarrollar el software es altamente cooperativo. (Alto).
RESL	3.29	Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo. (Alto)
PMAT	4.24	Nivel I Alto porque se encuentra en su primera etapa un poco avanzada. (Alto)

Tabla 12: Factor de Escala.

Multiplicador de esfuerzos

$$E_M = \prod (ME_i) = RCPX \times RUSE \times PDIF \times PERS \times PREX \times FCIL \times SCED$$

$$E_M = \prod (ME_7) = 1 \times 1 \times 1 \times 0.63 \times 0.87 \times 0.87 \times 1$$

$$E_M = 0.476847 \approx 0.48$$

Capítulo II: Descripción del Sistema.

Factores de escala

$$SF = \sum SF_i = PREC + FLEX + TEAM + RESL + PMAT$$

$$SF = \sum SF_i = 2.40 + 3.04 + 2.83 + 3.29 + 4.24$$

$$SF = 15.80$$

Valores de los coeficientes

A: se toma el valor por defecto del modelo, ajustado en 2.94.

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza Java (factor de conversión medio para lenguajes orientados a objetos = 30 SLOC/UFP). Entonces: [26]

UFP: Puntos de función desajustados 309

$$PM_{nominal} = A \times (Size)^B$$

$$Size = UFP \times FC$$

$$Size = 309 \times 30 = 9270 \text{ SLOC} \approx 9.3 \text{ KLOC}$$

$$B = 0.91 + 0.01 \times SF$$

$$B = 0.91 + 0.01 \times 15.80$$

$$B = 1.068$$

Esfuerzo nominal

$$PM_{nominal} = A \times (Size)^B$$

$$PM_{nominal} = 2.94 \times (9.3 \text{ KLOC})^{1.068} = 31.81 \text{ Meses} - \text{ Hombre}$$

Capítulo II: Descripción del Sistema.

Ajuste del esfuerzo

$$PM_{ajustado} = PM_{nominal} \times \prod(ME_i)$$

$$PM_{ajustado} = 30.81 \times 0.48 = 14.78 \text{ Meses} - \text{Hombres} \approx 14 \text{ Meses} - \text{Hombre}$$

Este resultado se interpreta como el tiempo requerido para que una persona desarrolle los cuatro módulos de la aplicación.

Se asume como salario promedio mensual \$350.00.

CH: Cantidad de hombres =2.

Tiempo: Tiempo total de desarrollo del software.

$$\text{Tiempo} = PM_{ajustado} / CH$$

$$\text{Tiempo} = 14 / 2 = 7 \text{ meses}$$

Costo

$$\text{Costo} = \text{SalarioPromedio} \times CH \times \text{Tiempo}$$

$$\text{Costo} = 350 \times 2 \times 7 \text{ Meses} = \$4900$$

Cálculo	Valor
Esfuerzo nominal	31.81Meses – Hombre
Esfuerzo ajustado	14 Meses – Hombre
Cantidad de hombres	2
Salario promedio	\$350.00
Tiempo	7 Meses
Costo	\$4900

Tabla 13: Resumen de resultados

2.4.1- Beneficios tangibles e intangibles.

Entre los beneficios tangibles obtenidos con la realización del software se pueden mencionar un ahorro sustancial de recursos humanos y materiales de oficina, lo

Capítulo II: Descripción del Sistema.

que permite una disminución del nivel de errores y los retrasos en el logro de los resultados finales. Todo lo anteriormente mencionado puede resumirse en la posibilidad de gestionar rápida y eficientemente la información asociada a los procesos ya mencionados.

2.4.2- Análisis de costos y beneficios.

El análisis de los beneficios que trae consigo la implementación del producto informático justifica su desarrollo. La solución propuesta “Sistema Informático para la Gestión del trabajo metodológico en la Universidad de Cienfuegos”, parte de un estudio realizado y surge para posibilitar una mejor gestión de la información asociada a estos procesos de manera que exista un control más estricto en cuanto a los mismos. Analizando el costo del proyecto, \$4900 MN, los beneficios que promete su puesta en marcha y dando cumplimiento a la necesidad de automatizar la gestión de la información de la Planificación y el control del trabajo metodológico

2.5- Planeación por funcionalidad.

En esta etapa se incluye la creación de un plan de alto nivel, en el cual la lista de funcionalidades es ordenada en base a la prioridad y a la dependencia entre cada funcionalidad. Además, las clases identificadas en la primera etapa son asignadas a cada programador.[6]

Funcionalidades de planificación del plan de trabajo metodológico:

Nro.	Funcionalidad	Prioridad	Fecha Inicial	Fecha Final	Días
1	Insertar plan de trabajo metodológico.	A	2/10/12	5/10/12	3
2	Gestionar plan de trabajo metodológico.	A	5/10/12	9/10/12	4
3	Buscar plan de trabajo metodológico.	A	9/10/12	12/10/12	3
4	Insertar objetivos al plan de	A	12/10/12	15/10/12	3

Capítulo II: Descripción del Sistema.

	trabajo metodológico.				
5	Gestionar objetivos del plan de trabajo metodológico.	A	15/10/12	19/10/12	4
6	Mostrar plan de trabajo metodológico según su nivel organizativo.	A	19/10/12	23/10/12	4
7	Insertar nuevo curso escolar	A	23/10/12	27/10/12	4
8	Listar planes de trabajo metodológico de cursos anteriores.	A	27/10/12	31/10/12	4
9	Exportar a PDF el plan de trabajo metodológico.	B	31/10/12	4/11/12	4
10	Imprimir el plan de trabajo metodológico.	B	4/11/12	8/11/12	4

Tabla 14: Primer bloque de funcionalidades planeado.

Funcionalidades del control a las actividades metodológicas:

11	Insertar facultad	A	8/11/12	11/11/12	3
12	Gestionar facultad	A	11/11/12	14/11/12	3
13	Insertar departamento	A	14/11/12	17/11/12	3
14	Gestionar departamento	A	17/11/12	20/11/12	3
15	Insertar carrera	A	20/11/12	23/11/12	3
16	Gestionar carrera	A	23/11/12	26/11/12	3
17	Insertar colectivo de año	A	26/11/12	29/11/12	3
18	Gestionar colectivo de año	A	29/11/12	2/12/12	3
19	Insertar disciplina	A	2/12/12	5/12/12	3
20	Gestionar disciplina	A	5/12/12	8/12/12	3
21	Insertar asignatura	A	8/12/12	12/12/12	3
22	Gestionar asignatura	A	12/12/12	16/12/12	3

Tabla 15: Segundo bloque de funcionalidades planeado.

Funcionalidades del control a las actividades metodológicas:

23	Insertar actividad metodológica.	A	16/12/12	19/12/12	3
24	Gestionar actividad metodológica.	A	19/12/12	22/12/12	3

Capítulo II: Descripción del Sistema.

25	Mostrar actividades metodológicas según su nivel organizativo.	A	7/1/13	10/1/13	3
26	Insertar cumplimiento de la actividad metodológica	A	10/1/13	23/1/13	4
27	Modificar cumplimiento de la actividad metodológica.	A	7/1/13	11/1/13	4
28	Listar cumplimiento de las actividades metodológicas.	A	11/1/13	14/1/13	3
29	Obtener informe sobre el cumplimiento de las actividades metodológicas.	A	14/1/13	18/1/13	4
30	Exportar a PDF el informe sobre el cumplimiento de las actividades metodológicas.	B	18/1/13	22/1/13	4
31	Ayuda del sistema.	B	22/1/13	25/1/13	3
32	Visualizar actividades metodológicas no cumplidas.	A	25/1/13	28/1/13	3
33	Insertar plan de control a clases.	A	28/1/13	31/1/13	3
34	Gestionar plan de control a clases.	A	31/1/13	4/2/13	4
35	Insertar control a clases.	A	4/2/13	7/2/13	3
36	Gestionar control a clases.	A	7/2/13	11/2/13	4
37	Exportar a PDF el plan de control a clases.	B	11/2/13	8/2/13	3
38	Imprimir el plan de control a clases	B	8/2/13	11/2/13	3
39	Mostrar guía para el control a clases.	B	11/2/13	14/2/13	2
40	Mostrar modelo de control a clases.	B	14/2/13	16/2/13	2
41	Insertar profesor de un departamento.	A	16/2/13	19/2/13	3
42	Gestionar profesor de un departamento.	A	19/2/13	23/2/13	4
43	Buscar profesor.	A	23/2/13	27/2/13	4
44	Obtener informe sobre el claustro de profesores de una carrera.	B	27/2/13	2/3/13	4

Capítulo II: Descripción del Sistema.

45	Exportar a PDF claustro de profesores de una carrera	B	2/3/13	6/3/13	4
46	Imprimir claustro de profesores de una carrera	B	6/3/13	9/3/13	3
47	Calcular el porcentaje de actividades metodológicas cumplidas hasta la fecha.	B	9/3/13	13/3/13	4

Tabla 16: Tercer bloque de funcionalidades planeado.

Funcionalidades de Administración:

48	Insertar usuario	A	13/3/13	17/3/13	4
49	Modificar usuario	A	17/3/13	21/3/13	4
50	Eliminar usuario	A	21/3/13	25/3/13	4
51	Listar usuario	A	25/3/13	29/3/13	4
52	Autenticarse	A	29/3/13	3/4/13	4
53	Cambiar contraseña	A	3/4/13	7/4/13	4

Tabla 17: Cuarto bloque de funcionalidades planeado.

2.6- Diseño por funcionalidad.

Un conjunto de funcionalidades son seleccionadas de la lista de funcionalidades.

El diseño y construcción de la funcionalidad es un proceso iterativo durante el cual las funcionalidades seleccionadas son producidas. Una iteración puede llevar desde unos pocos días a un máximo de dos semanas. Este proceso iterativo incluye tareas como inspección del diseño, codificación, prueba unitaria, integración e inspección del código. Luego que la iteración llega a su fin se realiza una construcción de la funcionalidad en el cual esta es integrada.[6]

2.6.1- Principios del diseño.

Los elementos gráficos o textuales que componen la interfaz son claros y de fácil identificación. Se hizo uso de términos y conceptos que se tomaron de la experiencia de las personas que más utilizarán el sistema. Adicionalmente se tuvieron presente los siguientes aspectos:

- ❖ Las operaciones comparables se activan de la misma forma.

Capítulo II: Descripción del Sistema.

- ❖ El sistema no provoca sorpresa a los usuarios.
- ❖ Se incluyeron mecanismos para permitir a los usuarios recuperarse de los errores mediante la confirmación de acciones destructivas.
- ❖ La interfaz provee retroalimentación significativa y características de ayuda sensible al contexto.
- ❖ La interfaz provee características de interacción apropiada para los diferentes tipos de usuarios.
- ❖ La manipulación directa: Interacción directa con los objetos de la pantalla, rápida e intuitiva y fácil de aprender.
- ❖ Pocas opciones en cada menú.

2.6.2- Modelo conceptual de la base de datos.

En general los modelos conceptuales por su nivel de abstracción y riqueza semántica constituyen una interfaz útil entre el informático y los usuarios finales en las primeras etapas del proceso de diseño de bases de datos, ya que reflejan en mayor medida la semántica, el significado de los datos y sus interrelaciones.[27]

Capítulo II: Descripción del Sistema.

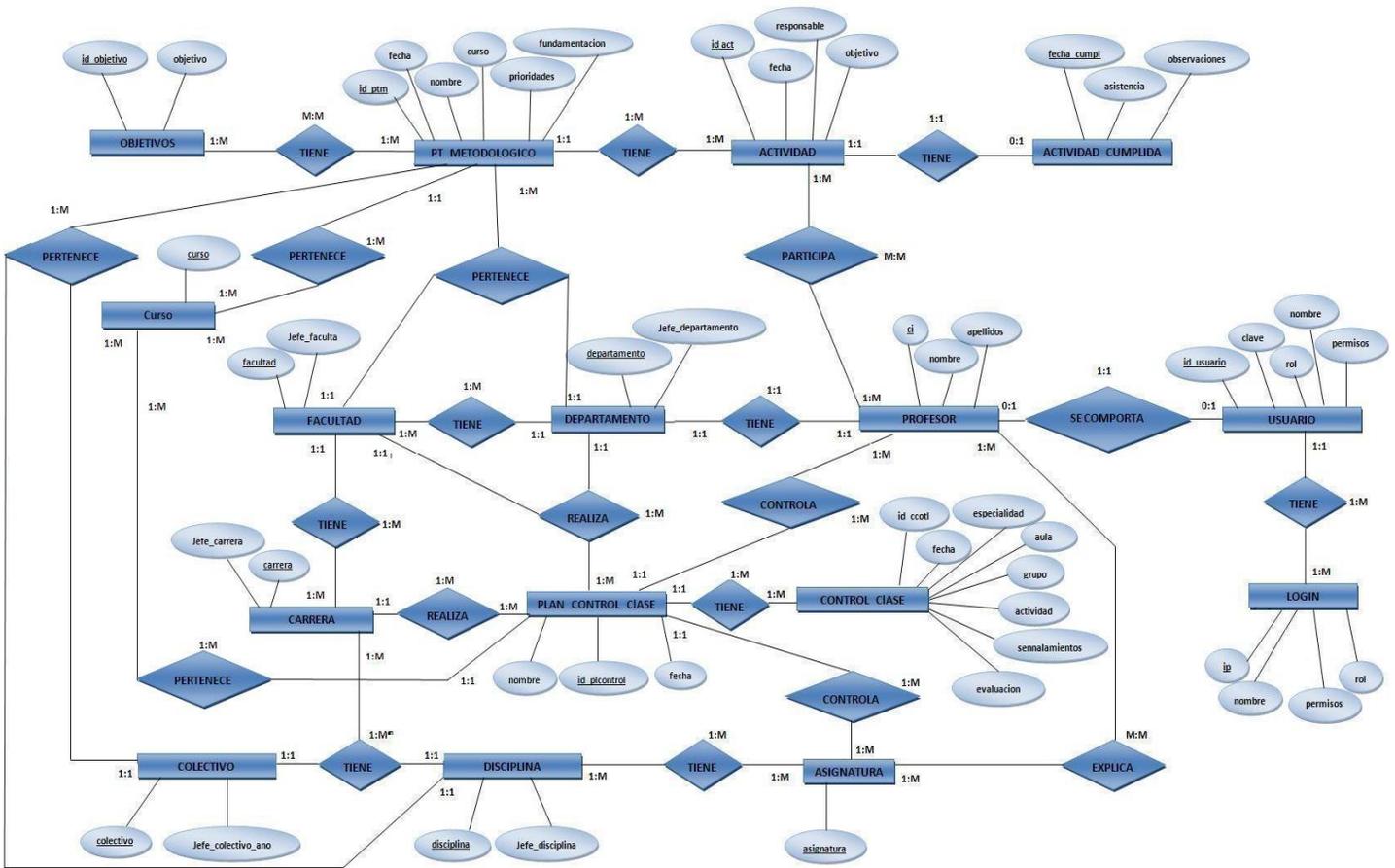


Ilustración 7: Modelo conceptual de la base de datos.

2.6.3- Modelo lógico de la base de datos.

El diagrama del modelo lógico de datos o diagrama de clases persistentes, muestra las clases capaces de mantener su valor en el espacio y en el tiempo.[27]

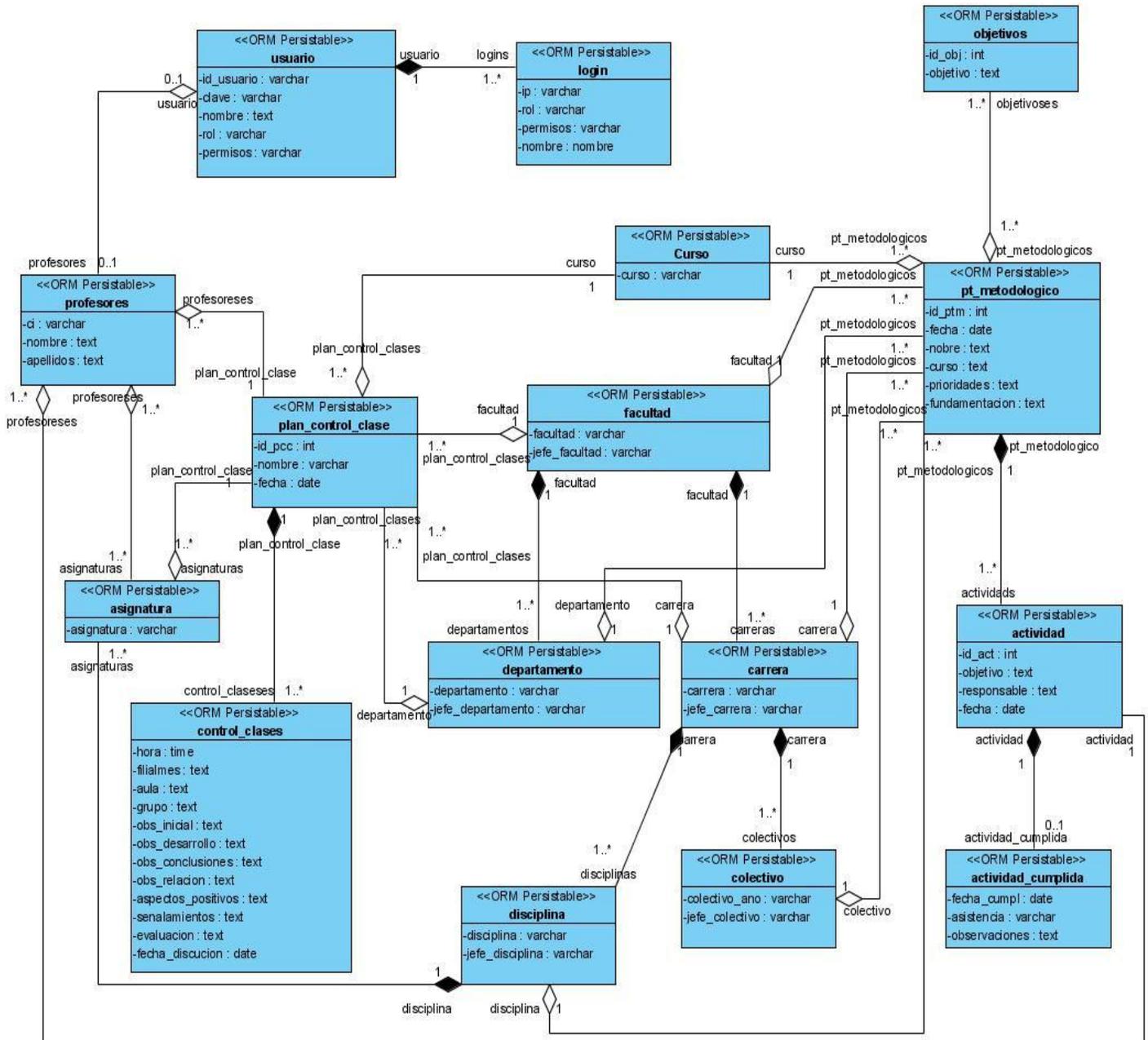


Ilustración 8 : Modelo lógico de la base de datos.

Capítulo II: Descripción del Sistema.

2.6.4- Modelo físico de la base de datos.

El modelo físico de datos incluye todos los aspectos de diseño de un modelo de base de datos, que se pueden modificar sin cambiar los componentes de la aplicación.[27]

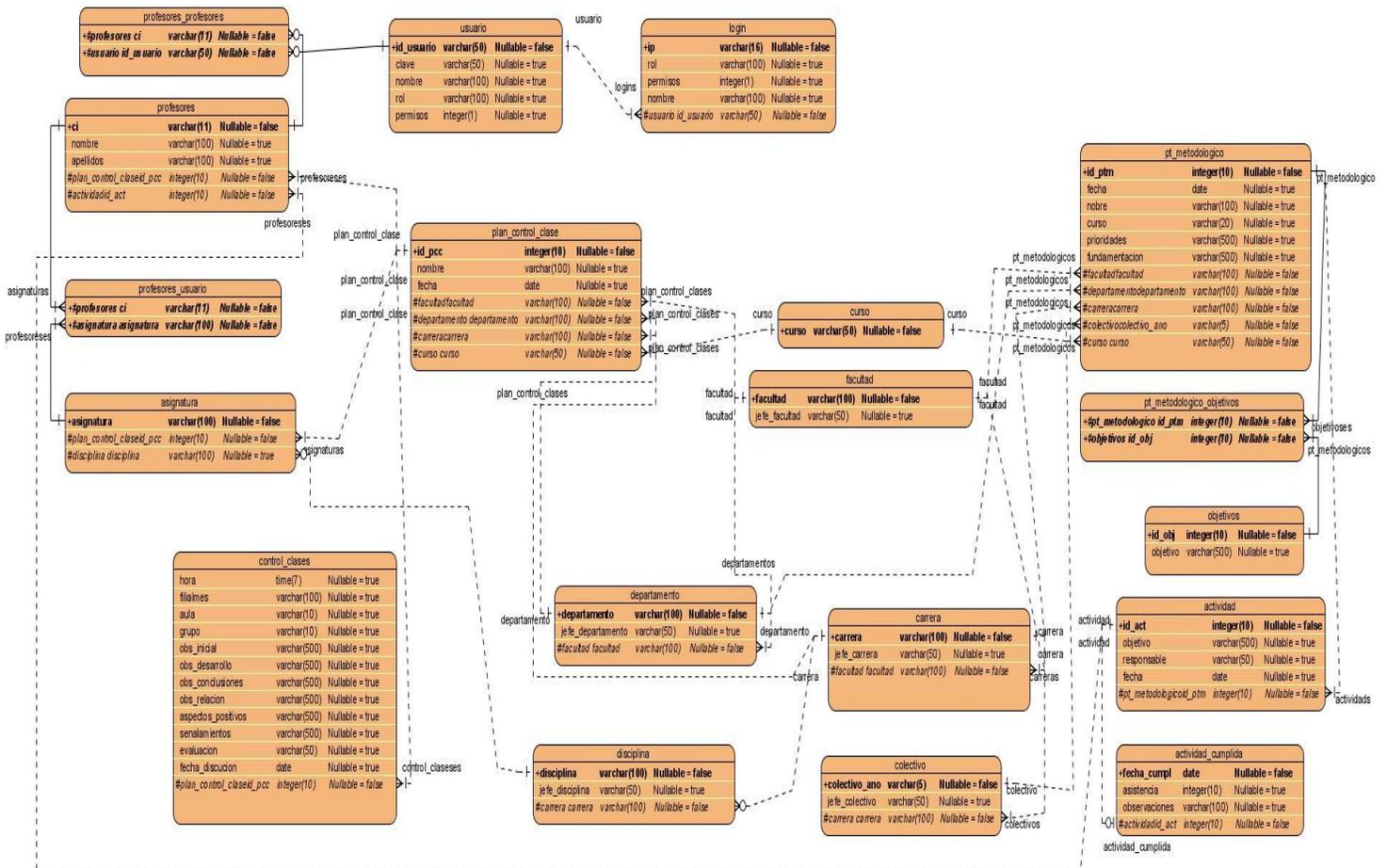


Ilustración 9: Modelo físico de la base de datos.

2.7- Conclusiones.

En este capítulo se crearon los artefactos correspondientes a las fases que propone la metodología FDD utilizando notación UML. Se analizó y describió el flujo de los procesos involucrados en la gestión de la información del trabajo metodológico que se genera en la Universidad de Cienfuegos (UCF). Se especificaron los requerimientos que debe tener el sistema, se desarrolló la planeación y como parte del diseño por funcionalidades, se elaboró el modelo conceptual, lógico y físico de la base de datos.

Asimismo se realizó un estudio de factibilidad del sistema el cual arrojó un costo estimado de \$4900.00 con un tiempo de desarrollo por dos personas de aproximadamente seis meses. Analizando el costo y los beneficios de la implantación del sistema y las mejoras que introduce en dichos procesos se concluyó: que el desarrollo del proyecto es factible

Capítulo III: Construcción del Sistema.

3.1- Introducción

En este capítulo se realiza una descripción del diseño del sistema utilizando los diagramas de clases. Se aborda todo lo relacionado con la seguridad del sistema y el tratamiento de errores. Adicionalmente se realiza la validación de la solución propuesta.

3.2- Descripción de la arquitectura propuesta

CodeIgniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación.

El Modelo representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos. Se implementó una clase del modelo para cada tabla de la base de datos.[18]

La Vista es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una Vista también puede ser un fragmento de una página con un encabezado o un pie de página. Las vistas de la aplicación poseen elementos comunes que fueron definidos en la plantilla además de las vistas diseñadas para cada una de las funcionalidades.[18]

El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web. Se definieron doce clases controladoras, de manera tal que cada una gestiona la información relacionada con una clase entidad, además una para la administración y la principal que es la que se configura para ser cargada por la vista principal del sistema.[18]

Capítulo III: Construcción del Sistema.

Teniendo en cuenta que la aplicación se implementó utilizando como base CodeIgniter desde el punto de vista técnico y arquitectónico posee las siguientes características:

- ❖ Bajo Acoplamiento: Acoplamiento es el grado que los componentes de un sistema dependen entre ellos. Mientras menos componentes dependan de otro, más reusable y flexible el sistema se vuelve. [18]
- ❖ Singularidad del Componente: Singularidad es el grado que más componentes tienen un propósito en el que enfocarse más estrecho. Cada clase y sus funciones son altamente autónomas para permitir máxima utilidad.[18]

De manera general el sistema es poco acoplado con gran singularidad de componentes. Posee simplicidad, flexibilidad y buen rendimiento.

3.3- Diagrama de clases

Un diagrama de clases es una colección de elementos declaratorios del modelo, como clases, tipos y sus relaciones; conectados unos a otros y a sus contenidos en forma de grafo. Se usa como medio para definir las páginas y sus hipervínculos.

Se modelaron los diagramas de clases que se presentan a continuación:

Funcionalidades	Diagrama de clases Web
Iniciar sesión de usuario.	Anexo A1
Cerrar sesión de usuario.	Anexo A2
Consultar ayuda.	Anexo A3
Cambiar contraseña.	Anexo A4
Gestionar facultad.	Anexo A5
Gestionar departamento.	Anexo A6
Gestionar carrera.	Anexo A7
Gestionar disciplina.	Anexo A8
Gestionar colectivo.	Anexo A9
Gestionar asignaturas de una disciplina.	Anexo A10

Capítulo III: Construcción del Sistema.

Gestionar profesor de un departamento.	Anexo A11
Gestionar plan de trabajo a metodológico.	Anexo A12
Gestionar objetivos del plan de trabajo metodológico.	Anexo A13
Gestionar actividades metodológicas.	Anexo A14
Gestionar actividades metodológicas cumplidas.	Anexo A15
Gestionar plan de control a clases	Anexo A16
Gestionar control a clases.	Anexo A17
Ver informe de las actividades metodológicas.	Anexo A18
Buscar claustro de profesores de una carrera	Anexo A19
Buscar profesor.	Anexo A20
Buscar plan de trabajo metodológico.	Anexo A21
Buscar plan de trabajo metodológico de cursos anteriores.	Anexo A22
Ver actividades metodológicas no cumplidas.	Anexo A23

Tabla 18: Diagramas de clases

3.4- Estándar de implementación

Para el adecuado mantenimiento del código del sistema es necesario establecer un estándar de codificación a emplear. Las variables, nombres de funciones, clases y objetos fueron declarados en las páginas, estos son cortos, claros, y describen su propósito. Los objetos se nombran según el valor de su contenido. Los inicios ({) y cierre (}) de ámbito se encuentran alineados debajo de la declaración a la que pertenecen. Los signos lógicos y de operación se separan por un espacio antes y después de los mismos. Se emplean comentarios para añadir información sobre determinadas líneas de código en las páginas y así ayudar a entender el objetivo del mismo. Se emplean además nombres similares entre clases de manera tal que se establezca claramente la relación entre la clase del modelo, la clase controladora que maneja sus datos y la tabla de la base de datos de donde obtienen la información necesaria.

3.5- Seguridad del sistema

- ❖ El mecanismo de seguridad y protección de la aplicación se basa en el empleo del nombre de usuario, contraseña y rol para acceder a las funcionalidades a las que tienen acceso.

Capítulo III: Construcción del Sistema.

- ❖ Cada usuario tiene definido a que módulo e información puede acceder, teniendo en cuenta la política de seguridad del nivel organizativo en que se encuentre.
- ❖ Los usuarios una vez registrados en su módulo no podrán acceder haciendo uso de la url a otra página a la cual no tiene acceso.
- ❖ Los usuarios una vez que se desconectan de su módulo no podrán regresar y realizar acciones dentro del mismo sin antes autenticarse.
- ❖ Se utilizaron mecanismos de encriptación de los datos usando para ello la función hash MD5.

3.6- Tratamiento de errores

A menudo las personas que usan sistemas informáticos cometen errores que pueden afectar el adecuado funcionamiento de los mismos. Por lo que es necesario siempre realizar un adecuado tratamiento para estas situaciones excepcionales que pueden suceder. Como por ejemplo la entrada de datos incorrectos o en un formato no soportado por el sistema.

El sistema está diseñado e implementado de forma tal, que las posibilidades de introducir información errónea sean mínimas. Aunque en algunos casos es necesario teclear datos y seleccionar elementos, se mantiene un nivel elevado de validación. Se muestran comentarios sobre el formato de la información a introducir en las áreas de textos de las pantallas. Sobre los botones se colocan etiquetas informativas para brindarle información al usuario sobre cada funcionalidad. Adicionalmente a la validación los mensajes de error que muestra el sistema se encuentran escritos en un lenguaje de fácil comprensión para los usuarios.

3.7- Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo

Capítulo III: Construcción del Sistema.

representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.[28]

El siguiente diagrama muestra la configuración hardware del sistema y los nodos físicos que lo componen. El sistema estará estructurado según la arquitectura cliente - servidor. En el lado del servidor estarán en funcionamiento, el servidor de bases de datos MySQL y el servidor Web Apache. Esta se comunicará con el cliente de la Intranet a través del protocolo HTTP. El cliente podrá visualizar la aplicación con el navegador que desee puede ser Mozilla Firefox o Internet Explorer .Además se cuenta con una impresora conectada a la máquina cliente que le brindará la posibilidad al usuario de imprimir la información que desee.

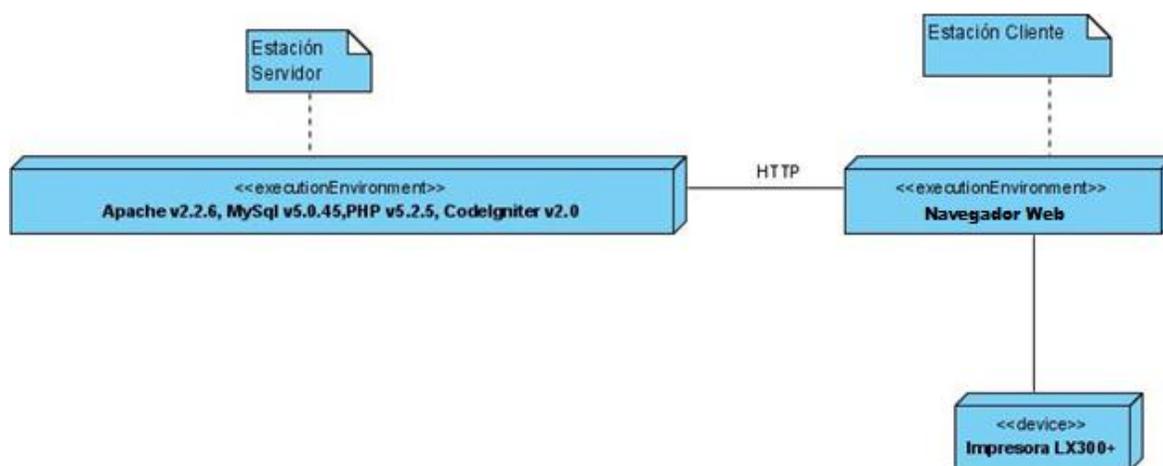


Ilustración 10: Diagrama de despliegue del sistema.

3.8- Validación de la solución propuesta

3.8.1- Prueba T para muestras pareadas.

Una prueba T de Student, o Test-T es cualquier prueba en la que el estadístico utilizado tiene una distribución T de Student si la hipótesis nula es cierta. Se aplica cuando la población estudiada sigue una distribución normal pero el tamaño muestral es demasiado pequeño como para que el estadístico en el que está basada la inferencia esté normalmente distribuido, utilizándose una estimación de la desviación típica en lugar del valor real [29].

Capítulo III: Construcción del Sistema.

Para realizar la validación del sistema informático propuesto se tomaron 15 observaciones del tiempo en minutos en que demoraba el proceso de planificación del trabajo metodológico antes y después del sistema informático.

Tiempos antes y después de la realización de los procesos		
	Procesos	
	Antes	Después
Muestra	110	26
	90	21
	92	22
	94	21
	89	24
	93	25
	97	21
	96	20
	91	24
	90	24
	98	19
	100	23
	91	23
	94	24
	97	23
Media	94.8	22.6

Tabla 19: Tiempos antes y después de la realización de los procesos.

Se observó que el procesamiento antes del software demoraba como promedio 95 minutos y después solo 23 minutos, por lo que a simple vista se aprecia la existencia de diferencias significativas entre ellos. Para comprobarlo estadísticamente se realizó la Prueba T para comparar las medias antes y después del sistema.

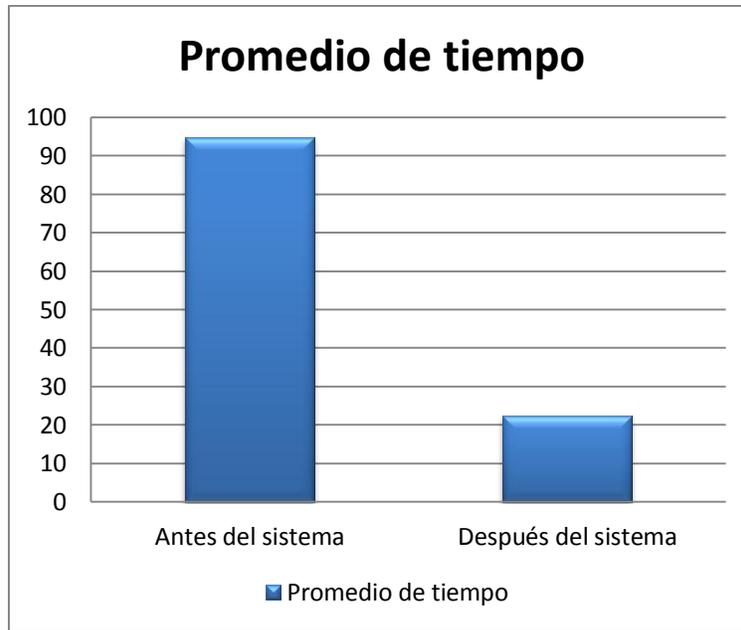


Ilustración 11: Comparación del tiempo promedio de los procesos antes y después del sistema.

Primeramente se comprobó que ambas variables (X- Tiempo antes del sistema y Y-Tiempo después del sistema.) seguían una distribución normal mediante la Prueba Kolmogorov-Smirnov.

Dicha prueba contrasta la hipótesis nula que plantea que la variable sigue una distribución normal contra la hipótesis alternativa en que se considera que la variable no sigue una distribución normal. Tomando como referencia un nivel de significación del 5 %, si este es mayor que la significación asintótica, entonces rechazamos H_0 , de lo contrario aceptamos. Utilizando un nivel de significación de 0,05 al comparar con la significación asintótica de los estadísticos calculados (0,951 y 0,342) puede concluirse que no se rechaza la hipótesis nula, demostrando que ambas variables siguen una distribución normal, por tanto al cumplirse este supuesto puede realizarse la Prueba T.

Capítulo III: Construcción del Sistema.

Prueba de Kolmogorov-Smirnov para una muestra

		antes	despues
N		15	15
Parámetros normales ^{a,b}	Media	94,8500	22,6549
	Desviación típica	3,20321	2,90111
Diferencias más extremas	Absoluta	,134	,249
	Positiva	,134	,159
	Negativa	-,099	-,242
Z de Kolmogorov-Smirnov		,518	,938
Sig. asintót. (bilateral)		,951	,342

a. La distribución de contraste es la Normal.

b. Se han calculado a partir de los datos.

Ilustración 12: Prueba de Kolmogorov-Smirnov para una muestra

La Prueba T para muestras relacionadas plantea como hipótesis nula que la media de X es igual que la media de Y, considerando que no hay diferencias significativas entre ellas y la hipótesis alternativa plantea que la media de X es diferente a la media Y, es decir ,que existen diferencias significativas entre ambas variables. Utilizando un nivel de significación de 0,05 al comparar con la significación del estadístico calculado (0,00) puede concluirse que se rechaza la hipótesis nula, y se acepta la alternativa, demostrando que hay diferencias significativas entre ambas medias.

Paired Samples Test

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 antes - despues	71,20000	4,53872	1,17189	68,68654	73,71346	60,756	14	,000

Ilustración 13: Resultado de la Prueba T

Ante estas ventajas se puede plantear que el sistema informático es rápido, confiable y maneja de forma segura toda la información.

3.9- Conclusiones

En este capítulo se abordaron temas como la descripción de la arquitectura propuesta, diagrama de clases, estándar de implementación, seguridad del sistema, tratamiento de errores y se elaboró el diagrama de despliegue. Adicionalmente se realizó la validación del sistema mostró resultados favorables a partir de la prueba de hipótesis donde el ahorro de tiempo y la confiabilidad son los mejores beneficios que aporta el sistema.

Conclusiones

Con la terminación de la presente investigación:

- ❖ Se realizó un estudio sobre las diferentes metodologías, tecnologías y herramientas más utilizadas en la actualidad y que se adecuan a las características del sistema informático.
- ❖ Asimismo, se realizó un estudio de los principales conceptos asociados al dominio del problema. En este estudio se consultaron los sistemas existentes a nivel nacional, vinculados a al trabajo metodológico, mostrando sus principales características y deficiencias.
- ❖ Se logró desarrollar un Sistema Informático para la Gestión del Trabajo Metodológico en la Universidad de Cienfuegos; dando cumplimiento de este modo al objetivo general del trabajo, lo cual fue validado mediante una prueba T para muestras pareadas.
- ❖ Se realizó el estudio de factibilidad del sistema informático, el cual refleja los beneficios tangibles e intangibles.

Recomendaciones

A pesar de haberse cumplido los objetivos específicos trazados para la realización del trabajo de diploma, esta propuesta es la primera etapa de un proyecto más amplio.

Se recomienda como pasos que den continuidad:

- ❖ Poner a prueba todas las funcionalidades del sistema propuesto durante un período prolongado de tiempo, permitiendo esto comprobar dichas funcionalidades de forma práctica y poder entonces detectar posibles mejoras a realizar en el futuro.
- ❖ Extender el uso de este sistema a otras Universidades del país.
- ❖ Evaluar la posibilidad de integrar los resultados de este trabajo al Sistema de Gestión de la Nueva Universidad (SIGENU).

Referencias Bibliográficas

- [1] Dr. Pedro Horruitiner Silva, «El trabajo metodológico una concepción desde la Vicerectoría Académica», 2001.
- [2] Ideleichy Lombillo, «Las TIC en la universidad cubana.» 2009. [Online] Available: <http://www.monografias.com/trabajos91/tic-modelo-semipresencial-universidad-cubana/tic-modelo-semipresencial-universidad-cubana.shtml>
- [3] Lic. Denis Popa Garcés, «El trabajo metodológico en la universidad cubana.» 2007. [Online] Available: <http://www.monografias.com/trabajos75/trabajo-metodologico-nueva-universidad-cubana/trabajo-metodologico-nueva-universidad-cubana.shtml>.
- [4] P. Bartle., «Información para la gestión de la información». 08-jun-2010. [Online] Available: <http://www.scn.org/mpfc/modules/mon-miss.htm>.
- [5] I. R. L. P. S. MSc. Lic. Lucía Sotomayor Guevara y Ms. I. M. Á. T. A. Ing. Vivian Aymeé Milanés Clavijo., «Sistema informático para la gestión de la planificación Docente.», dic. 2010.
- [6] Alberto Molpeceres, «Procesos de desarrollo: RUP, XP y FDD.», mar. 2010.
- [7] M. Pérez, «Introducción a UML.» 24-mar-2009. [Online] Available: <http://www.programacion.com/tutorial/uml/>.
- [8] A. Teruel, «Arquitectura de Capas.» 24-mar-2010. [Online] Available: <http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>.
- [9] Fernando «seacat» Velo, «CodeNiter Guia de usuario», dic. 2011.
- [10] «Internet Explorer». Enr-2013. [Online]. Available: <http://en.wikipedia.org/wiki/ieexplorer>.
- [11] «CSS: Hojas de estilo». 24-abr-2010. [Online]. Available: <http://es.kioskea.net/contents/css/cssintro.php3>.
- [12] José Antonio Rodríguez, «Manual de JavaScript». 2009-2012.
- [13] «Servidor HTTP Apache». 22-may-2013. [Online] Available: http://es.wikipedia.org/wiki/Servidor_HTTP_Apache.
- [14] «Tutorial PHP». 10-feb-2011. [Online] Available: <http://www.webestilo.com/php/php00.phtml>.
- [15] Javier J. Gutiérrez., « ¿Que es un Framework?», mar. 2011. .
- [16] «Symphony, un CMS open source flexible». 2010. [Online] Available: <http://www.dacostabalboa.com/es/symphony-un-cms-open-source-muy->

Referencias Bibliográficas

flexible/6060.

[17] «Tutorial de Cake PHP». 22-dic-2011. [Online] Available: <http://auladigital.com/tutorial/tutorial-de-cake.php>.

[18] «CodeIgniter». 13-feb-2008. [Online] Available: <http://en.wikipedia.org/wiki/codeiniter>.

[19] «Adobe Dreamweaver». 07-nov-2011. [Online] Available: <http://www.cavsi.com/preguntasrespuestas/que-es-adobe-dreamweaver/>.

[20] «Adobe Photoshop». 11-feb-2013. [Online] Available: http://en.wikipedia.org/wiki/adobe_photoshop.

[21] «PHPMyAdmin». 20-mar-2011. [Online] Available: <http://www.phpmyadmin.net/>.

[22] «Visual Paradigm for UML». 15-jun-2008. [Online] Available: <http://www.freedownloadmanager.org/es/>.

[23] «Manual de MySQL.» 26-feb-2010. [Online] Available: <http://www.mysql.com/download>.

[24] «Características de PostgreSQL». 24-oct-2011. [Online] Available: www.manualesdeayuda.com/manuales/basesdedatos/postgresql/caracteristicas-de-postgresql-01844.html.

[25] G. B. I. J. J. Rumbaugh, «El Proceso Unificado de Desarrollo de Software». 2002.

[26] M. Peralta, «Estimación del esfuerzo Basada en casos de uso.». 2006.

[27] Laura Toledo, «Conferencia 2 Diseño Conceptual de Bases de Datos». 2006.

[28] I. thomson., «Desarrollo web con PHP y MySQL». 2011.

[29] «Prueba T de Student», 2008. [Online] Available: http://es.wikipedia.org/wiki/t_student.

Bibliografía

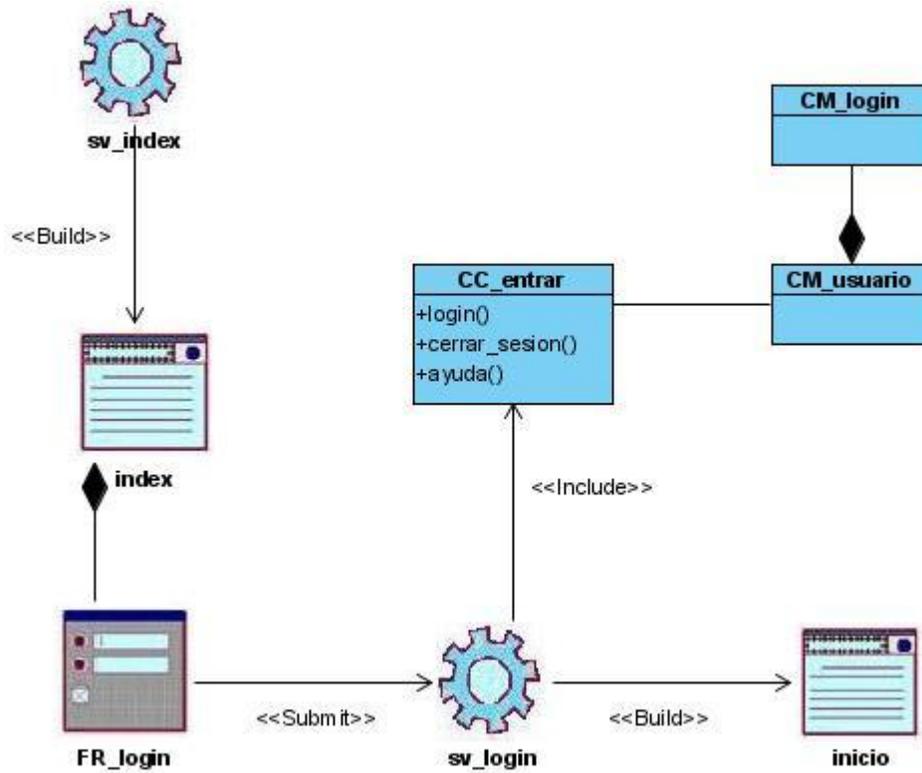
- [1] G. B. I. J. J. Rumbaugh, «El Proceso Unificado de Desarrollo de Software». 03-may-2006.
- [2] I. R. L. P. S. MSc. Lic. Lucía Sotomayor Guevara y Ms. I. M. Á. T. A. Ing. Vivian Aymeé Milanés Clavijo., «Sistema informático para la gestión de la planificación Docente.», dic. 2010.
- [3] Dr. Pedro Horruitiner Silva, «El trabajo metodológico una una concepcion desde la Vicerectoria Academical», 2001.
- [4] Laura Toledo, «Conferencia 2 Diseño Conceptual de Bases de Datos». 2006.
- [5] Lic. Denis Popa Garcés, «El trabajo metodológico en la universidad cubana.» 2007. [Online]. Available: <http://www.monografias.com/trabajos75/trabajo-metodologico-nueva-universidad-cubana/trabajo-metodologico-nueva-universidad-cubana.shtml>
- [6] P. Craig Larman, «Una introducción al análisis y diseño orientado a objetos y al proceso unificado», 2008.
- [7] «Prueba T de Student». 2008. [Online]. Available: http://es.wikipedia.org/wiki/t_student.
- [8] «CodeIgniter». 13-feb-2008. [Online]. Available: <http://en.wikipedia.org/wiki/codeiniter>.
- [9] «Visual Paradigm for UML». 15-jun-2008. [Online]. Available: <http://www.freedownloadmanager.org/es/>.
- [10] Ideleichy Lombillo, «Las TIC en la universidad cubana.» 2009. [Online]. Available: <http://www.monografias.com/trabajos91/tic-modelo-semipresencial-universidad-cubana/tic-modelo-semipresencial-universidad-cubana.shtml>.

- [11] Jose Antonio Rodríguez, «Manual de JavaScrip». 2012-2009.
- [12] M. Pérez, «Introducción a UML.» 24-mar-2009. [Online]. Available: <http://www.programacion.com/tutorial/uml/>.
- [13]«Symphony, un CMS open source flexible». 2010. [Online]. Available: <http://www.dacostabalboa.com/es/symphony-un-cms-open-source-muy-flexible/6060>.
- [14]«Manual de MySQL.» 26-feb-2010. [Online]. Available: <http://www.mysql.com/download>.
- [15] Alberto Molpeceres, «Procesos de desarrollo: RUP,XP y FDD.», mar. 2010.
- [16]«CSS: Hojas de estilo». 24-abr-2010. [Online]. Available: <http://es.kioskea.net/contents/css/cssintro.php3>.
- [17] P. Bartle., «Información para la gestión de la información». 08-jun-2010. [Online]. Available: <http://www.scn.org/mpfc/modules/mon-miss.htm>.
- [18] I. thomson., «Desarrollo web con PHP y MySQL». 2011.
- [19]«Tutorial PHP». 10-feb-2011. [Online]. Available: <http://www.webestilo.com/php/php00.phtml>.
- [20]«PHPMyAdmin». 20-mar-2011. [Online]. Available: <http://www.phpmyadmin.net/>.
- [21] A. Teruel, «Arquitectura de N Capas.» 24-mar-2011. [Online]. Available: <http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>.
- [22] Javier J. Gutiérrez., «¿Que es un Framework?», mar. 2011.
- [23]«Características de PostgreSQL». 24-oct-2011. [Online]. Available: www.manualesdeayuda.com/manuales/basesdedatos/postgresql/caracteristicas-de-postgresql-01844.html.

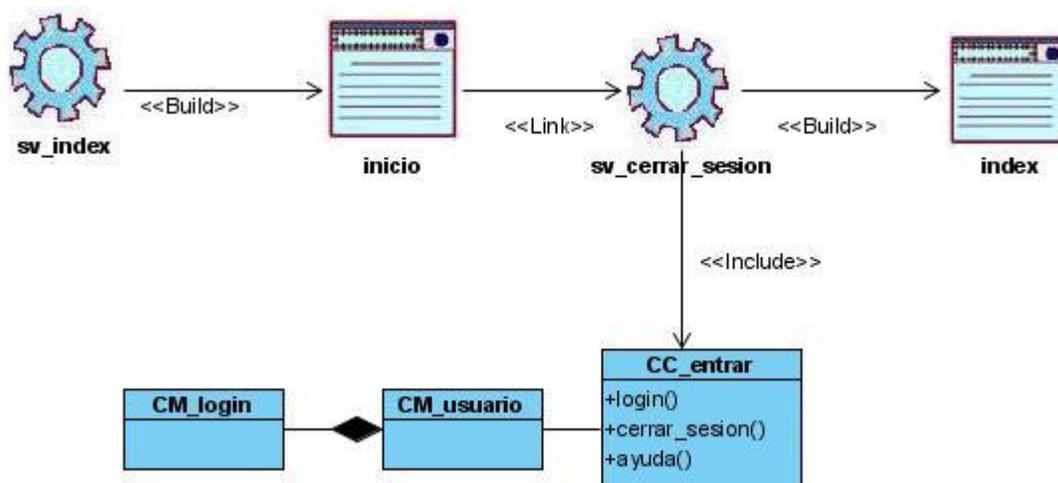
- [24] «Adobe Dreamweaver». 07-nov-2011. [Online]. Available: <http://www.cavsi.com/preguntasrespuestas/que-es-adobe-dreamweaver/>.
- [25] Fernando «seacat» Velo, «CodeNiter Guia de usuario», dic. 2011.
- [26] «Tutorial de Cake PHP». 22-dic-2011. [Online]. Available: <http://auladigital.com/tutorial/tutorial-de-cake.php>.
- [27] «Internet Explorer». enr-2013. [Online]. Available: http://en.wikipedia.org/wiki/Internet_Explorer.
- [28] «Adobe Photoshop». 11-feb-2013. [Online]. Available: http://en.wikipedia.org/wiki/Adobe_Photoshop.
- [29] «Servidor HTTP Apache». 22-may-2013. [Online]. Available: http://es.wikipedia.org/wiki/Servidor_HTTP_Apache.
- [30] M. Peralta, «Estimación del esfuerzo Basada en casos de uso.» 2008.
- [31] «(FDD) Feature Driven Development», 10-feb-2010. [Online]. Available: <http://www.featuredrivendevelopment.com/es/>

Anexos

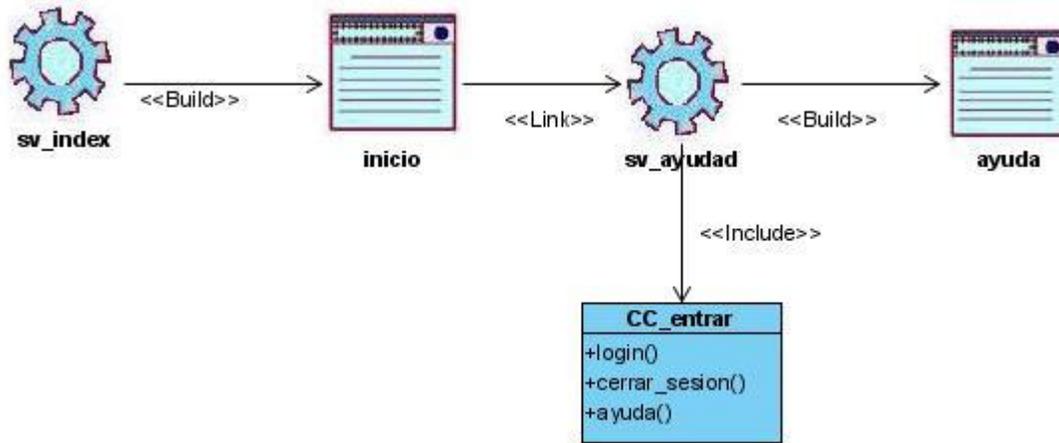
Anexo A1: Iniciar sesión de usuario.



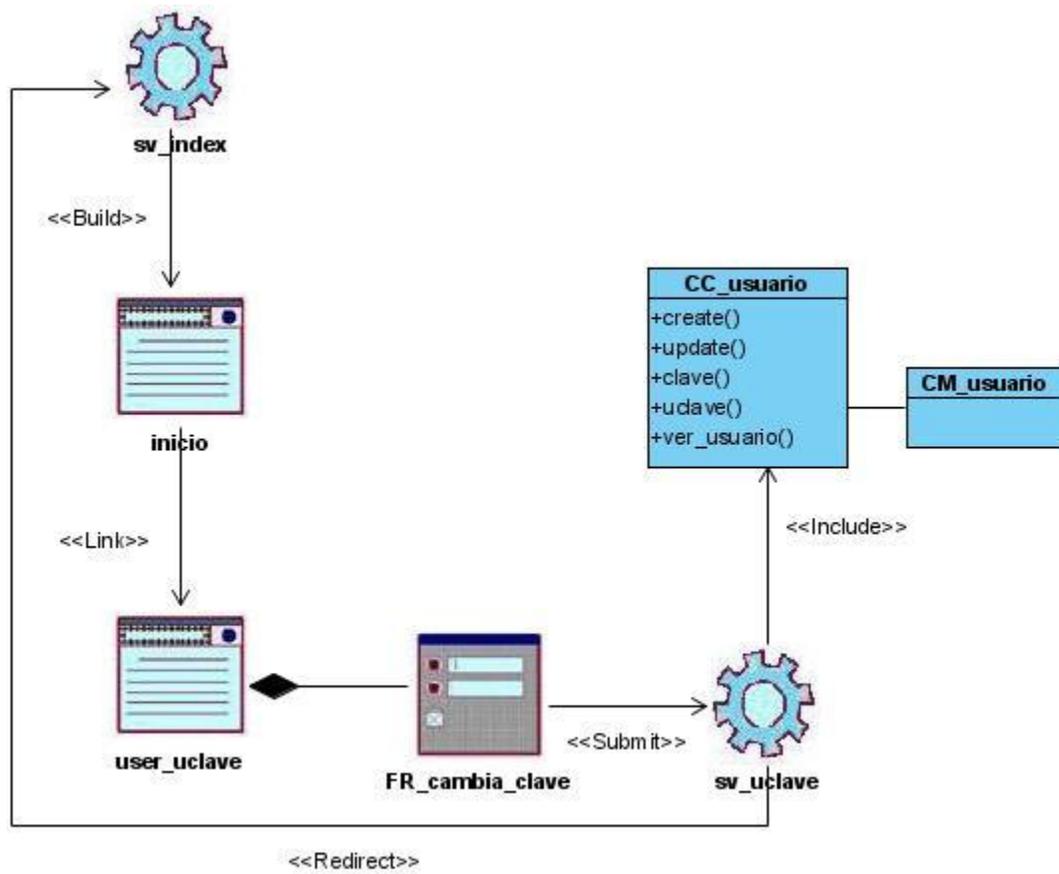
Anexo A2: Cerrar sesión de usuario.



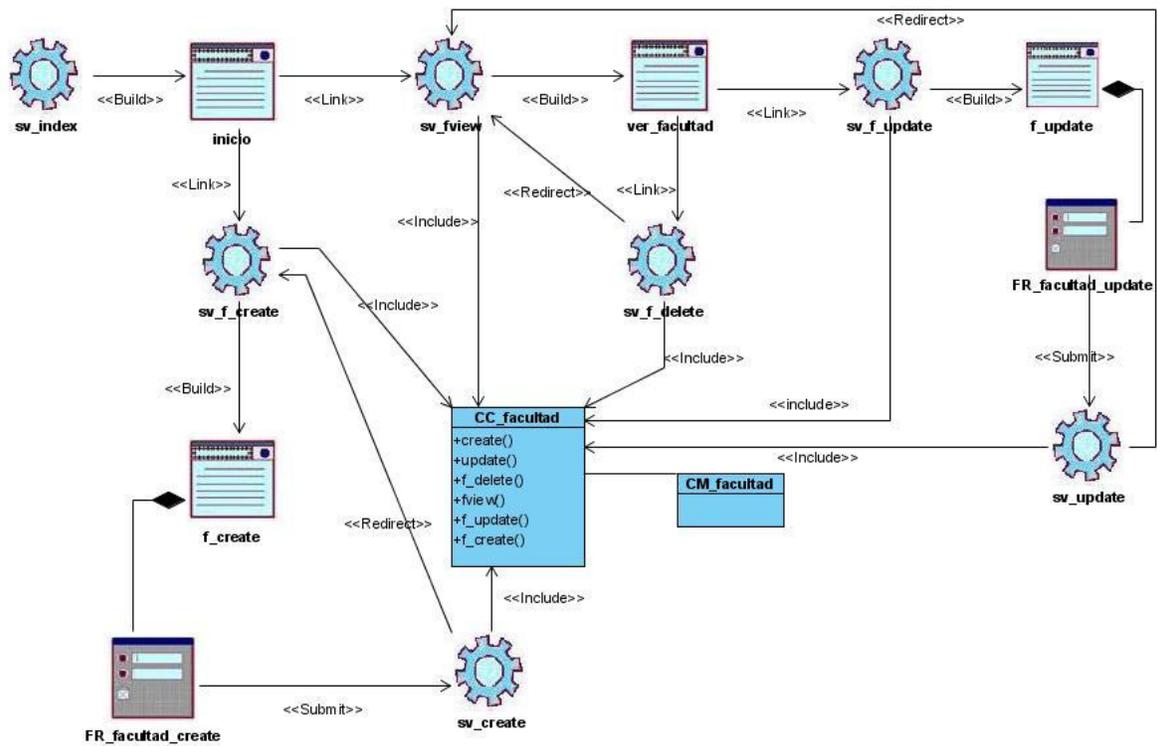
Anexo A3: Consultar ayuda.



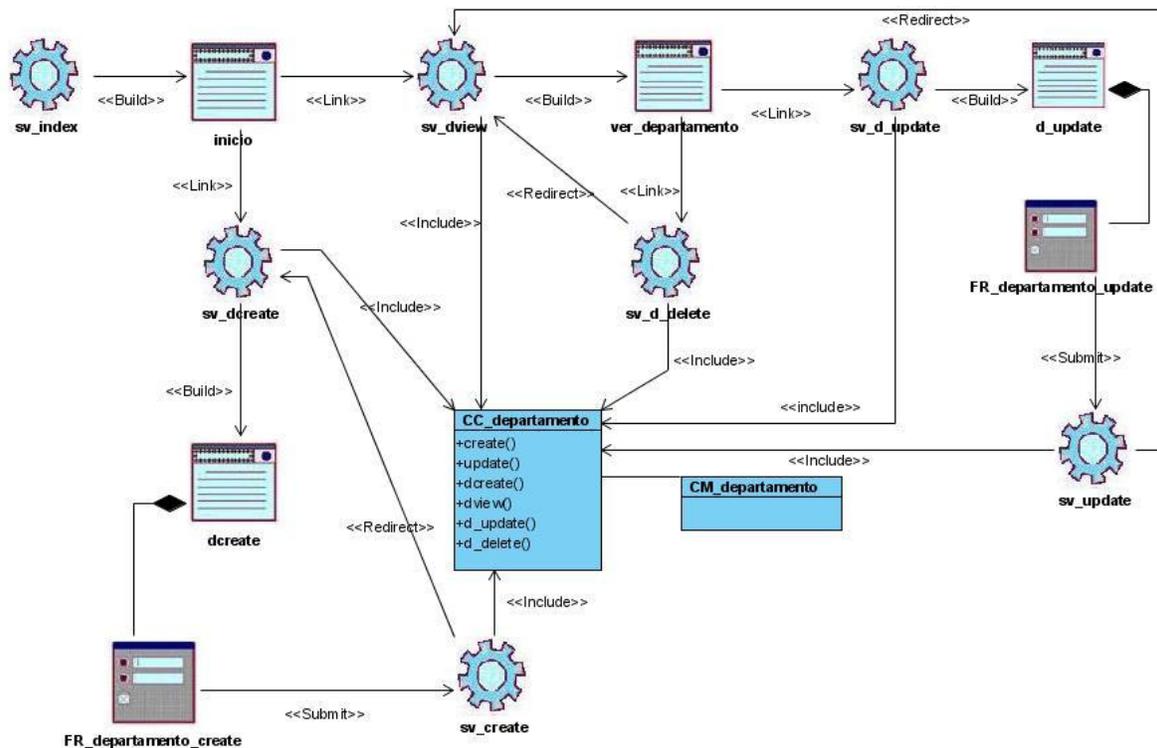
Anexo A4: Cambiar contraseña.



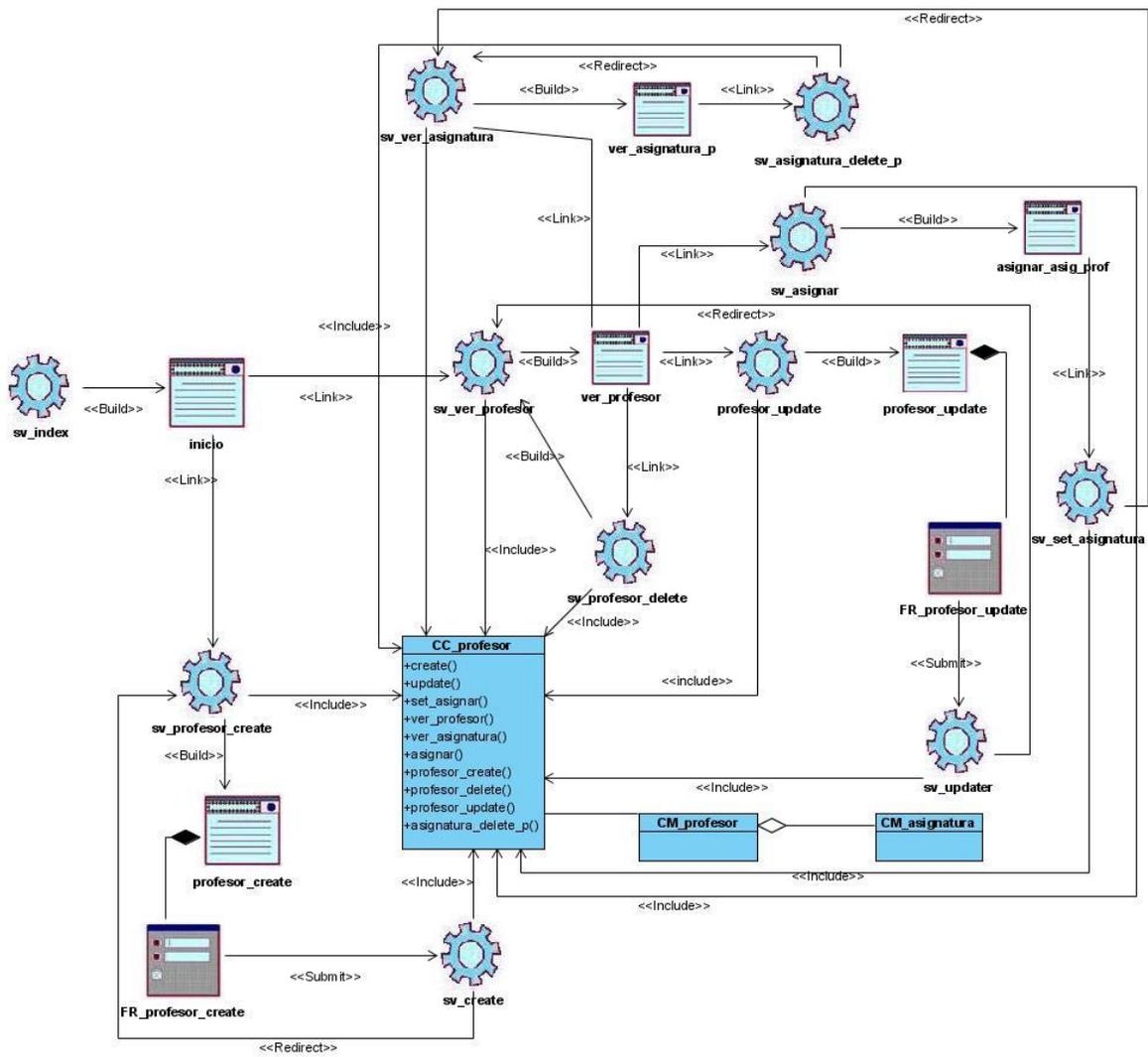
Anexo A5: Gestionar facultad.



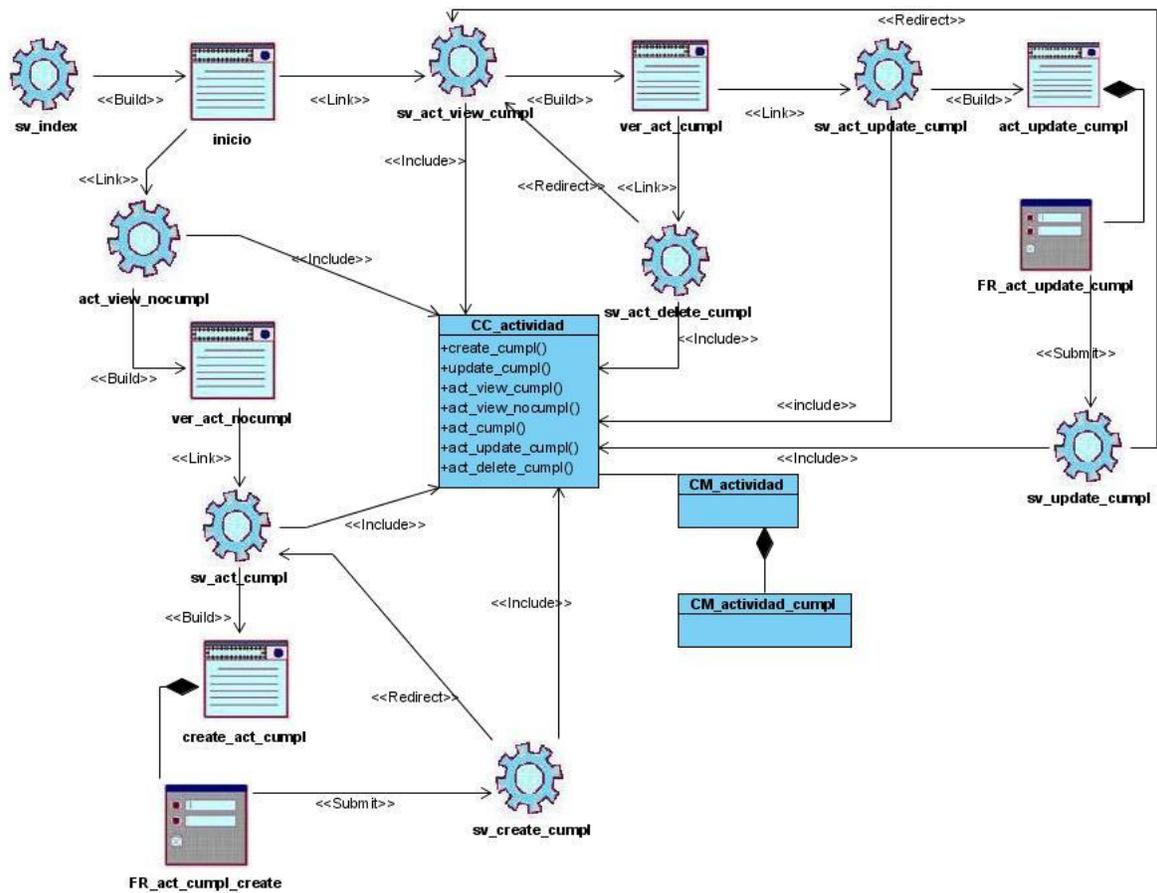
Anexo A6: Gestionar departamento.



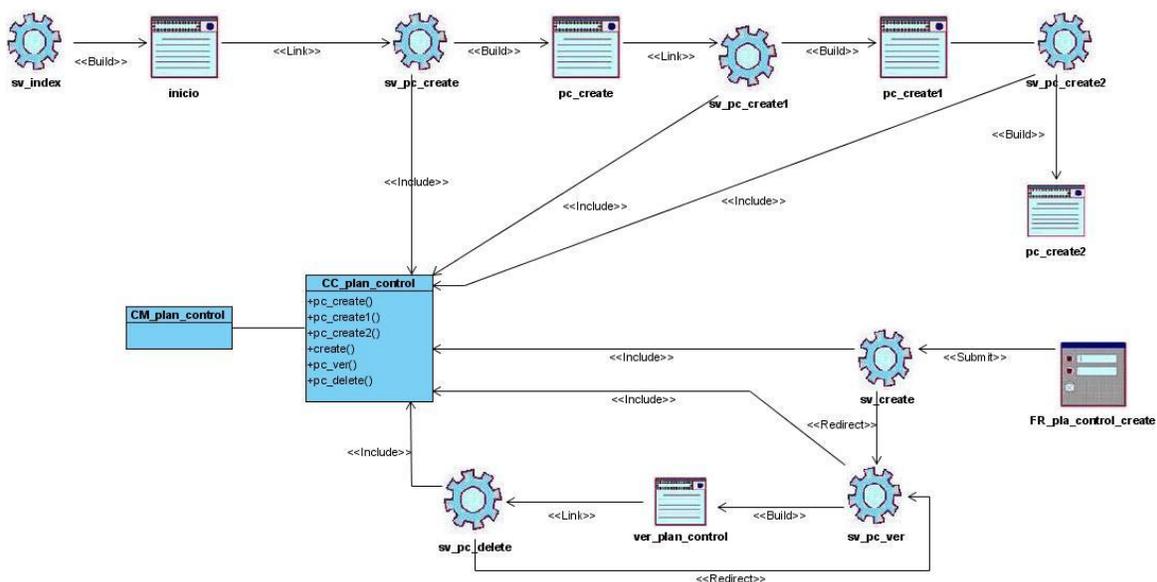
Anexo A11: Gestionar profesor de un departamento.



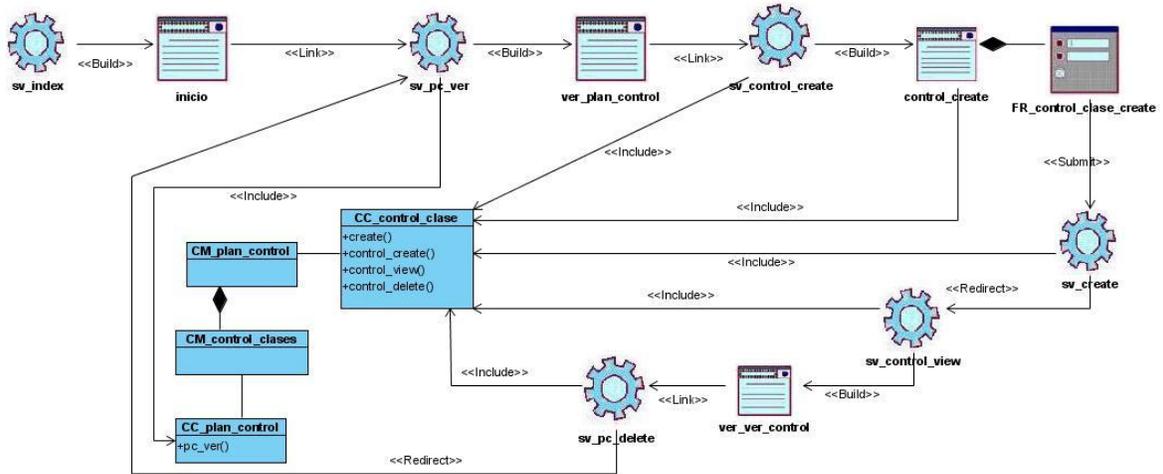
Anexo A15: Gestionar actividades metodológicas cumplidas.



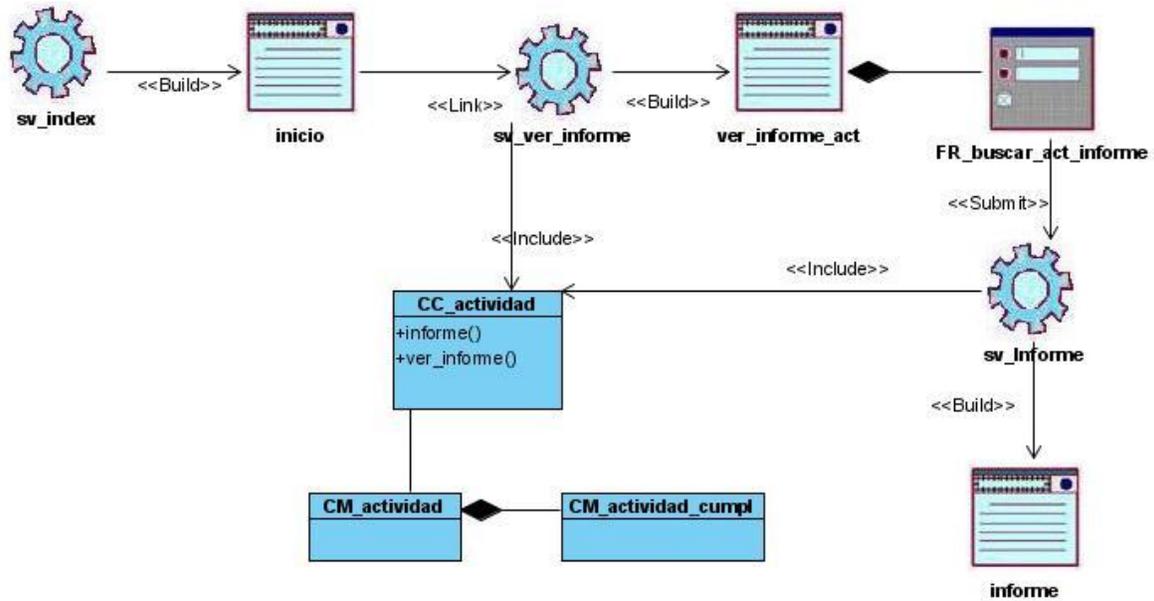
Anexo A16: Gestionar plan de control a clases.



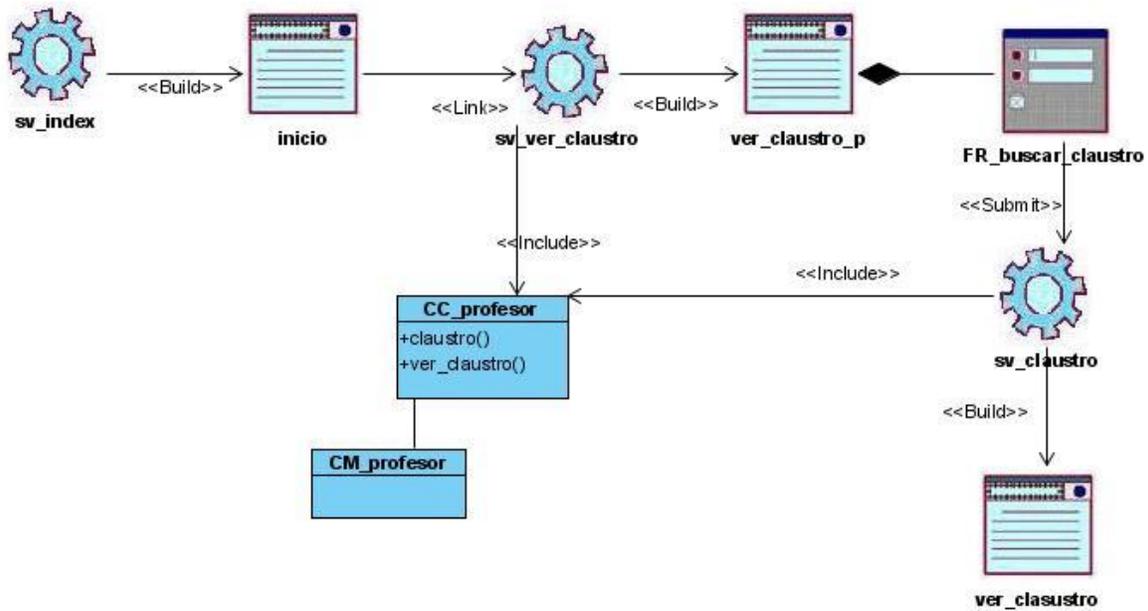
Anexo A17: Gestionar control a clases.



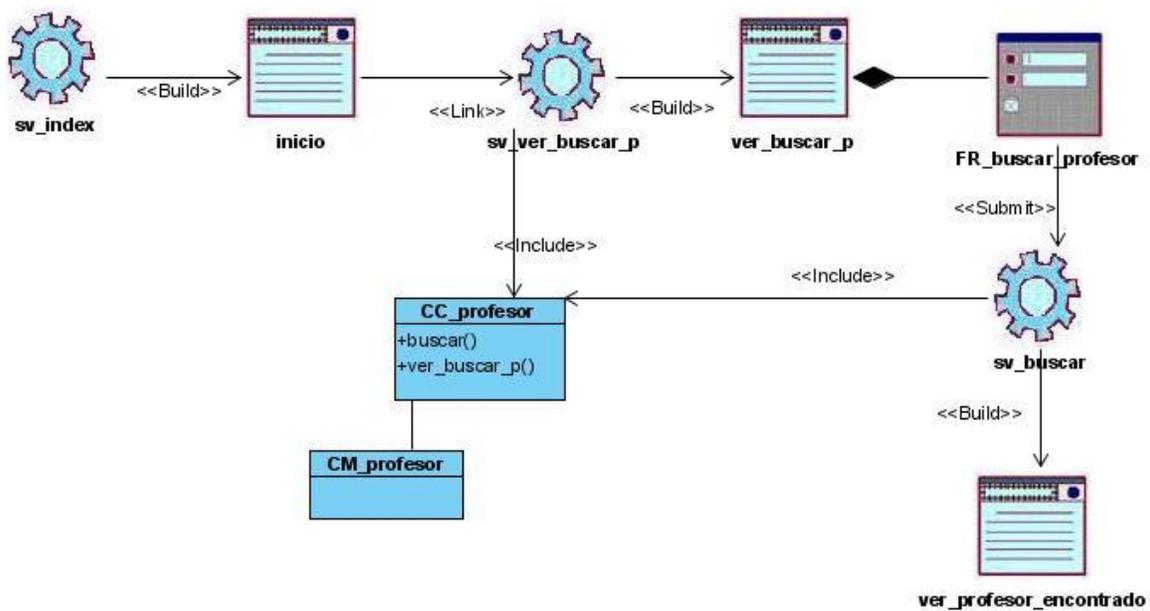
Anexo A18: Ver informe de las actividades metodológicas.



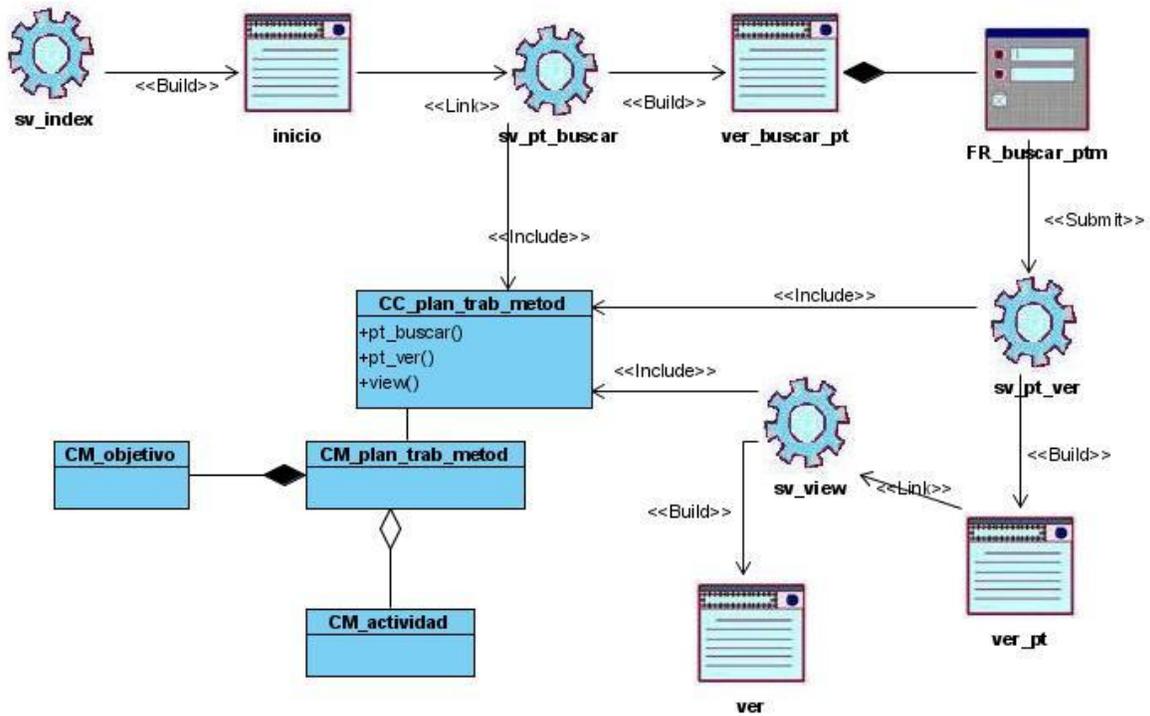
Anexo A19: Buscar claustro de profesores de una carrera.



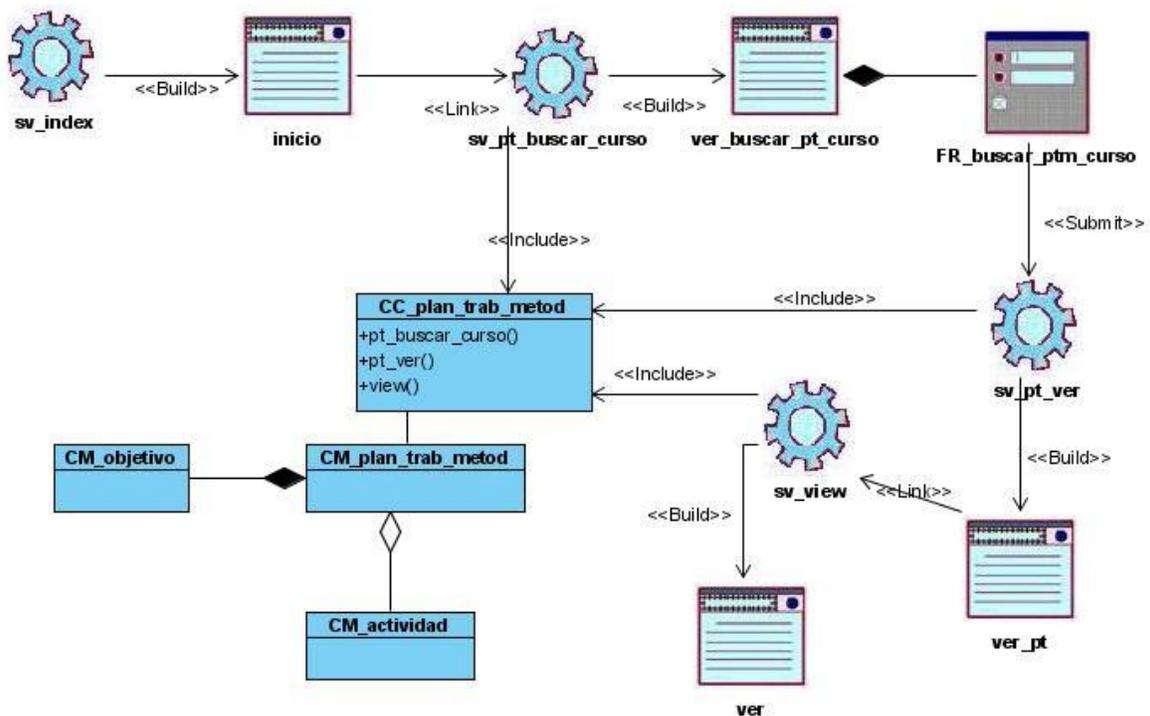
Anexo A20: Buscar profesor.



Anexo A21: Buscar plan de trabajo metodológico.



Anexo A22: Buscar plan de trabajo metodológico de cursos anteriores.



Anexo A23: Ver actividades metodológicas no cumplidas.

