

Universidad “Oscar Lucero Moya”, Holguín

Facultad de Informática y Matemática

**Trabajo de Diploma en opción al Título de Ingeniero
Informático**

**SISTEMA PARA LA GESTIÓN DE INFORMACIÓN DEL
TRABAJO METODOLÓGICO EN LA UNIVERSIDAD DE
HOLGUÍN “OSCAR LUCERO MOYA”**

Autora: Elizabeth Bárbara Infante Hidalgo

Tutores: M.Sc Jenny Ruiz de la Peña

Ing. Ángel Luis Scull Pupo



Holguín 2014

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo al Departamento de Informática de la Universidad de Holguín “Oscar Lucero Moya” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del _____.

Nombre completo del autor

Nombre completo del tutor

Nombre completo del tutor

A dos personas que siempre están en mi corazón: mis dos soles Nirma y Elvia.

A Carlitos por ser mi guía, mi amigo, mi compañero, mi amante y mi fe.

A mi familia, por su educación y todo su amor.

A mí por tener bien claro la persona que quiero ser, por no darme por vencida y luchar por mi futuro, por tener la fuerza de enfrentar las más difíciles circunstancias casi siempre sola; y sobre todo por haber creído en mí cuando más nadie lo hizo...

Agradecimientos

A mami, por su apoyo incondicional y por quererme tanto, por darme la vida y sobre todo porque lo es todo para mí.

A mis tutores por todo su tiempo y dedicación para conmigo, gracias por confiar en mí y haberme permitido realizar esta investigación que ha sido, realizar un sueño.

A mami Nirma por ayudar en mi formación, por apoyarme siempre, y por considerarme su hija, por darme su amor y tener la fuerza de ser siempre optimista, y sobre todas las cosas por ayudarme a sobreponerme a tantas dificultades.

A Carlos Daniel, gracias mi amor por estar siempre a mi lado todo este tiempo y por soportar tantos momentos de preocupación y desánimo, gracias por tanta ayuda y por poder siempre, estar ahí para mí incondicionalmente, y por escucharme, quererme, apoyarme y darme luz cuando todo se oscurece.

A mis amistades por su preocupación y su ayuda.

A mis compañeros de aula, por su apoyo en los momentos necesarios.

A los profesores de la carrera por su educación y maravillosa enseñanza durante estos cinco años.

A todas las personas que de una forma u otra depositaron en mi su confianza, y su fe para que mi deseo se cumpliera de la mejor manera.

A todos muchas gracias.

Resumen

La Universidad de Holguín “Oscar Lucero Moya” lleva a cabo cinco procesos sustantivos fundamentales que sostienen todos los resultados obtenidos en la esfera educativa por este centro. Estos procesos son: formación de profesionales, universalización, postgrado, extensión universitaria e investigación científica.

Como parte esencial del proceso formativo, se desarrolla el trabajo metodológico, el mismo contempla una serie de actividades que contribuyen a fomentar la preparación de los docentes y se encuentra dirigido principalmente a priorizar la formación y desarrollo de valores en los estudiantes.

Entre estas actividades se pueden destacar: las reuniones metodológicas, clases metodológicas, clases abiertas, talleres metodológicos, controles a clases y preparación de la carrera en general. Actualmente los métodos utilizados para almacenar y guardar la documentación no ofrecen garantía de protección, no aseguran posibilidades de acceso futuras y dificulta la actualización y corrección de los datos almacenados.

Para ampliar las posibilidades de controlar este proceso con mayor eficacia, en esta investigación se propone un “Sistema para la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”, el cual brindará facilidades de operación a los trabajadores del centro; al brindar rapidez en las consultas y facilidad de acceso mediante la red, así como un mayor flujo de información en menos tiempo y fiabilidad en los datos.

Este sistema es considerado sostenible, perdurable y muy adecuado para la gestión del trabajo metodológico por satisfacer las necesidades que lo originaron.

Abstract

The Holguin University "Oscar Lucero Moya" carries out five fundamental processes that sustain all the results obtained in the educational sphere by this center. These processes are: professionals' formation, universalization, postgraduate, university extension and scientific investigation.

As essential part of the formative process, the work is methodological developed that contemplates a series of activities that contribute to foment the preparation of the educational and is directed mainly to prioritize the formation and development of values in all students. Among these activities can stand out: the methodological meetings, methodological classes, opened classes, methodological shops, control to classes and preparation of the career in general.

At the moment the methods used to store and keep the documentation they don't offer guarantee of protection, they don't assure future possibilities of access and bring to update and correction of the stored data to enlarge with bigger effectiveness the possibilities to control this process intends a System for the administration of information of the methodological work in the Holguin University "Oscar Lucero Moya", will offer facilities operation to workers of the center; offering speed in the consultations and easiness way of access to net, as well as a bigger flow of information in less time and reliability in the data. This system is considered sustainable and very appropriate for the administration of the methodological work to satisfy the necessities that they originated it.

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS.....	9
1.1. TRABAJO METODOLÓGICO EN LA UNIVERSIDAD DE HOLGUÍN.....	9
1.2. TRABAJO METODOLÓGICO.....	11
1.3. TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE SISTEMAS INFORMÁTICOS ...	16
1.3.1. <i>Gestión de información.....</i>	17
1.3.2. <i>Sistemas de Información.....</i>	17
1.3.3. <i>Sistemas de información para gestionar el trabajo metodológico en la universidad</i> 18	
1.3.4. <i>Arquitectura Cliente – Servidor.....</i>	19
1.3.7. <i>Lenguajes de programación Web.....</i>	23
1.3.8. <i>Frameworks.....</i>	24
1.3.8.1. <i>Frameworks para el desarrollo Web.....</i>	25
1.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	27
1.4.1. <i>Proceso Unificado de Rational (RUP, por sus siglas en inglés).....</i>	28
1.4.2. <i>ICONIX.....</i>	29
1.4.3. <i>Herramientas de Apoyo a la Ingeniería de Software.....</i>	31
1.4.3.1. <i>Enterprise Architect.....</i>	31
1.5. TECNOLOGÍAS DE BASE DE DATOS (BD).....	32
1.5.1. <i>Sistemas de Gestión de Base de Datos.....</i>	33
1.6. IREPORT.....	35
1.7. TECNOLOGÍAS DE SERVIDORES <i>WEB</i>	36
1.7.1. <i>APACHE TOMCAT.....</i>	36
1.8. <i>SPRING SECURITY.....</i>	37

1.9. CONCLUSIONES DEL CAPÍTULO.....	39
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.	40
2.1 DEFINICIÓN Y ANÁLISIS DE LOS REQUISITOS.....	40
2.1.1 <i>Análisis de requisitos</i>	40
<i>Requerimientos Funcionales</i>	41
2.1.2 <i>Modelo del dominio</i>	45
2.1.3. <i>Prototipación rápida</i>	46
2.2 ANÁLISIS Y DISEÑO PRELIMINAR	47
2.2.1. <i>Modelos de casos de uso</i>	47
2.2.2. <i>Actores del sistema</i>	50
2.2.3. <i>Descripción de casos de uso</i>	50
2.2.4. <i>Análisis de robustez</i>	52
2.2.5. <i>Arquitectura técnica</i>	53
2.3 DISEÑO	58
2.3.1 <i>Diagramas de Secuencia</i>	59
2.3.2 <i>Diagrama de Clases</i>	60
2.3.3 <i>Modelo de datos</i>	61
2.4 IMPLEMENTACIÓN	62
2.4.1 <i>Estándares de codificación</i>	62
2.4.2 <i>Valoración de sostenibilidad</i>	64
2.4.2.1. <i>Dimensión Administrativa</i>	65
2.4.2.2. <i>Dimensión Socio-Humanista</i>	71
2.4.2.3. <i>Dimensión Ambiental</i>	72

2.4.2.4. Dimensión Tecnológica	74
2.5 CONCLUSIONES DE LA VSPI	75
2.6 VALORACIÓN DE LOS RESULTADOS OBTENIDOS.....	75
2.7 CONCLUSIONES DEL CAPÍTULO	76
CONCLUSIONES.....	77
RECOMENDACIONES	79
GLOSARIO DE TÉRMINOS	80
REFERENCIAS BIBLIOGRÁFICAS.....	83
ANEXOS:	I

Índice Figuras

Figura 1: Arquitectura cliente-servidor.	20
Figura 2: Esquema de trabajo ICONIX	30
Figura 3: Modelo del dominio.....	46
Figura 4: Diagrama de paquetes.....	48
Figura 5: Diagrama de casos de uso del Paquete Gestión de Codificadores.	49
Figura 6: Diagrama de robustez. Caso de Uso: Insertar actividad metodológica planificada en la sede central por facultad.	53
Figura 7: Arquitectura técnica del Sistema.	54
Figura 8: Diagrama del modelo de despliegue.	58
Figura 9: Diagrama de secuencia Caso de uso: Insertar actividad metodológica planificada en sede central por facultad.....	60
Figura 10: Diagrama de Clases	61
Figura 11: Modelo de datos	62
Figura 12: Diagrama de Casos de Uso: Paquete Seguridad	I
Figura 13: Paquete Gestión del trabajo metodológico de la facultad	II

Índice Tablas

Tabla 1: Actores del sistema.	50
Tabla 2: Insertar actividad metodológica planificada en la sede central por facultad.	52
Tabla 3: Factores de Complejidad técnica.	67
Tabla 4: Cálculo del TCF.....	67
Tabla 5: Factor ambiental.....	68
Tabla 6: Cálculo del ECF.	69
Tabla 7: Estimación del esfuerzo y costo.	70

Introducción

La Universidad de Holguín “Oscar Lucero Moya” (UHOLM), como centro de la Educación Superior ha centrado todos sus esfuerzos en lograr la formación integral de todos sus estudiantes y profesores. En sus aulas se forman profesionales dotados de los conocimientos y habilidades esenciales propias de su profesión, los cuales se encuentran altamente capacitados para desempeñar sus funciones laborales de forma independiente y creadora.

En su ascenso evolutivo tras años de infundir valores éticos y morales la universidad se ha convertido en el centro de referencia social cuando de educación, compromiso, transformación, e instrucción se trata.

Este centro de altos estudios cuenta con cinco procesos sustantivos: formación de profesionales, universalización, postgrado, extensión universitaria e investigación científica. Sus dos procesos de apoyo son el de Aseguramiento Material y Financiero (Proceso de Gestión económico-financiera) y Gestión de Recursos humanos y con tres procesos transversales Defensa y Protección, Relaciones Internacionales e Informatización [1].

La universidad concibe la formación de profesionales como un proceso de vital importancia en el que, de modo consciente y sobre bases científicas, se garantiza la preparación integral de los estudiantes universitarios. La preparación que reciben los educandos se concreta en una sólida formación científico-técnica, humanística y de altos valores ideológicos, políticos, éticos y estéticos, con el fin de lograr profesionales revolucionarios, cultos, competentes, independientes y creadores, para que puedan desempeñarse exitosamente en los diversos sectores de la economía y de la sociedad en general.

Por tanto, la labor educativa en la universidad constituye la principal prioridad en el proceso de formación y se desarrolla al utilizar un enfoque integral, que involucre a toda la comunidad universitaria con la participación activa de estudiantes, profesores y trabajadores en general. Para realizar esta labor cuenta con una

infraestructura integrada por sedes centrales y sedes universitarias enclavadas en los municipios; así como, con un claustro único integrado por profesores a tiempo completo, profesores a tiempo parcial, recién graduados en adiestramiento y alumnos ayudantes.

Como parte del proceso docente educativo, el trabajo metodológico es la actividad que se realiza en la universidad con el fin de mejorar de forma continua la preparación didáctica que poseen los profesores de las diferentes disciplinas y asignaturas. Los resultados obtenidos se concretan, fundamentalmente, en el desarrollo con calidad del proceso docente educativo, al lograr una adecuada integración de las clases con la actividad investigativa y laboral, así como con las tareas de alto impacto social y demás tareas de carácter extracurricular que cumplen los estudiantes [1].

Las funciones principales del trabajo metodológico son la planificación, la organización, la regulación y el control del proceso docente educativo. El adecuado desempeño de estas funciones, que tienen como sustento esencial lo didáctico, garantiza el eficiente desarrollo del proceso docente educativo.

El trabajo metodológico se realiza tanto de forma individual como colectiva y se desarrolla tanto en las sedes centrales como en las sedes universitarias, en cada uno de los departamentos, pertenecientes a cada una de las facultades existentes y se realiza por carreras, por disciplinas y por asignaturas.

En particular, el departamento docente es el nivel de dirección que rige el trabajo de los profesores y se ocupa directamente de proyectar un trabajo metodológico dirigido a priorizar la formación y desarrollo de valores en los estudiantes, desde el contenido de las asignaturas y disciplinas que son de su competencia [1].

En la UHOLM se desarrolla una serie de actividades de carácter metodológico que involucra a los profesores y a sus respectivos departamentos en cada una de las facultades así como a los directivos del centro. Durante todo el año se realizan disímiles actividades que complementan la labor profesional de los profesores entre

las cuales se pueden destacar: reuniones metodológicas, clases metodológicas, clases abiertas, talleres metodológicos, controles a clases y preparación de carrera en general.

A nivel institucional el trabajo metodológico se orienta y planifica trazando líneas de trabajo que son derivadas a las facultades y cada uno de los departamentos existentes en ellas. Estas líneas rigen las actividades que se llevan a cabo en el colectivo de disciplina así como en los colectivos de asignaturas. El colectivo de disciplina tiene como principal objetivo, lograr el cumplimiento con calidad de los objetivos generales de la disciplina. La conducción de este colectivo metodológico está a cargo del jefe de colectivo de disciplina, quien responde por esa labor, tanto en la sede central como en las sedes universitarias, ante el jefe del departamento al cual está subordinada esa disciplina.

A los colectivos de asignaturas corresponde elaborar el programa analítico de la asignatura para ambas modalidades de estudio (presencial o semipresencial), en correspondencia con el plan del proceso docente aprobado. Estos programas se someten a la consideración del colectivo de la disciplina correspondiente y del colectivo de la carrera. El jefe del departamento docente que corresponda aprueba el programa analítico de la asignatura, tomando en consideración la opinión del colectivo de la carrera.

En la actualidad, el registro y control de toda la información referente al trabajo metodológico se realiza de forma manual y es almacenada en formato Word y Excel. El proceso comienza cuando inicialmente la universidad de Holguín” Oscar Lucero Moya” elabora su estrategia de trabajo metodológico, la cual contempla todas las actividades que se realizarán durante un período de tiempo estimado. A partir de esa información, cada facultad del centro crea su propio plan de trabajo metodológico, con las acciones específicas que llevará a cabo y que se derivan de la estrategia general. Así posteriormente cada departamento ejecuta la misma acción, al realizar su plan de actividades con aquellas acciones que se realizarán a

este nivel. De igual forma se establecen actividades por disciplinas y por asignaturas.

Mensualmente se emiten informes que recogen los resultados obtenidos en cada una de las actividades y esta acción se realiza a nivel de departamento, de facultad y de institución. Como resultado el proceso no fluye con la rapidez y la calidad demandada, haciendo que las informaciones no estén organizadas adecuadamente y en ocasiones se torne impreciso el hecho de establecer comparaciones específicas entre los resultados obtenidos. El método que utilizan para guardar la documentación no ofrece garantía de protección, no asegura posibilidades de acceso futuras y dificulta en gran medida la actualización y corrección de los datos almacenados.

Estas deficiencias traen como consecuencia insatisfacción por parte del personal involucrado directamente con el control del proceso, lo cual se traduce entre otros aspectos, en resultados incorrectos de informes solicitados, lentitud de respuesta a consultas realizadas y acumulación de grandes volúmenes de documentación, pues es necesario preservar los datos para la obtención de informes de actividades realizadas con anterioridad elementos estos que evidencian la necesidad de un cambio.

Teniendo en cuenta lo antes expuesto, se plantea el siguiente **problema científico**:

¿Cómo favorecer la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”?

El problema anteriormente definido se enmarca en el **objeto de estudio**: Proceso de gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”.

El **objetivo** es: Desarrollar un sistema informático para favorecer la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”.

Se delimita como **campo de acción**: Informatización de la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”.

Para dar cumplimiento al objetivo propuesto se plantean las siguientes preguntas científicas:

- ✓ ¿Cuáles son los fundamentos teóricos asociados a la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”?
- ✓ ¿Cómo se realiza actualmente la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”?
- ✓ ¿Cómo diseñar un sistema informático que favorezca la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”?
- ✓ ¿Será sostenible la solución que se propone?
- ✓ ¿Cómo implementar el sistema informático propuesto?
- ✓ ¿Satisface el sistema informático las necesidades que lo originaron?

Para darle respuesta a las preguntas científicas y cumplir el objetivo general de la investigación se llevaron a cabo las siguientes tareas:

1. Compilar los fundamentos teóricos para comprender la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”.
2. Diagnosticar el estado actual del proceso de gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”.
3. Capturar los requerimientos del sistema informático que se propone.
4. Valorar la sostenibilidad del sistema informático.

5. Diseñar el sistema informático que se propone.
6. Implementar el sistema informático que se ha propuesto.
7. Valorar la efectividad de la solución propuesta mediante encuestas a especialistas.

Para darle solución a las tareas planteadas se utilizaron métodos de investigación teóricos y empíricos.

Métodos teóricos

Análisis y síntesis: Se utilizó para la elaboración de los fundamentos teóricos, procesamiento de la información y descomposición de los requerimientos del sistema para una mejor investigación y en la valoración de sostenibilidad del producto y síntesis conclusivas.

Modelación: Se utilizó en las diferentes etapas de elaboración del sistema, desde su análisis, hasta su implantación; se utilizó la metodología de ingeniería de *software* *ICONIX*, la cual emplea Lenguaje Unificado de Modelado (UML por sus siglas en inglés) como lenguaje de modelado, obteniéndose con esto los modelos que la metodología describe y la arquitectura del producto informático, permitió un mayor entendimiento de los diferentes procesos que se realizan y de los requisitos funcionales del sistema.

Histórico lógico: Permitted adquirir toda información relacionada con los procedimientos de gestión de la personalización, seguridad y funcionamiento del sistema, las normas, políticas y leyes por la que se rige la entidad, así como el análisis de las diferentes políticas de seguridad que debe cumplir el sistema y la lógica del negocio al posibilitar expresar las necesidades existentes en la Universidad de Holguín “Oscar Lucero Moya”.

Métodos empíricos

Revisión de documentos: permitió recopilar toda la información que necesaria para la fundamentación del proceso y la determinación de las herramientas a utilizar en la elaboración de la aplicación, recogida en las referencias bibliográficas.

Entrevista: Se utilizó para obtener informaciones a partir de los puntos de vista y opiniones referentes a la gestión del proceso y las deficiencias existentes, así como para asimilar conocimientos de especialistas con experiencia en el tema para el desarrollo de la investigación. Además permitió definir las necesidades y los requerimientos funcionales del sistema para adquirir un producto informático que cumpla las expectativas del cliente.

Observación científica: Se empleó esencialmente para el análisis, diseño e implementación del sistema. Facilitó que se tuviera una mejor seguridad y exactitud en las decisiones tomadas y los resultados obtenidos acerca del producto, además lograr la información directa sobre el proceso de desarrollo y objeto que está siendo investigado.

Encuesta: Fueron desarrolladas fundamentalmente para evaluar la satisfacción de los usuarios con respecto a los resultados obtenidos con la implantación del sistema.

Métodos estadísticos

Para el tratamiento de las encuestas se hizo uso de métodos estadísticos tales como el trabajo con frecuencias y porcentajes.

Con este trabajo se dispondrá de una herramienta informática que garantizará que la información relacionada con el trabajo metodológico que se lleva a cabo por la Universidad de Holguín “Oscar Lucero Moya” transite eficientemente por las vías correspondientes en sus diferentes etapas. Con la investigación se eliminará el trabajo manual que se hace actualmente, almacenará un gran volumen de información, agilizará las búsquedas de los documentos que se encuentran

inmersos en el proceso antes mencionado y se generarán los informes que debe entregarse cada cierto tiempo.

Con el objetivo de exponer los resultados de la investigación se ha dividido el presente documento en introducción, dos capítulos, conclusiones, recomendaciones, bibliografía y anexos.

El **Capítulo 1** (Fundamentos teóricos) está orientado a mostrar los principales fundamentos, conceptos, metodologías y herramientas utilizados para llevar a cabo el desarrollo de la investigación.

En el **Capítulo 2** (Descripción de la solución propuesta) se muestran los flujos de trabajo principales de la ingeniería de software realizada al sistema propuesto, a través de la metodología ICONIX y se incluye un estudio de factibilidad y una valoración de sostenibilidad del sistema como producto informático, según las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

Capítulo 1: Fundamentos teóricos.

En este capítulo se exponen los fundamentos teóricos que sustentan la presente investigación; se caracterizan las principales categorías relacionadas con el objeto de estudio para entender con mayor claridad el proceso de gestión del trabajo metodológico. Además, se exponen conceptos, herramientas y técnicas sobre las que se apoya la propuesta, haciendo énfasis en sus características y ventajas. Por último se fundamenta la metodología de desarrollo a utilizar.

1.1. Trabajo Metodológico en la Universidad de Holguín.

En la Universidad de Holguín “Oscar Lucero Moya” se desarrolla una serie de actividades de carácter metodológico como parte del proceso docente-educativo, al iniciar cada año, un nuevo curso escolar. Estas actividades involucran a todos los departamentos existentes en cada una de las facultades, así como a cada uno de los profesores, alumnos en adiestramiento y directivos del centro. Estas actividades se realizan tanto en los municipios como en las sedes centrales y en esta última se llevan a cabo tanto a nivel de facultad como de departamento.

A nivel institucional el trabajo metodológico se orienta y planifica trazando líneas de trabajo que son derivadas a las facultades y cada uno de los departamentos existentes en ellas. Inicialmente la universidad elabora su estrategia de trabajo metodológico, la misma contempla todas las actividades que se realizarán durante un período de tiempo estimado, luego a partir de esa información cada facultad del centro, crea su propio plan de trabajo metodológico, con las acciones específicas que llevará a cabo y que se derivan de la estrategia general. Así posteriormente cada departamento ejecuta la misma acción, al realizar su plan de actividades con aquellas acciones que se realizarán a este nivel. De igual forma se pueden establecer actividades por disciplinas y por asignaturas.

Las principales actividades que se realizan son, la preparación de la carrera, la reunión metodológica, la clase metodológica, la clase abierta, el taller

metodológico y el control a clase; este último de vital importancia para comprobar la preparación de los profesores en cada una de las asignaturas; las veces que se realiza el mismo está en dependencia de la categoría docente del profesor al cual va dirigido. Pueden supervisar esta actividad los rectores, vicerrectores, decanos, vicedecanos, asesores técnicos docentes, jefes y segundos jefes de departamentos docentes, directores y subdirectores de sedes universitarias, profesores principales, responsables de unidades docentes, así como los jefes de colectivos metodológicos de la sede central y de las sedes universitarias. Conjuntamente con ellos podrán participar otros profesores, teniendo en cuenta que su categoría docente no sea menor que la del controlado.

Según [1] los documentos que se conservan en las sedes universitarias para avalar el trabajo metodológico son:

- ✓ El plan de trabajo metodológico de la sede universitaria aprobado por la dirección del centro de educación superior
- ✓ Los documentos que recogen la planificación de las diferentes actividades presenciales y el control del proceso docente educativo
- ✓ Los documentos que plasman los resultados de los controles realizados a las actividades docentes por parte de la sede, los departamentos docentes y las facultades

Según [1] los documentos que se conservan en las facultades para avalar el trabajo metodológico son:

- ✓ El plan de trabajo metodológico de la facultad aprobado por la dirección del centro de educación superior
- ✓ Los documentos rectores del proceso docente educativo de las carreras que dirigen, así como los documentos que avalan sus modificaciones

- ✓ Los documentos que avalan el sistema de planificación, organización y control del proceso docente educativo en ese nivel, en correspondencia con las indicaciones del rector al respecto
- ✓ Los documentos que plasman los resultados de los controles realizados a las actividades docentes

Según [1] los documentos que se conservan en el departamento docente para avalar el trabajo metodológico son:

- ✓ El plan de trabajo metodológico aprobado por el decano
- ✓ Los documentos que plasman los resultados de los controles a las actividades docentes realizadas
- ✓ Los programas analíticos de las asignaturas impartidas en el curso.
- ✓ Los documentos que avalan el sistema de planificación, organización y control del proceso docente-educativo en ese nivel, en correspondencia con las indicaciones del rector al respecto

1.2. Trabajo Metodológico.

El trabajo metodológico es de vital importancia para la universidad pues su realización va encaminada a alcanzar óptimos resultados en dicho proceso. Este trabajo jerarquiza la labor educativa desde la instrucción, para satisfacer plenamente los objetivos formulados en los planes de estudio. El trabajo metodológico se concreta fundamentalmente, en el desarrollo con calidad del proceso docente educativo. Propicia una adecuada integración de las clases con la actividad investigativa y laboral, así como con las tareas de alto impacto social y demás tareas de carácter extracurricular que cumplen los estudiantes [1].

Las funciones principales del trabajo metodológico son la planificación, la organización, la regulación y el control del proceso docente educativo. El adecuado

desempeño de estas funciones, que tienen como sustento esencial lo didáctico, garantiza el eficiente desarrollo del proceso docente educativo.

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

Según [1] los tipos fundamentales del trabajo docente-metodológico son:

- ✓ Preparación de la carrera
- ✓ Reunión metodológica
- ✓ Clase metodológica
- ✓ Clase abierta
- ✓ Taller metodológico
- ✓ Control a clase

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, al desarrollar investigaciones, o utilizar los 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.

La preparación de la carrera es la labor que realizan los profesores que integran ese colectivo para asegurar un trabajo coordinado y sistemático que oriente, desde el punto de vista didáctico, el desarrollo de las disciplinas y los años en que se estructura la carrera. Para cumplir con el modelo del profesional, y tomar como elemento fundamental el enfoque integral para la labor educativa y político

ideológica de los estudiantes, la preparación de la carrera involucra además a los docentes en adiestramiento [1].

La reunión metodológica es el tipo de trabajo docente–metodológico que viabiliza el análisis, y toma de decisiones acerca de temas vinculados al proceso docente educativo para su mejor desarrollo. Las reuniones metodológicas se desarrollarán en los centros de Educación Superior, facultades, sedes universitarias, filiales, unidades docentes, departamentos docentes y colectivos metodológicos en general y estarán dirigidas por los jefes de cada nivel de dirección o colectivo metodológico o por profesores de vasta experiencia y elevada maestría pedagógica. La planificación de las reuniones metodológicas aparecerá en el plan de trabajo metodológico elaborado al principio de cada curso en cada uno de los niveles correspondientes.

La clase metodológica es el tipo de trabajo docente-metodológico que, mediante la demostración, la argumentación y el análisis, orienta a los profesores sobre algunos aspectos de carácter metodológico que contribuyen a su preparación para la ejecución del proceso docente educativo. La clase puede tener carácter demostrativo o instructivo, y responderá a los objetivos metodológicos formulados. En la clase metodológica instructiva la orientación se realizará mediante la argumentación y el análisis de los aspectos propios del contenido objeto de la actividad. En la clase metodológica demostrativa la orientación se realizará mediante el desarrollo de una actividad docente modelo en la que, preferiblemente, estarán presentes los estudiantes.

Las clases metodológicas se realizarán, fundamentalmente, en los colectivos de asignatura y de disciplina, aunque pueden organizarse también en otros niveles de dirección y colectivos metodológicos, cuando sea necesario y se llevarán a cabo por los jefes de cada nivel de dirección o colectivos metodológicos, así como por profesores de vasta experiencia y elevada maestría pedagógica. La planificación de las clases metodológicas aparecerá en el plan de trabajo metodológico

elaborado al principio de cada curso en cada una de las instancias correspondientes.

La clase abierta es el tipo de trabajo docente-metodológico que permite, mediante el análisis de una actividad docente prevista para los estudiantes, la orientación a los profesores de un departamento docente o de un colectivo metodológico, sobre algunos aspectos de carácter metodológico que contribuyen a su preparación para la ejecución del proceso docente educativo. La clase abierta responderá a los objetivos metodológicos formulados y se realizarán, fundamentalmente, en los colectivos de asignatura y de disciplina, aunque pueden organizarse también en otros niveles de dirección y colectivos metodológicos, cuando sea necesario. Una vez concluida la clase abierta, se discutirá por todos los participantes, sin la presencia de los estudiantes. El dirigente de la actividad resumirá la discusión, al señalar los principales logros y deficiencias observados en la clase y emitirá las recomendaciones que correspondan [1].

Las clases abiertas estarán dirigidas por los jefes de cada nivel de dirección o colectivo metodológico. La planificación de las clases abiertas aparecerá en el plan de trabajo metodológico elaborado al principio de cada curso en cada una de las instancias correspondientes.

El taller metodológico es el tipo de trabajo docente-metodológico que tiene como objetivo debatir acerca de una problemática relacionada con el proceso de formación y en el cual los profesores presentan experiencias relacionadas con el tema tratado. En él se proyectan alternativas de solución a dicho problema a partir del conocimiento y la experiencia de los participantes.

El control de la actividad docente es aquel que se realiza a una de las formas organizativas del proceso docente educativo, previsto en el horario de clases de los estudiantes, y estará dirigido a comprobar el logro de los objetivos propuestos para dicha actividad. Al finalizar el control, sin la presencia de los estudiantes, el responsable dirigirá el análisis, dará las conclusiones al controlado, señalará los

principales logros, las deficiencias y las recomendaciones. Los resultados del control se recogerán en un documento que será firmado por el docente controlado, como constancia de que fue informado de sus resultados.

El control de la actividad docente se calificará mediante las categorías de Excelente, Bien, Regular o Mal. Se exceptuarán de una calificación, los casos que se determinen por el responsable del control. Podrán realizar controles a las actividades docentes los rectores, vicerrectores, decanos, vicedecanos, asesores técnicos docentes, jefes y segundos jefes de departamentos docentes, directores y subdirectores de sedes universitarias, profesores principales, responsables de unidades docentes, así como los jefes de colectivos metodológicos de la sede central y de las sedes universitarias. Conjuntamente con ellos podrán participar otros profesores, teniendo en cuenta que su categoría docente no sea menor que la del controlado.

Los controles a realizar a cada profesor dependerán de su maestría pedagógica y del dominio que tenga del contenido de la asignatura que imparte. El jefe de departamento docente y los coordinadores de carrera de las sedes universitarias, según corresponda, determinarán la cantidad de controles a clases que se realizará a los docentes. El plan de controles a las actividades docentes se elabora por cada jefe de departamento docente. Para las carreras que se estudian en las sedes universitarias participará en la elaboración del plan el coordinador de la carrera en cuestión. Los profesores no podrán conocer las actividades docentes en las que serán controlados.

Todos los profesores que impartan por primera vez una asignatura deberán ser controlados al menos una vez en el período docente, con el objetivo de verificar su adecuada preparación. Los decanos, vicedecanos y directores de sede elaborarán también su plan de controles a clases en coordinación con el jefe de departamento docente y con el coordinador de la carrera en la sede, según corresponda.

A nivel institucional el trabajo metodológico se orienta y planifica trazando líneas de trabajo que son derivadas a las facultades y cada uno de los departamentos existentes en ellas. Estas líneas rigen las actividades que se llevan a cabo en el colectivo de disciplina así como en los colectivos de asignaturas. El colectivo de disciplina tiene como principal objetivo, lograr el cumplimiento con calidad de los objetivos generales de la disciplina. La conducción de este colectivo metodológico está a cargo del jefe de colectivo de disciplina, quien responde por esa labor, tanto en la sede central como en las sedes universitarias, ante el jefe del departamento al cual está subordinada esa disciplina.

A los colectivos de asignaturas corresponde elaborar el programa analítico de la asignatura para ambas modalidades de estudio (presencial o semipresencial), en correspondencia con el plan del proceso docente aprobado. Estos programas se someten a la consideración del colectivo de la disciplina correspondiente y del colectivo de la carrera. El jefe del departamento docente que corresponda aprueba el programa analítico de la asignatura, tomando en consideración la opinión del colectivo de la carrera.

1.3. Tendencias y tecnologías actuales para el desarrollo de sistemas informáticos

El desarrollo de aplicaciones informáticas requiere del empleo de modernas tecnologías y herramientas que permitan realizar productos informáticos que se ajusten a las necesidades de los usuarios y lo hagan con la calidad requerida. Para ello se escogió un grupo de las mismas por las ventajas y características que proporcionan su empleo para el desarrollo de este producto informático. A continuación se hace un estudio detallado de las mismas.

1.3.1. Gestión de información

La gestión de la información puede definirse como “el proceso de analizar y utilizar la información que se ha recabado y registrado para permitir a los administradores (de todos los niveles) tomar decisiones documentadas”[2].

El valor de la gestión de la información en las organizaciones constituye un factor crítico de éxito, que se acepta unánimemente como recurso indispensable para ampliar la competitividad, aumentar la calidad y la satisfacción de los usuarios, así como para desenvolverse en el ámbito global. Esto ha conducido al desarrollo de sistemas de información para el manejo, tratamiento y uso de la información en las organizaciones que impulsados por los recursos humanos en su condición de agentes y líderes del cambio, posibilitan la agilidad y facilidad de acceso necesaria para ejecución de los procesos organizacionales y la toma de decisiones.

1.3.2. Sistemas de Información

Con el avance de la Tecnologías de la Información (TI), las organizaciones se apoyan en Sistemas de Información (SI) para informatizar los procesos productivos y de esta forma ganar ventajas competitivas e información para la toma de decisiones.

Un SI se define como un grupo de componentes interrelacionados que trabajan en conjunto hacia un objetivo común al aceptar entradas y producir salidas en un proceso de transformación organizado. Los equipos computacionales o *hardware*, el personal que interactúa con el sistema, los datos e información que son introducidos, almacenados, procesados y visualizados a través del sistema y el *software*, son los principales elementos que posibilitan su funcionamiento [3]. Los SI pueden comunicarse con otros sistemas e incluso tener inmersos subsistemas [2].

Según [4] el empleo adecuado de los SI en la organización mejora de manera gradual la:

- ✓ Calidad en productos y servicios
- ✓ Atención a los clientes
- ✓ Relación con los proveedores
- ✓ Condiciones en el ambiente de trabajo
- ✓ Comunicación interpersonal
- ✓ Estimulación de la participación de los trabajadores
- ✓ Reducción del número de procesos de gestión/producción
- ✓ Simplificación de los procesos de gestión/producción
- ✓ Eficiencia en el uso de los recursos
- ✓ Diseño de nuevas y mejores herramientas para la gestión de la dirección

La información que generan los SI sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones. Son interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final. Apoyan la toma de decisiones que, por su misma naturaleza, son repetitivas y de decisiones no estructuradas que no suelen repetirse.

Los elementos que interactúan entre sí son: el equipo computacional, el recurso humano, los datos o información fuente, programas ejecutados por las computadoras, las telecomunicaciones y los procedimientos de políticas y reglas de operación.

1.3.3. Sistemas de información para gestionar el trabajo metodológico en la universidad

Se llevó a cabo una investigación en otras instituciones educacionales del país, para determinar la existencia de sistemas informáticos previos, que fuesen empleados para gestionar toda la información referente al trabajo metodológico.

Entre las instituciones seleccionadas para el estudio se encuentran la Universidad Central (UCLV) "Marta Abreu" de Las Villas, la Universidad de Oriente (UO), la Universidad de Camagüey (UC) "Ignacio Agramonte Loynaz" y la Universidad de Cienfuegos (UCf) "Carlos Rafael Rodríguez", en cada una de las cuales la gestión y control de este proceso se desarrolla de forma manual, archivando la información, en formatos como Word y Excel medios que no aseguran la confiabilidad de los datos, la integridad de la información y mucho menos el acceso a la misma desde cualquier entorno. De igual manera se comprobó en la esfera internacional la inexistencia de sistemas similares.

En la Universidad de Holguín "Oscar Lucero Moya", el sistema para la gestión de información del trabajo metodológico que se desarrolló, es la primera herramienta diseñada para procesar toda la información relacionada con el tema.

1.3.4. Arquitectura Cliente – Servidor

El manejo de la información ubicada en los diferentes sistemas existentes en las organizaciones, dado el uso imprescindible con carácter laboral o personal de las computadoras, hace que esta se esparza a través de la red a la que tienen acceso todos los clientes internos.

La arquitectura cliente-servidor es "La tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo (...) en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores"[5].

En la arquitectura cliente-servidor, un servidor (proveedor de servicio), se activa y espera las solicitudes de los clientes (consumidor de servicios). Habitualmente, programas cliente múltiples comparten los servicios de un programa servidor común.

En la Figura 1 se puede observar las entidades que interactúan con esta arquitectura para lograr su adecuado funcionamiento.

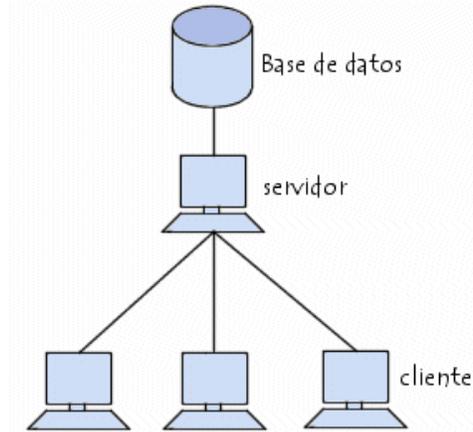


Figura 1: Arquitectura cliente-servidor.

Fuente: Elaboración propia

Según [6] dentro de los aspectos fundamentales que caracteriza esta arquitectura se encuentran los siguientes:

1. Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
2. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores se realizan de una manera transparente para el usuario final.
3. La interrelación entre el *hardware* y el *software* están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
4. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones clientes.

5. Muchos clientes pueden utilizar un mismo servicio sin tenerlo, lo que ahorra espacio y disminuye la carga de trabajo del cliente.

Para la comunicación entre los componentes cliente y servidor, existe una infraestructura lógica de comunicaciones que proporciona los mecanismos básicos de direccionamiento y transporte.

La arquitectura Cliente/Servidor proporciona una mayor consistencia a la información de la empresa al contar con un control centralizado de los elementos compartidos y llevar fácilmente la información a donde se necesita. Las ventajas anteriormente referidas inducen a la autora a su utilización en la aplicación que se propone.

1.3.4.1 Aplicaciones de Escritorio

Por aplicaciones de escritorio se entiende toda aplicación que ha sido desarrollada para ser ejecutada en una plataforma específica, ya sea *Windows*, *Linux* o *Mac*. Según [7] el desarrollo sobre una plataforma, normalmente, implica que la aplicación “no” pueda ser ejecutada en otras.

Las principales ventajas de una aplicación de escritorio según [8] son:

- ✓ Navegación e interfaz de usuario más rápidas que en una aplicación *Web*
- ✓ Normalmente no es necesaria una conexión a *Internet* ya que funciona localmente.
- ✓ Es mejor opción cuando es necesario realizar numerosos cálculos o procesos de Unidad Central de Procesamiento (CPU, por sus siglas en inglés) muy intensivos
- ✓ Fácil acceso a recursos locales: disco duro, impresora, memoria de vídeo, sonido, *mouse*, teclado, etc.

- ✓ Fácil acceso a aplicaciones locales en caso de que sea necesario compartir datos entre aplicaciones

1.3.4.2 Aplicaciones Web

Una aplicación *Web* es aquella que los usuarios pueden utilizar accediendo a un servidor *Web* a través de *Internet* o de una intranet mediante un navegador. Es una aplicación *software* que se codifica en un lenguaje soportado por los navegadores *Web* (Lenguaje de Marcas Hipertextuales o HTML por sus siglas en inglés, JavaScript, Java, etc.), en la que se confía la ejecución al navegador. Es una de las herramientas más eficaces para visualizar y gestionar información y se logra una mayor eficiencia en el tratamiento y manejo de la misma [6].

Para un adecuado uso de estas aplicaciones, se debe tener dominio de las ventajas que ofrece la misma, entre ellas se pueden mencionar:

- ✓ Se facilita el trabajo a distancia
- ✓ No se requieren complicadas combinaciones de *hardware* y *software* para utilizar estas aplicaciones
- ✓ Tendrá mayor control de datos y mejor seguridad en las diferentes secciones del sitio
- ✓ Permite tener un avanzado sistema de consultas, altas, bajas y modificaciones de datos provenientes de cualquier área de la empresa; lo cual mantendrá la información siempre actualizada
- ✓ Las aplicaciones *web* son notorias debido a lo práctico del navegador *web* como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones *web* sin distribuir e instalar *software* a miles de usuarios potenciales

Por las ventajas anteriormente mencionadas y en especial porque no se requiere de instalar un software en los clientes para poder acceder a la información desde

cualquier entorno, se propone emplear este tipo de aplicación para la elaboración de la solución informática propuesta.

1.3.5. Lenguajes de programación *Web*

Actualmente las aplicaciones *Web* generan dinámicamente una serie de páginas en un formato estándar, como HTML o Lenguaje Extensible de Marcado de Hipertexto (XHTML, por sus siglas en inglés), soportadas por los navegadores *Web* comunes. Se utilizan lenguajes interpretados en el lado del cliente, para añadir elementos dinámicos a la interfaz de usuario. Durante la sesión, el navegador *Web* interpreta y muestra en pantalla las páginas, se muestra como cliente para cualquier aplicación *Web*.

También se utilizan lenguajes de lado servidor, que son aquellos lenguajes que se caracterizan por implementar la lógica del negocio dentro del servidor, además de encargarse de la tarea del acceso a las bases de datos y el tratamiento de la información.

Groovy

Groovy es un lenguaje de programación dinámico creado por Guillaume Laforge, nacido con la misión de acoplarse de forma efectiva con Java, trabajar en conjunto con la Máquina Virtual de Java (JVM por sus siglas en inglés) y soportar los tipos de datos estándar, pero añadiendo características dinámicas y sintácticas presentes en otros lenguajes como Python [9].

El código fuente Groovy se compila a bytecodes, y es posible instanciar objetos Java desde Groovy y viceversa. Lo que Groovy aporta es una sintaxis que aumenta enormemente la productividad. Todo lo anterior, unido a que la mayor parte de código escrito en Java es totalmente válido en Groovy hace que este lenguaje sea de muy fácil adopción.

Según [10] las principales características que definen este lenguaje se encuentran:

- ✓ Soporta el tipado fuerte y dinámico: Le aporta una mayor flexibilidad al lenguaje al permitir que el programador defina el tipo de dato que soportará una variable o admita el tipo de dato que reciba por parte del usuario
- ✓ Lenguaje de Dominio Específico (DSL, por sus siglas en inglés): Debido a su naturaleza dinámica puede utilizarse para modelar un dominio o un área de conocimiento particular con el objetivo de proporcionar construcciones sintácticas que se acerquen al dominio que se quiere representar
- ✓ Multiparadigma: Además de la programación orientada a objetos admite la programación funcional y programación orientada a aspectos
- ✓ Curva de aprendizaje reducida: Su similitud con la sintaxis de Java y otros lenguajes muy estandarizados como es el caso de *Python*, además de las flexibilidades que ofrece lo convierte en un lenguaje fácil de aprender y utilizar por diferentes tipos de programadores
- ✓ Sintaxis compacta: De esta forma su código es fácil para leer y comprender.

Por todos los elementos anteriormente expuestos la autora se inclina a utilizar Groovy como lenguaje de programación para el desarrollo de la aplicación que se propone.

1.3.6. Frameworks

Un *framework* es una estructura software compuesta de componentes para el desarrollo de una aplicación. Se puede considerar como una aplicación genérica incompleta y configurable a la que es posible añadirle las últimas piezas para construir una aplicación concreta. Representa un patrón para el desarrollo y/o la implementación [11].

Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un *framework Web*, por tanto, es un conjunto de

componentes (por ejemplo clases en Java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas *Web*.

Un *framework* está compuesto por el patrón Modelo-Vista-Controlador (MVC) es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema [6].

1.3.6.1 Frameworks para el desarrollo Web

Un framework web constituye el esqueleto sobre el cual varios objetos son integrados para una solución dada. Son diseñados con el intento de facilitar el desarrollo de *software*, permitiendo de esa manera a los diseñadores y programadores pasar más tiempo al identificar los requerimientos de *software* y tratar con los detalles de bajo nivel para proveer un sistema funcional [11].

La validación de las entradas de los usuarios respecto a las reglas del negocio es una de las tareas más engorrosas en el desarrollo de aplicaciones *web*. Sin embargo la mayoría de los *frameworks* dedicados al desarrollo *web* tienen, como mínimo, una estrategia para definir reglas de validación y mostrar mensajes de error cuando estas reglas no se cumplen correctamente [12].

En conclusión, un *framework* de tipo web es una base de programación que atiende a sus sucesores de una forma estructural y/o en cascada permitiendo cualquier respuesta ante sus miembros, o secciones de una aplicación *web*.

Grails

Grails es un *framework* de código abierto que se integra bien con aplicaciones web para la máquina virtual de java (JVM por sus siglas en inglés), desarrollada sobre el lenguaje de programación *Groovy*. El mismo pretende ser un marco de trabajo altamente productivo siguiendo paradigmas tales como convención sobre configuración, al proporcionar un entorno de desarrollo estandarizado y ocultar gran parte de los detalles de configuración al programador [13].

Al utilizar *Groovy* como lenguaje de programación incorpora propiedades nuevas en las clases del útiles de desarrollo de Java (JDK por sus siglas en inglés), o incluso en los tipos primitivos, lo que le permite añadir métodos de búsqueda dinámicos a las clases del modelo de datos. Estas funcionalidades, le permiten optimizar código disminuyendo el tiempo de desarrollo y lo hacen mucho más flexible [14].

El objetivo de *Grails* es brindar al usuario un entorno de alta productividad, extensible y fácil de utilizar, ofreciendo el balance adecuado entre consistencia y funcionalidades potentes. Una de las principales ventajas que ofrece este *frameworks* es el uso de *GORM (Grails Object Relational Mapping)* por sus siglas en inglés, una adaptación de *Hibernate* que le permite a los desarrolladores representar su modelo del dominio en correspondencia con aquellos elementos que considera relevante en su negocio. A partir de estos objetos o clases relacionados, se obtendrán cada una de las entidades de la Base de Datos y sus respectivas relaciones [15]. Además no debe preocuparse por la conexión con esta, pues el *framework* tiene integrado un driver para la conexión de base de dato en Java (JDBC por sus siglas en inglés) que lo realiza. Esto se traduce en una disminución del tiempo de desarrollo y en una aproximación al paradigma orientado a objeto pues le permite al programador centrarse en estos desde el inicio mientras que las entidades le resultan transparentes durante el desarrollo.

La funcionalidad que más parece impactar al nuevo usuario es la denominada “*scaffolding*” y permite, a partir de una clase de dominio, generar el comportamiento para listado, creación, edición y eliminación de objetos de dicha clase, con la ejecución de un simple comando [10]. Al ejecutar este comando, *Grails* toma la clase de dominio seleccionada y crea las acciones correspondientes, así como las vistas necesarias. También toma en cuenta las restricciones de los campos de dicha clase del dominio y las refleja en validaciones para los formularios (con mensajes de error predefinidos), así como también en ajustes sobre la estructura de la base de datos [16].

Dentro de la arquitectura de *Grails* el hecho más significativo es la integración transparente de los *frameworks* Java de código abierto más utilizados, como lo son: *Spring*, *Hibernate*, y *SiteMesh*.

La filosofía de *Grails* está fuertemente vinculada a pruebas; esto se ve reflejado en la creación automática de casos de prueba al crearse nuevas clases del dominio, controladores o servicios. Es por ello que otro de los *frameworks* Java incluidos es JUnit, el más usado para pruebas unitarias. También existe soporte para las pruebas de integración y de interfaz a través de extensiones.

Por todos los argumentos expuestos se escoge este *framework* para el desarrollo de la propuesta, puesto que permite a desarrolladores Java trabajar sobre un ambiente familiar, robusto y confiable, sin necesidad de extensos archivos de configuración, pudiendo integrar sistemas existentes, y obtener así un proceso de desarrollo ágil y productivo.

1.4 Metodologías de desarrollo de software

La construcción de sistemas informáticos se enfrenta a continuos cambios de las tecnologías de implementación, lo que implica realizar esfuerzos importantes tanto en el diseño de la aplicación para integrar las tecnologías que incorpora, como en el mantenimiento, para adaptar la aplicación a cambios en los requisitos; el volumen y complejidad de los productos en la industria de software aumentan

considerablemente, y se exige mayor calidad y productividad en menos tiempo, por parte de los clientes; por tanto en el proceso de desarrollo de *software* la metodología no debe omitirse si se quiere lograr un sistema de calidad; es decir, que cumpla con los requerimientos de los usuarios, disminuya los costos, la complejidad y los problemas de mantenimiento [5].

El paso inicial consiste en elegir cuál de las dos tendencias existentes (ágiles y tradicionales) se adapta mejor tanto a los intereses del cliente como a los del equipo de desarrollo, de forma tal que el equipo maximice sus capacidades y que los clientes queden satisfechos más allá de las necesidades definidas al inicio.

"Una metodología de desarrollo de *software* consiste en el conjunto de filosofías, fases, procedimientos, técnicas, reglas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar aplicaciones informáticas con el fin de hacerlo más predecible y eficiente" [17]. No existe una metodología de *software* universal, su elección depende de las características de cada proyecto.

A continuación se describen algunas características de estas metodologías.

1.4.1. Proceso Unificado de Rational

Proceso Unificado de Rational (RUP por sus siglas en inglés), es una de las metodologías robustas más utilizadas durante el proceso para el desarrollo de un proyecto de *software* que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Dicho proceso tiene tres características fundamentales.

- ✓ Es dirigido por casos de uso, es decir, el proyecto se orienta a la importancia que tiene para el usuario lo que debe hacer el producto

- ✓ Es centrado en la arquitectura pues relaciona la toma de decisiones que indica cómo tiene que ser constituido el sistema y en qué orden se debe hacer
- ✓ Es iterativo e incremental, divide el proyecto en mini proyectos donde los casos de usos y la arquitectura cumplen sus objetivos de manera más depurada [18]

1.4.2. ICONIX

Es una metodología elaborada por Doug Rosemberg y Kerdall Scott. Considerado por este autor como un “proceso” de desarrollo de *software* práctico, emplea el modelo de desarrollo de *software* basado en el enfoque o paradigma de desarrollo de *software* iterativo e incremental y utiliza el método orientado a objetos con el enfoque basado en el Lenguaje Unificado de Modelado (UML por sus siglas en inglés); es una metodología ágil que toma varios aspectos de RUP y de Programación Extrema (XP, por sus siglas en inglés), y constituye el punto medio entre estas metodologías, la primera muy pesada y la segunda muy ágil [19]

ICONIX es dirigido por casos de uso como RUP, pero sin la excesiva cantidad que este propone, y relativamente ligero y pequeño como XP, pero no desprecia el análisis y diseño como éste.

Presenta claramente las actividades de cada fase y exhibe una secuencia de pasos que deben ser seguidos, ofrece trazabilidad, que no es más que la capacidad de seguir una relación entre los diferentes artefactos del *software* producidos. La metodología brinda un uso dinámico del UML porque utiliza solo algunos diagramas, al contrario de RUP [20].

La Figura 2 muestra el esquema general de la metodología ICONIX, en ella se reflejan los principales modelos y los diagramas.

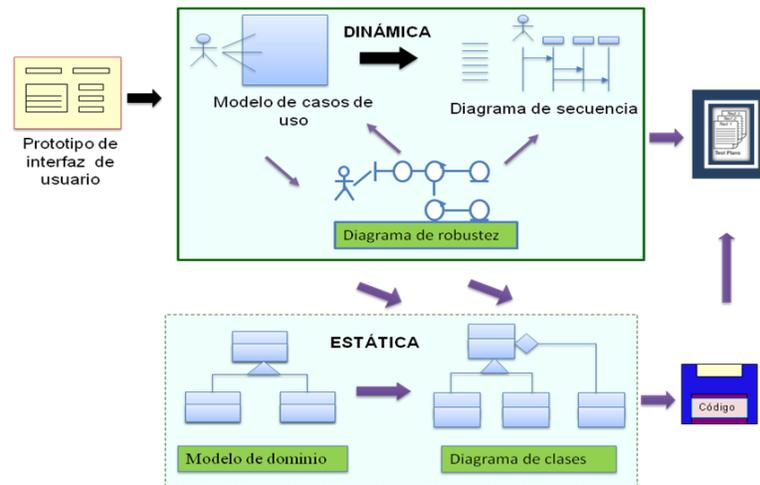


Figura 2: Esquema de trabajo ICONIX

Fuente: [12]

Según [5] modelos ágiles de metodologías como *ICONIX* ofrecen una mayor ventaja en su curva de aprendizaje. La metodología *ICONIX* incluye el análisis de robustez, el cual reduce la ambigüedad en las descripciones de caso de uso y asegura que sean escritos en el contexto de un modelo de dominio.

"Este proceso es iterativo e incremental, pues permite varias iteraciones entre el desarrollo del modelo del dominio e identificar y analizar los casos de usos. El modelo estático es incremental refinado a través del modelo dinámico" [21]. Cuenta con amplias bondades en los servicios de negocios. Las soluciones de negocios se enfocan en los servicios en áreas primarias (la experiencia del usuario, funcionalidad comercial, e infraestructura), con la planeación y la estrategia fortalece cada área. Esta especialización en las áreas primarias tributa al éxito final de los productos que se entregan a los clientes

ICONIX se adapta a los requerimientos ya que está menos orientada al documento y más centrada en el código fuente, exigiéndose menor documentación y ahorro de tiempo. No existe una enorme plantilla de documentación para los casos de uso, sino que estos hablan de qué están haciendo los usuarios en las interfaces finales,

es conducida por casos de uso pero no incorpora tantos artefactos UML, sino que utiliza un subconjunto mínimo pero suficiente de estos para realizar un buen trabajo de Ingeniería de Software en poco tiempo [12]

Para el caso de esta investigación ha sido seleccionada la metodología ICONIX para ser utilizada en el sistema propuesto ya que sus características y su modo de proceder permiten la flexibilidad y dinamismo del sistema a desarrollar. Además el paso por cada una de las etapas de desarrollo que ICONIX plantea, brinda organización y consistencia durante la concepción de nuevos sistemas, al contar con los artefactos necesarios para lograr productos cada vez con mayor calidad.

1.4.3. Herramientas de Apoyo a la Ingeniería de Software

La Ingeniería de *Software* Asistida por Computación (CASE, por sus siglas en inglés) se considera como la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

Las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar los Sistemas de Información Gerenciales. También permiten aumentar la productividad en el desarrollo de un proyecto y como herramientas que son, deben ser aplicadas a una metodología determinada.[22]

1.4.3.1. Enterprise Architect

Como herramienta de modelado UML se utilizó el *Enterprise Architect* (EA, por sus siglas en inglés) el cual cubre todos los aspectos del ciclo de desarrollo de un *software*, el mismo está enfocado para la metodología *ICONIX*, desde la recopilación de requisitos, a través de análisis, diseño de modelos, pruebas, control de cambios y hasta el mantenimiento de la aplicación, para dotar al *software* de trazabilidad completa. EA es una herramienta poderosa para la

especificación, la documentación y la construcción de su *software* y proyectos de procesos de negocio. EA es muy conocido por sus numerosas funcionalidades.

Algunas de las características claves se destacan en la siguiente lista según [20]:

- ✓ Centralizar toda la documentación de los procesos y sistemas de información
- ✓ Dependencias entre los elementos del modelo, la dinámica del sistema y el estado
- ✓ Permite integrar las instalaciones de *Enterprise Architect* con IDEs y otras tecnologías, y plantillas
- ✓ Velocidad: un intérprete espectacularmente rápido, incluso con modelos grandes
- ✓ Facilidad de uso: la mayoría de los usuarios están de acuerdo, EA le permite comenzar rápidamente, con una interfaz de usuario enriquecida y la capacidad de crear modelos, plantillas, vistas de modelos y colecciones de "favoritos" de los elementos de uso común y diagramas

Además el EA provee un fuerte soporte para la administración de los proyectos, pues permite la estimación de la envergadura del mismo, medir el riesgo y el esfuerzo. También provee soporte para pruebas, control de cambios y mantenimiento.

1.5. Tecnologías de Base de Datos (BD)

La gestión de grandes volúmenes de datos se ha convertido en una necesidad en la actual sociedad, es necesario por tanto controlar la cantidad de información que se genera, especialmente en cuestiones de gestión empresarial, donde la capacidad humana de análisis y búsqueda no es lo suficientemente elevada para determinar cuánta y cuál información es absolutamente necesaria y al mismo

tiempo usarla en beneficio propio. Surgen así las bases de datos para almacenar, procesar y gestionar los datos de las organizaciones.

Una base de datos es un conjunto de datos almacenados y no redundante, organizados de forma independiente a su utilización o implantación en máquina, accesibles en tiempo real y compatibles con usuarios concurrentes; diseñada para satisfacer los requerimientos de información de una organización [23].

Históricamente las BD se han creado con el objetivo de poner a disposición de los usuarios un grupo de datos o información, que pueden ser utilizados en la toma de decisiones y la gestión de diversos procesos y que después de determinadas manipulaciones la información resultante fuera aún coherente y utilizable [24]. Para agregar, acceder y procesar los datos almacenados en una base de datos computacional, se necesita un sistema administrador de base de datos las cuales se desarrollan especialmente para manejar enormes cantidades de información en pocos segundos y juegan un rol central en el tratamiento de la información, ya sea como utilidad autónoma o parte de una aplicación.

1.5.1. Sistemas de Gestión de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, para garantizar la seguridad e integridad de los mismos [25].

Un SGBD debe permitir:

Definir una base de datos: especificar tipos, estructuras y restricciones de datos.

Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.

Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Existen diversas características, que al mismo tiempo se convierten en indicadores de la fiabilidad y capacidad de gestión de la información de un SGBD:

Independencia: consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Redundancia mínima: consiste en evitar la aparición de información repetida o redundante mediante un buen diseño de base de datos.

Seguridad: consiste en garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información.

Respaldo y recuperación: los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Los gestores de base de datos han evolucionado de manera considerable, facilitándose la manipulación de grandes volúmenes de datos y ahorra a los usuarios detalles acerca del almacenamiento. Ellos proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos [25].

MySQL

El gestor de base de datos utilizado para manipular la información fue MySQL, propiedad actual de Oracle. Este gestor se caracteriza por su amplio subconjunto del Lenguaje de Consultas Estructurado (SQL), y disponibilidad en gran cantidad de plataformas y sistemas. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen gran cantidad de librerías y otras herramientas que permiten su uso a través de

diferentes lenguajes de programación, además de su fácil instalación y configuración, fue diseñado con estas características, debido a que lo que se buscaba era un gestor de bases de datos con una gran rapidez de respuesta [26].

MySQL es un sistema de administración de bases de datos relacionales; una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos.

MySQL es *Open Source* porque se puede descargar el software de Internet y usarlo de forma gratuita. Se puede estudiar el código fuente y cambiarlo de acuerdo a las necesidades del usuario. Tomándose en cuenta lo anteriormente expuesto acerca de los diferentes SGBD, se decidió escoger MySQL por ser el manejador de bases de datos de código abierto más ajustado a los requerimientos que debe cumplir el sistema propuesto [27].

1.6. iReport

La herramienta *iReport* es un diseñador de informes visual, poderoso, intuitivo y fácil de usar para *JasperReport* escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, entre otros. Está integrado con *JFreeChart*, una de las bibliotecas gráficas de código abierto más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones *JDBC*

Esta herramienta está 100% escrito en Java y además es Opensource. Maneja el 98% de las etiquetas de *JasperReports*. Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, *subreports* (subreportes).

Este editor gráfico se utilizó para el diseño de los diferentes reportes junto a la librería *JasperReport* para interpretar los archivos generados en dicho editor en la

aplicación, ya que esta es una librería que permite generar informes. El funcionamiento se basa en escribir un xml donde se recogen las particularidades del informe que se necesita. Este xml lo tratan las clases del *Jasper* para obtener una salida. Esta salida puede ser un PDF, XML, HTML, CSV, XLS, RTF, TXT, etc. Lo que permite que el usuario obtenga un reporte de acuerdo a sus necesidades y lo visualice o imprima si desea en cualquier formato [7].

1.7. Tecnologías de Servidores Web

Un servidor *web* es un programa que sirve datos en forma de páginas *Web*, hipertextos o páginas HTML (*HyperText Markup Language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP. Con esto, un servidor *Web* se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un Navegador Web [28].

El servidor responde al cliente y envía el código HTML de la página; el navegador cuando recibe el código, lo interpreta y lo muestra en pantalla. El Cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor se encarga de transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. A continuación se tratan algunos de los servidores más utilizados:

1.7.1. Apache Tomcat

También conocido como simplemente *Tomcat* o *Jakarta Tomcat*, es un servidor web multiplataforma que funciona como contenedor de *servlets* y que se desarrolla bajo el proyecto denominado *Jackarta* perteneciente a la *Apache Software Foundation* bajo la licencia Apache 2.0 y que implementa las especificaciones de los *servlets* y de *Java Server Pages* o *JSP* de *SunMicrosystem*.

Dentro de las principales características de este servidor web según encontramos:

- ✓ Puede funcionar como servidor web por sí mismo en entornos de alto nivel de tráfico y alta disponibilidad.
- ✓ Es multiplataforma.
- ✓ Limpieza interna de código.
- ✓ Soporte para la inclusión de contenidos externos directamente en una aplicación web.
- ✓ Autenticación de acceso básico.
- ✓ Protocolo de Seguridad de Lenguaje de Marcado de Hipertexto (HTTPS, por sus siglas en inglés)
- ✓ Consola de administración.
- ✓ Negociación de credenciales.

Siendo Apache uno de los servidores más antiguos, es uno de los más utilizados hoy en día, por lo que es fácil hallar documentación sobre el mismo. Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos. Por las razones antes expuestas, es el servidor web escogido para el desarrollo de este PI.

1.8. Spring Security

Spring Security es un *framework* de apoyo al marco de trabajo *Spring*, que dota al mismo de una serie de servicios de seguridad aplicables para sistemas basados en la arquitectura basados en J2EE, enfocado particularmente sobre proyectos construidos usando *Spring Framework*. De esta dependencia, se minimiza la curva de aprendizaje.

Los procesos de seguridad están destinados principalmente, a comprobar la identidad del usuario mediante la autenticación que es el proceso de establecimiento de la identidad de que un usuario (persona, dispositivo, sistema, ente) sea quién dice ser. De igual forma, existen los permisos asociados al mismo mediante la autorización y se refiere al proceso de decisión sobre si un usuario identificado puede acceder a determinados recursos o realizar distintas acciones en una determinada aplicación. La autorización es dependiente de la autenticación ya que se produce posteriormente a su proceso [29].

Este *framework* posee modelos flexibles de autenticación permitiendo crear modelos propios siendo este proceso bastante simple, por lo que se integra fácilmente en la mayoría de sistemas.

A veces el simple proceso de autenticación no es suficiente y se necesita diferenciar la seguridad basada en la interacción del usuario con la aplicación, esto ayuda a proteger el proceso de recuperación de contraseñas por ataques de fuerza bruta, o simplemente para hacer más complicado a las personas duplicar el contenido de claves de la aplicación. Para conseguir esto Spring Security soporta "canales de seguridad" automáticos integrándose con *JCaptcha* para detección de usuarios humanos [29].

Independientemente de cómo fue realizada el proceso de autenticación, *Spring Security* proporciona un conjunto amplio de capacidades de autorización que podemos diferenciar en tres áreas:

- ✓ Autorización basada en solicitudes Web
- ✓ Autorización basada en métodos que pueden ser invocados
- ✓ Autorización de acceso a instancias de objetos que pertenecen a un dominio individual

- ✓ En la actualidad soporta integración de autenticación o autorización con tecnologías como *Spring*, *LDAP*, contenedor de integración con *JBoss*, *Jetty* y *Tomcat*

1.9. Conclusiones del Capítulo

1. A partir del análisis del proceso de gestión del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya” se detectaron deficiencias que impiden su desarrollo con la calidad requerida, lo que permitió determinar el problema científico planteado y el diseño de la presente investigación, ofreciendo como solución la realización de un sistema informático que favorece el desarrollo de este proceso.
2. Dada las características que debe cumplir el *software* y las bondades que ofrecen las tecnologías antes descritas se escogieron: *Groovy* como lenguaje de programación, el *framework* o marco de trabajo *Grails*, como SGBD MySQL, *Apache Tomcat* como servidor *web*, el EA como herramienta CASE para el modelado de diagramas, para la generación de informes la biblioteca *JasperReport* en conjunto con el *iReport* como diseñador de plantillas, el *framework* Spring Security que se encarga de manejar toda la seguridad y emplea las fases propuestas por la metodología de desarrollo ICONIX en la elaboración del sistema propuesto.

Capítulo 2: Descripción de la solución propuesta.

En este capítulo se desarrolla la ingeniería de software aplicada al desarrollo del Sistema para la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”, utilizando como guía la metodología de desarrollo ICONIX, la cual sustenta la organización de todo el proceso de desarrollo de la Aplicación al permitir la captura de los Requerimientos Funcionales (RF), y especificar los Requerimientos No Funcionales (RNF) para modelar las propiedades que el sistema requiere. Este proceso es una guía, que describe como ir desde los casos de uso hasta la codificación del software.

El capítulo incluye un análisis de sostenibilidad en las dimensiones administrativa, socio-humanista, ambiental y tecnológica, que tiene la aplicación como producto terminado en el marco de la institución. Además del criterio de expertos para evaluar la satisfacción con respecto al PI.

2.1 Definición y análisis de los requisitos

En esta fase del proceso se tiene como punto de partida el levantamiento de los requisitos realizados, como fruto de las entrevistas con el cliente, donde el mismo especifica las funcionalidades que necesita realice el software, así como un primer acercamiento al diagrama del modelo del dominio del sistema.

2.1.1 Análisis de requisitos

Los requisitos son las condiciones o capacidades que un sistema debe satisfacer, así como las especificidades de sus acciones. Este análisis se realiza para determinar si estos son confusos, incompletos, ambiguos, o contradictorios. Los requisitos se determinan a partir de la información brindada por el cliente acerca del sistema, son funcionalidades que debe realizar el software.

El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del problema a solucionar. Los requerimientos

pueden dividirse en Requerimientos Funcionales y Requerimientos No Funcionales.

Los requerimientos funcionales que se enumeran a continuación han sido agrupados en los paquetes definidos, que representan las funcionalidades de interés para el usuario final.

Requerimientos Funcionales

Los requerimientos funcionales detallan acciones que el Producto Informático (PI) debe ser capaz de realizar. Los requerimientos de este PI se agruparon en paquetes que se nombran según el proceso que se analiza como eje fundamental de esta investigación. Además se organizaron en un diagrama de paquetes y los diagramas de caso de uso del sistema.

A continuación se presenta una lista desglosada de estos requerimientos:

Paquete Seguridad.

1. Autenticar usuario.
2. Insertar usuario.
3. Eliminar usuario.
4. Modificar usuario.
5. Cerrar sesión.
6. Validar usuario.

Paquete Gestión del Trabajo Metodológico del departamento.

7. Insertar actividad metodológica planificada en sede central.
8. Modificar actividad metodológica planificada en sede central.

9. Eliminar actividad metodológica planificada en sede central.
10. Insertar actividad metodológica realizada en sede central.
11. Modificar actividad metodológica realizada en sede central.
12. Eliminar actividad metodológica realizada en sede central.
13. Insertar actividad metodológica planificada en municipio.
14. Modificar actividad metodológica planificada en municipio.
15. Eliminar actividad metodológica planificada en municipio.
16. Insertar actividad metodológica realizada en municipio.
17. Modificar actividad metodológica realizada en municipio.
18. Eliminar actividad metodológica realizada en municipio.
19. Insertar control a clase planificado por profesor.
20. Modificar control a clase planificado por profesor.
21. Eliminar control a clase planificado por profesor.
22. Insertar nota a control a clase por profesor.
23. Modificar nota a control a clase por profesor.
24. Eliminar nota a control a clase por profesor.
25. Visualizar control a clase por profesor.
26. Visualizar control a clase por departamento.
27. Visualizar control a clase por fecha.
28. Visualizar control a por nota.
29. Visualizar control a clase por período.

30. Visualizar control a clase por categoría docente.
31. Generar informe por período de las actividades metodológicas en la sede central.
32. Generar informe por período de las actividades metodológicas en los municipios.
33. Generar informe por período de los controles a clases por nota.
34. Generar informe por período de los controles a clases por profesor.
35. Generar informe por período de los controles a clases por fecha.
36. Generar informe por período de los controles a clases por categoría docente.
37. Generar informe por período de los controles a clases.
38. Visualizar actividades metodológicas planificadas en sede central.
39. Visualizar actividades metodológicas realizadas en sede central.
40. Visualizar actividades metodológicas planificadas en municipio.
41. Visualizar actividades metodológicas realizadas en municipio.
42. Visualizar actividades metodológicas no realizadas.
43. Insertar profesor.
44. Modificar profesor.
45. Eliminar profesor.

Paquete Gestión del Trabajo Metodológico de la facultad

46. Insertar actividad metodológica planificada en sede central por facultad.

- 47. Modificar actividad metodológica planificada en sede central por facultad.
- 48. Eliminar actividad metodológica planificada en sede central por facultad.
- 49. Insertar actividad metodológica realizada en sede central por facultad.
- 50. Modificar actividad metodológica realizada en sede central por facultad.
- 51. Eliminar actividad metodológica realizada en sede central por facultad.
- 52. Generar informe por período de las actividades metodológicas en la sede central por facultad.
- 53. Visualizar actividades metodológicas planificadas en sede central por facultad.
- 54. Visualizar actividades metodológicas realizadas en sede central por facultad.

Paquete Gestión de Codificadores

- 55. Insertar facultad.
- 56. Modificar facultad.
- 57. Eliminar facultad.
- 58. Insertar departamento de facultad.
- 59. Modificar departamento de facultad.
- 60. Eliminar departamento de facultad.
- 61. Insertar categoría docente.
- 62. Modificar categoría docente.
- 63. Eliminar categoría docente.

- 64. Insertar categoría científica
- 65. Modificar categoría científica
- 66. Eliminar categoría científica
- 67. Insertar tipo de actividad.
- 68. Modificar tipo de actividad.
- 69. Eliminar tipo de actividad.

2.1.2 Modelo del dominio

El modelo del dominio es la representación gráfica mediante un diagrama de clases de los principales procesos del negocio. Este modelo captura los objetos más importantes en el contexto del sistema y representa las cosas y los eventos que suceden en el entorno en el que se desarrolla la aplicación [20].

El proceso de *ICONIX* lo define como un artefacto colaborativo que representa el entendimiento actual del problema y que es actualizado constantemente a través del ciclo de vida del proyecto [30].

La elaboración del modelo del dominio constituye una parte esencial dentro de las etapas que componen la metodología de desarrollo *ICONIX*, pues refleja la comprensión actual del ámbito del problema.

Según [30] el modelo del dominio es un glosario de los términos (sustantivos y frases sustantivas) fundamentales usados en el proyecto, pero más que eso, es la representación gráfica de los mismos a través de las relaciones de agregación (tiene-un) y generalización (es-un). Además, define el alcance del proyecto y crea las bases para el desarrollo de los casos de uso que se abordarán posteriormente.

En la figura 5 se muestra el modelo del dominio del sistema informático que se propone.

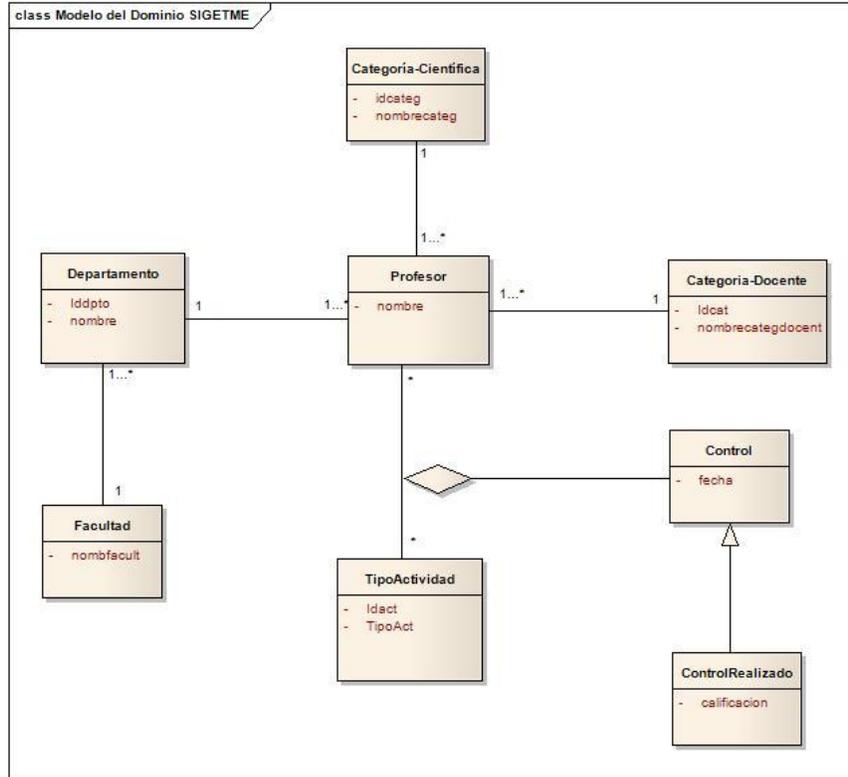


Figura 3: Modelo del dominio.

Fuente: Elaboración propia.

2.1.3. Prototipación rápida

La prototipación rápida en *ICONIX* se usa para simular el diseño del sistema. Se espera que los usuarios lo evalúen como si fuera el sistema final. Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El proceso se repite y finaliza cuando los usuarios están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando es evidente que no se obtendrá mayor beneficio con una iteración adicional [30].

2.2 Análisis y diseño preliminar

El diseño preliminar representa el punto medio que existe entre el análisis y el diseño el cual posibilita el completo entendimiento de los requerimientos, eliminando las imprecisiones de los mismos mediante el vínculo existente entre los casos de uso y los objetos que componen el modelo del dominio [19].

2.2.1. Modelos de casos de uso

Luego de la captura de los requerimientos funcionales, se realiza la modelación de los casos de uso, que comprende la identificación de los actores y los casos de uso involucrados en el sistema.

La metodología *ICONIX* como uno sus aspectos notables tiene que un caso de uso detalla un comportamiento a la vez que un requisito describe una regla para el comportamiento. Esta metodología asume que el modelo del dominio inicial no es correcto y provee un adecuado mejoramiento incremental del mismo a medida que se analizan los casos de uso.

Se tomó en consideración dividir en cuatro paquetes los modelos de casos de uso (ver Figura 4) de la aplicación propuesta debido al gran volumen de los mismos, por la relación existente entre ellos y haciendo uso del mecanismo que UML provee para dicha agrupación se establecieron en un diagrama de paquetes. La división se efectuó de la siguiente manera:

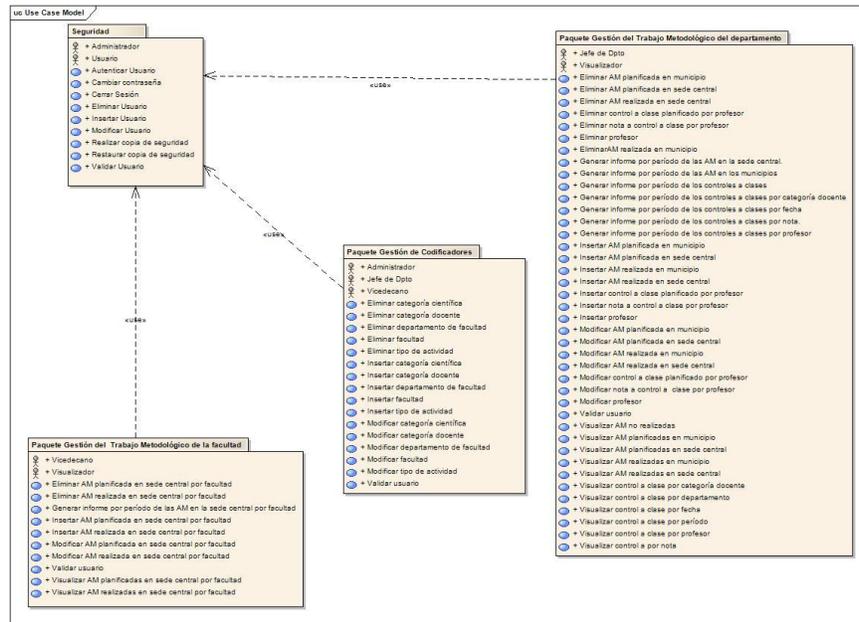


Figura 4: Diagrama de paquetes.

Fuente: Elaboración propia.

Seguidamente en la Figura 5 se muestra el diagrama de Caso de uso correspondiente al Paquete Gestión de Codificadores representado anteriormente. Los restantes diagramas de Trabajo Metodológico de Caso de uso se encuentran en los **Anexos** y en la **Documentación Digital Adjunta** la cual recoge los demás diagramas del proceso de desarrollo de *ICONIX*.

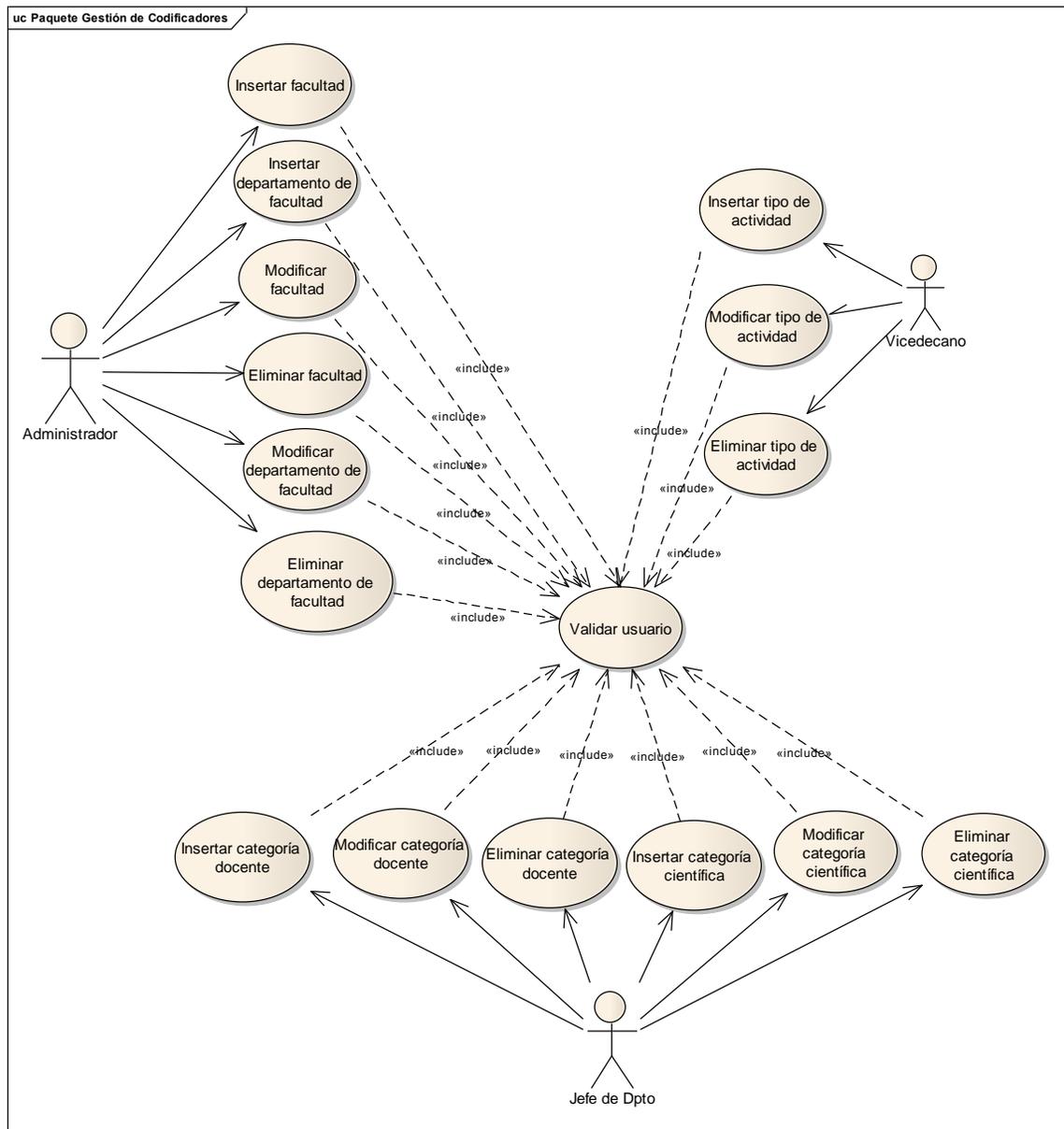


Figura 5: Diagrama de casos de uso del Paquete Gestión de Codificadores.

Fuente: Elaboración propia.

2.2.2. Actores del sistema

Autor	Descripción
Usuario	Persona que tiene acceso al sistema y puede realizar las operaciones de visualizar información sobre la aplicación, y efectuar registro.
Administrador	Persona encargada de realizar la gestión del todas las entidades y recursos del sistema.
Decano	Persona que se registra en el sistema que puede observar toda la información de la facultad.
Vicedecano	Persona registrada en el sistema que puede observar toda la información de la facultad. Puede añadir las actividades metodológicas a nivel de facultad (plan y real).
Jefe de Dpto	Persona que se registra en el sistema que puede observar toda la información de su departamento. Puede añadir las actividades metodológicas de su departamento, incluyendo controles a clases.

Tabla 1: *Actores del sistema.*

Fuente: Elaboración Propuesta

2.2.3. Descripción de casos de uso

En ICONIX, a diferencia de otras metodologías, como RUP, la descripción de los casos de uso, se escribe en voz activa, es decir, se redactan de forma tal que se

capturen las acciones del usuario y las respuestas del sistema. En la tabla 1, se muestra la descripción textual del caso de uso “Insertar actividad metodológica planificada en la sede central por facultad”. El resto de las descripciones por cada caso de uso de los diagramas correspondientes, se encuentran en el fichero EAP, el cual recoge todas las descripciones de los casos de uso y demás diagramas del proceso de desarrollo de ICONIX.

Caso de Uso	Insertar actividad metodológica planificada en la sede central por facultad.
Curso Básico	El Vicedecano hace clic en la opción “Actividades Metodológicas” que brinda la interfaz gráfica, para insertar una actividad planificada en la sede central a nivel de facultad. El sistema verifica los permisos y muestra el formulario “Gestionar Actividades”. El Vicedecano llena los campos pertenecientes al tipo de actividad, fecha de la misma, lugar de realización, y nombre de la persona encargada de supervisarla así como de a quién va dirigida; al finalizar realiza un clic en el botón “Insertar Actividad”. El sistema chequea la información y actualiza el listado de actividades planificadas en la base de datos.
Curso Alterno	El sistema muestra un mensaje de error informando que existen espacios en blanco o que algunos de los campos de entrada tienen un formato incorrecto.

	<p>El sistema verifica que el usuario no tiene permiso para realizar esta acción.</p> <p>No se inserta la actividad.</p>
--	--

Tabla 2: Insertar actividad metodológica planificada en la sede central por facultad.

Fuente: Elaboración Propia

2.2.4. Análisis de robustez

Para obtener el diseño del sistema se “convirtieron” los casos de uso en relaciones entre objetos. La técnica que propone la metodología ICONIX es el **análisis de robustez**, puente que une la descripción de los casos de uso y el diseño. Además, los diagramas de robustez son la vía para identificar el primer conjunto de objetos de cada caso de uso.

Según plantea que la descripción de los casos de uso y los diagramas de robustez se deben acoplar correctamente. Por ello, el texto de la descripción de los casos de uso y los objetos del diagrama de robustez deben tener una relación 1:1, es decir, las frases sustantivas se convierten en entidades e interfaces, mientras que los verbos se convierten en controladores de estos objetos.

A continuación se ejemplifica el diagrama de robustez perteneciente al caso de uso “Insertar actividad metodológica planificada en la sede central por facultad” (Ver figura 6) descrito en la tabla 2. Los restantes diagramas de robustez se encuentran en los **Anexos** y en la **Documentación Digital Adjunta**.

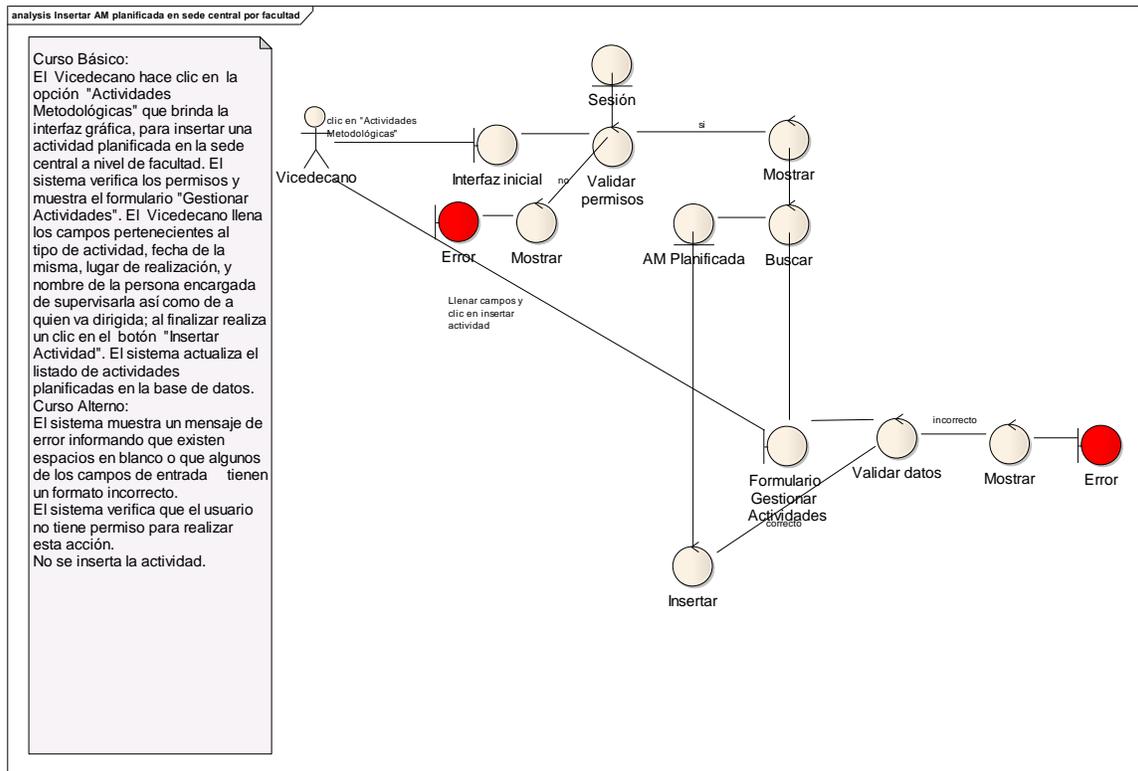


Figura 6: Diagrama de robustez. Caso de Uso: Insertar actividad metodológica planificada en la sede central por facultad.

Fuente: Elaboración propia

Concluida esta fase se actualizó el modelo del dominio, en el cual se incluyeron nuevas clases y se le asignaron los atributos a las mismas.

2.2.5. Arquitectura técnica

La arquitectura técnica define un conjunto de decisiones básicas que se necesitan tomar, en cuanto a las tecnologías utilizadas en el desarrollo. Se realiza con el objetivo de satisfacer los requerimientos del negocio, incluye aspectos como la topología del sistema que se está desarrollando, o sea, localización física en la red, la elección del servidor de aplicación [31].

La arquitectura técnica es conocida como la arquitectura del sistema y de software, con el propósito de describir el sistema que se construirá en términos de estructura,

satisfacer los requerimientos del negocio y el nivel de servicio del sistema que se va a desarrollar, incluyendo además la topología del sistema, es decir los nodos del servidor, localización física en la red, la elección del servidor de aplicación.

Como se muestra en la (Figura 7) la arquitectura está dividida por capas, como establece el framework Grails.

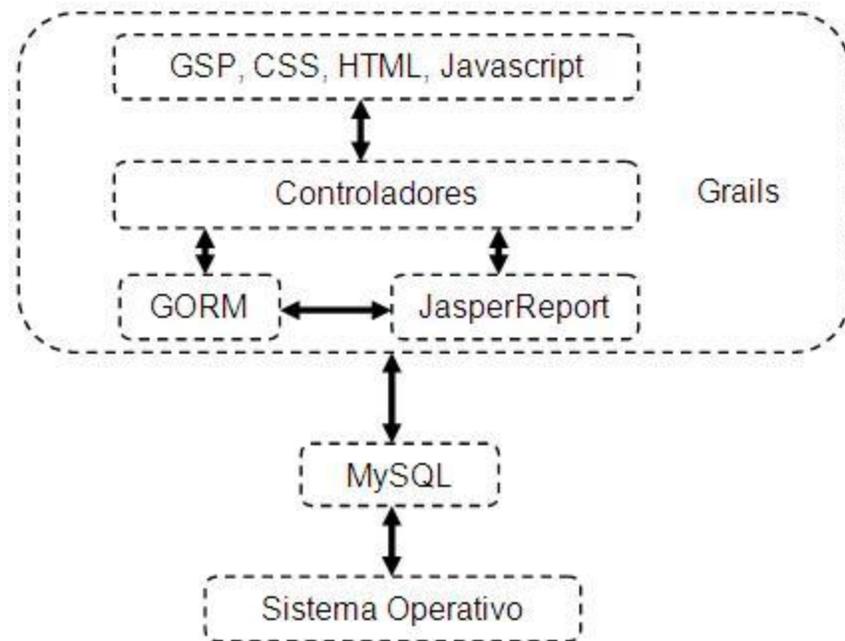


Figura 7: *Arquitectura técnica del Sistema.*

Fuente: [16]

Para la capa de presentación se emplearon los lenguajes CSS, HTML, Javascript y GSP (Groovy Server Pages), las peticiones realizadas por el cliente, así como las respuestas generadas por el servidor, son administradas y gestionadas por clases controladoras.

En la capa de persistencia de los datos se empleó el framework propuesto por Grails: el framework GORM, en ese mismo nivel se ubica el API para la generación de informes Jasper Report.

Más abajo MySQL como gestor de BD y el sistema operativo que se desee emplear.

Dentro de la arquitectura técnica se encuentran además:

- ✓ Captura de requerimientos no funcionales.
- ✓ Realización del modelo de despliegue.

2.2.5.1 Captura de requerimientos no funcionales

Los requerimientos no funcionales son cualidades que el producto terminado debe cumplir, se caracterizan por hacer al producto atractivo, usable, rápido y confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Estos se deben ajustar a la arquitectura técnica que adopta el sistema en desarrollo.

Apariencia o Interfaz externa

- ✓ El diseño debe ser agradable y atractivo a los usuarios para lograr una mejor concentración, sin desviar demasiado su atención del contenido de trabajo
- ✓ La interfaz debe aprovechar el espacio de trabajo para proporcionar una adecuada distribución de los elementos
- ✓ Los colores deben encontrarse en la gama del negro, gris, y blanco, pero con tonalidades suaves y relajantes para evitar mucho esfuerzo visual.
- ✓ La interfaz no debe recargarse con imágenes para proporcionar una navegación cómoda.
- ✓ El vocabulario utilizado debe ser en el idioma español.

Usabilidad

- ✓ Funcionalidades visibles en todo momento que faciliten la navegación.
- ✓ Elementos organizativos para las funcionalidades como botones.
- ✓ El sistema posibilitará el acceso a otras páginas sin mucho esfuerzo ni consumo de recursos tanto humanos como computacionales
- ✓ Se muestran mensajes de error ante la imposibilidad de ejecutar una acción.

Ayuda en línea

- ✓ Se confeccionará un Manual de Usuario que servirá al personal encargado de la gestión de la información.

Rendimiento

- ✓ Precisión alta en los resultados.
- ✓ Reducción de los tiempos de respuestas y alta velocidad de procesamiento de la información a través de peticiones asincrónicas al servidor.

Portabilidad

- ✓ El sistema puede ser usado desde la máquina computadora-cliente y bajo la plataforma Windows o Linux
- ✓ La máquina computadora Servidor puede funcionar bajo la plataforma Windows o Linux

Seguridad

- ✓ Sólo los usuarios autorizados podrán acceder al sistema
- ✓ Garantizar que las funcionalidades del sistema se realicen de acuerdo con el nivel del usuario que esté activo

- ✓ No existe información que se pueda obtener sin ser usuario del sistema
- ✓ Sólo el administrador del sistema tendrá acceso a la BD, a los ficheros fuentes del sistema y es responsable de la autorización en general del mismo
- ✓ El sistema debe tener protección contra acciones no autorizadas así como evitar la corrupción y estados inconscientes que puedan afectar la integridad de la información almacenada

Políticos-culturales

- ✓ Se rige bajo las resoluciones estipuladas por el Ministerio de Educación Superior.

Software

- ✓ El Sistema Operativo (SO) de la máquina computadora cliente debe ser Windows o superior
- ✓ La máquina computadora servidor debe tener Windows 2000 o superior, *Servidor Web Apache Tomcat, MySQL y Java Runtime Enviroment (JRE)*

Hardware

- ✓ El servidor debe tener como mínimo un procesador Pentium III a 1 GHz, con 512 MB de memoria RAM, una tarjeta de red Protocolo Ethernet 10/100 MB/s.
- ✓ La máquina computadora servidor y las computadoras clientes deben estar conectadas a la red

2.2.5.2 Modelo de Despliegue

El modelo de despliegue es un diagrama que se usa para representar las arquitecturas de la aplicación. El modelo de despliegue del sistema (ver Figura 8) especifica la distribución desde el punto de vista de *hardware* y *software* que se espera tenga la implantación del sistema, se utilizó para describir la topología de la aplicación propuesta, mostrando la configuración los dispositivos que participan en la ejecución.

Este modelo, se usó como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

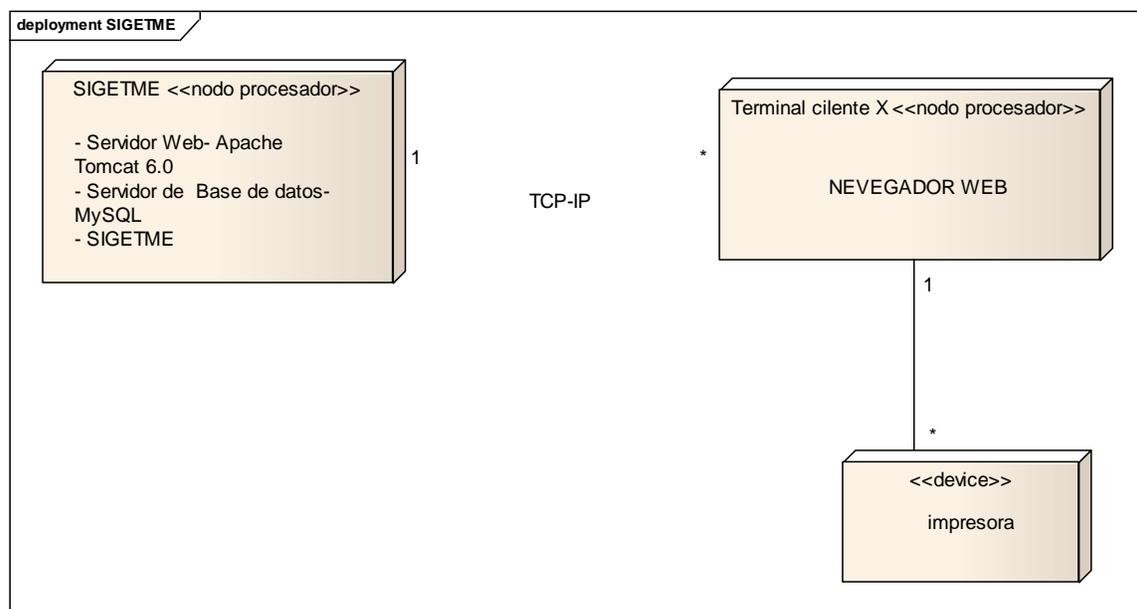


Figura 8: Diagrama del modelo de despliegue.

Fuente: Elaboración propia

2.3 Diseño

El diseño de la aplicación es uno de los pasos fundamentales que propone la metodología ICONIX. Este ocurre luego de terminado el análisis y captura de los

requisitos. En el diseño, se crean los diagramas de secuencia en los cuales se detalla cómo y a través de qué mensajes interactúan las clases. Además, se crean los diagramas de clase del diseño, en los que se definen las clases a codificar con sus atributos, métodos y relaciones con otras clases.

2.3.1 Diagramas de Secuencia

Los diagramas de secuencia muestran la forma en la que el código debe funcionar. En este estado de la construcción del sistema, se colocaron los mensajes en las clases pertinentes. Un aspecto importante que tratan estos diagramas es el descubrimiento de nuevas clases que deben ser incluidas en el modelo del dominio. Por ejemplo, cuando se elaboran los diagramas de robustez surgen los objetos de interfaz y entidades, los que se relacionan a través de controladores que en ese momento son funciones o mensajes que se pasan estos objetos, pero que todavía no están asociados a ninguna clase. En los diagramas de secuencia se incluyeron las clases controladoras y clases de otros frameworks, como las de Grails, que deben realizar la lógica del negocio de la aplicación.

Estos diagramas tienen tres objetivos fundamentales:

- Asignar comportamiento a las clases
- Mostrar en detalle la interacción entre las clases
- Concluir la distribución de operaciones entre las clases

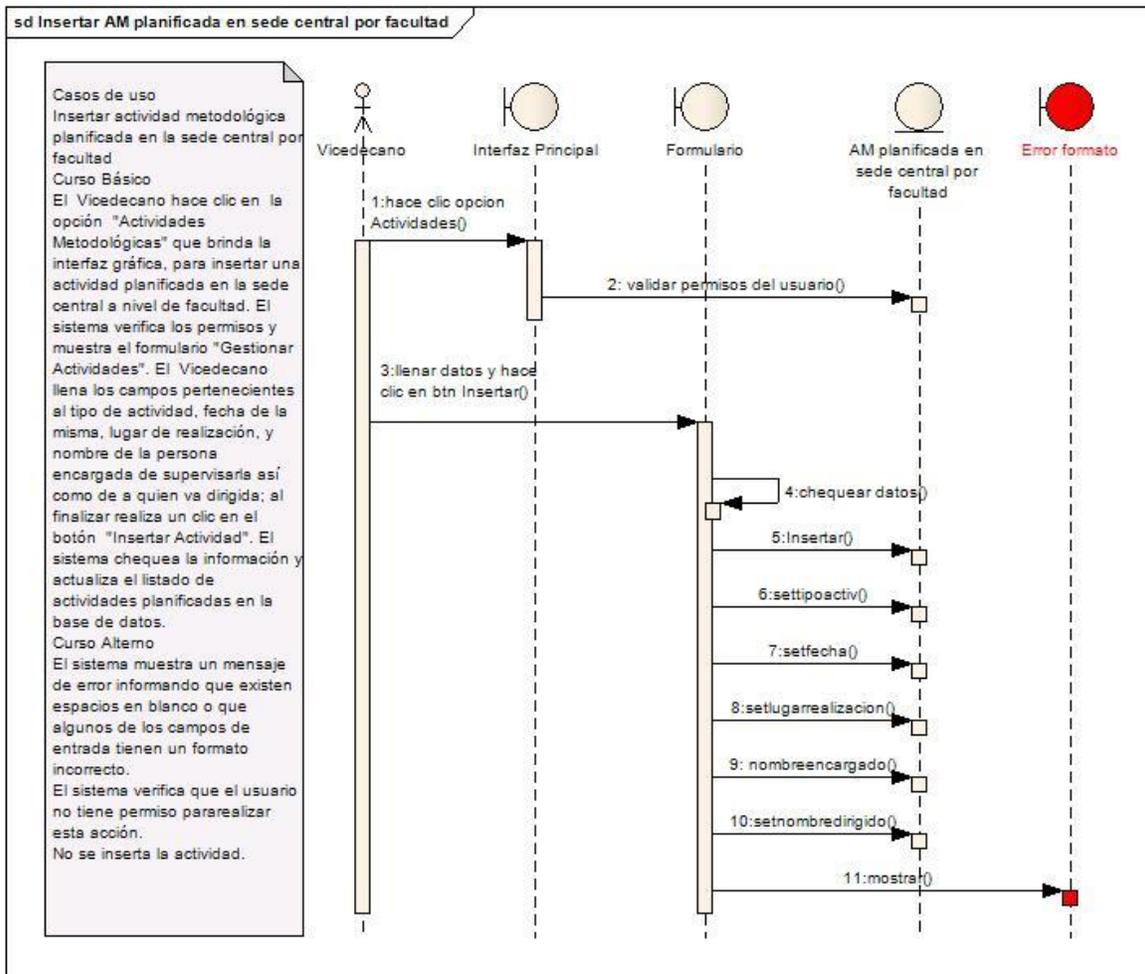


Figura 9: Diagrama de secuencia Caso de uso: Insertar actividad metodológica planificada en sede central por facultad.

Fuente: Elaboración propia

2.3.2 Diagrama de Clases

El diagrama de clases representa el modelo del dominio actualizado. En él, los objetos del dominio se convierten en clases, las cuales contienen los métodos y atributos de estos objetos.

A continuación en la Figura 10 se muestra el diagrama de clases desarrollado para el sistema de la presente investigación.

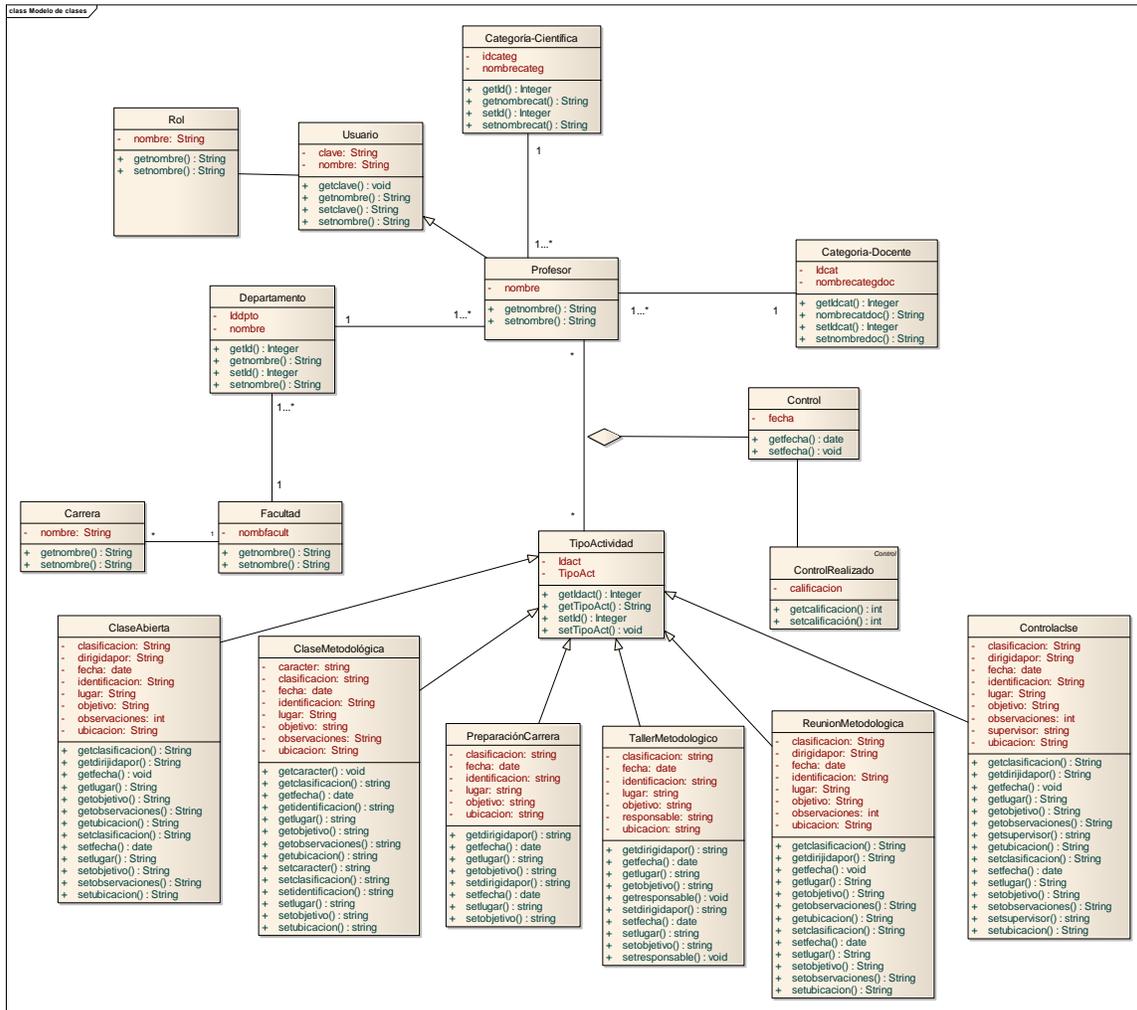


Figura 10: Diagrama de Clases

Fuente: Elaboración propia

2.3.3 Modelo de datos

El modelo de datos define las estructuras de datos que se utilizan en el sistema y describe la estructura lógica de la información que queda almacenada en la base de dato [12]. Para determinar el modelo de datos se analizaron cuáles eran las clases candidatas a convertirse en persistentes en el diagrama de clases del epígrafe anterior.

A continuación en la Figura 11 se muestra el modelo de datos utilizado en aplicación desarrollada.

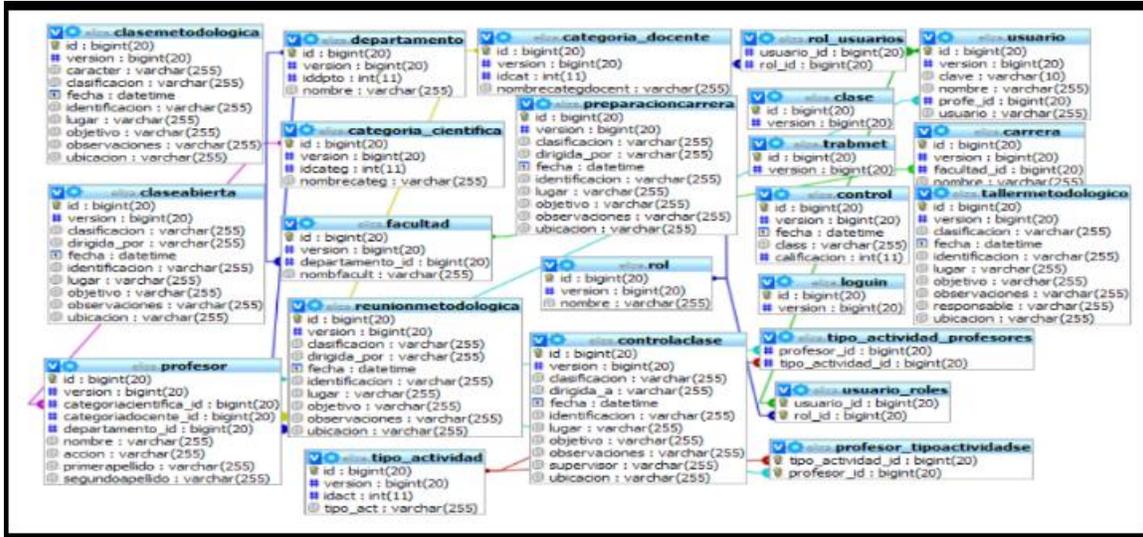


Figura 11: Modelo de datos

Fuente: Elaboración propia

2.4 Implementación

La fase de implementación se realizó para concretar el diseño y análisis de los requerimientos planteados en el problema y los que surgieron durante las fases anteriores. Se generó el código fuente necesario para el funcionamiento correcto de la aplicación.

2.4.1 Estándares de codificación

Para garantizar una implementación uniforme del *software*, así como facilitar el mantenimiento y la reutilización se crean los estándares de codificación que aseguren la legibilidad del código.

Tabuladores o espacios: nunca se mezclaron tabuladores y espacios para evitar confusiones.

Tamaño máximo de línea: se limitan todas las líneas a un máximo de 100 caracteres. Para cadenas de texto largas (cadenas de documentación o comentarios), es aconsejable limitar a 100 caracteres.

Líneas en blanco: se usaron líneas en blanco antes y después de:

- La declaración de una estructura o una clase
- La implementación del método de una clase
- Comentarios no relacionados con el código
- Bloques de códigos complejos.

Espacios en blanco en expresiones y sentencias: se evitaron espacios en blanco extra inmediatamente después de entrar en un paréntesis o antes de salir de un paréntesis, corchete o llave. Inmediatamente antes de una coma, punto y coma, o dos puntos. Inmediatamente antes de abrir un paréntesis para una lista de argumentos de una llamada. Más de un espacio alrededor de un operador de asignación (u otro operador) para alinearlos con otro.

Comentarios: los comentarios están escritos en español, siendo frases completas. Si un comentario es una frase o sentencia, la primera palabra está en mayúsculas, a menos que sea un identificador que comience con una letra en minúsculas. Si un comentario es corto, se puede omitir el punto al final. Los comentarios de bloque generalmente consisten en uno o más párrafos contruidos con frases completas, y cada frase termina con un punto.

Nombres a Evitar: nunca se utilizan los caracteres 'l' (letra ele minúscula), 'O' (letra o mayúscula), o 'I' (letra i mayúscula) como nombres de variables de un solo carácter. En algunas fuentes, estos caracteres son indistinguibles de los numerales uno y cero. Cuando se está tentado de usar 'l', se utiliza una 'L' en su lugar.

Nombre para Clases e Interfaces: las letras de la palabra están en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no exceda los 20 caracteres. Se puede utilizar guiones bajos en el nombre del sistema si mejora la legibilidad.

Nombres de Paquetes: las letras de la palabra están en Notación Camell, se deben emplear nombres cortos formados en su totalidad por letras minúsculas. Se puede utilizar guiones bajos en los nombres de los paquetes si mejora la legibilidad.

Nombre para Métodos: las letras de la palabra están en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no exceda los 30 caracteres.

Argumentos de métodos: es necesario que el nombre indique propósito, que contenga letras y que no se demasiado largo.

2.4.2 Valoración de sostenibilidad

La valoración de sostenibilidad se define como un proceso de evaluación de impactos ambientales, socio humanistas, administrativos y tecnológicos de un producto informático, previsibles desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente.

Con el propósito de favorecer la gestión del trabajo metodológico que se realiza en la Universidad de Holguín “Oscar Lucero Moya” con el sistema informático como solución propuesta al problema planteado, se tuvo en cuenta que el mismo fuera sostenible desde el punto de vista: administrativo, socio-humanista, ambiental y tecnológico.

2.4.2.1. Dimensión Administrativa

Esta dimensión se sustenta en el ahorro, gasto, calidad de la producción y los servicios, administración de recursos y en la toma de decisiones administrativas que brinda como resultado el sistema informático.

La estimación por Puntos de Caso de Uso es el método empleado para estipular el esfuerzo que requiere el desarrollo del software, mediante una aproximación a los primeros casos de usos desarrollados. Este resulta efectivo para este proyecto pues la metodología que sustenta su desarrollo está basada en iteraciones continuas donde los primeros casos de uso son los que permiten diseñar la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos. Este método ayuda a estimar el tiempo de desarrollo de un PI mediante la asignación de "pesos" a cierto número de factores que este plantea tener en cuenta, para posteriormente, contabilizar el tiempo total estimado para el PI a partir de dichos factores.

El paso inicial para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar (UUCP). Este valor se calcula a partir del Factor de Peso de los Actores sin ajustar (UAW) y el Factor de Peso de los Casos de Uso sin ajustar (UUCW).

Estos valores se calculan mediante un análisis de la cantidad de actores y casos de uso presentes en el sistema y la complejidad de cada uno de ellos respectivamente.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe calcular los Puntos de Casos de Uso ajustados (UCP) teniendo en cuenta los Puntos de Casos de Uso sin ajustar (UUCP), el Factor de complejidad técnica (TCF) y el Factor de complejidad ambiental (ECF).

El TCF se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema, como se muestra en la tabla.

Factor	Descripción	Peso	Valor	TCF
TCF01	Sistema Distribuido	2.00	0.00	0.00
TCF02	Objetivos de performance o tiempo de respuesta	1.00	1.00	3.00
TCF03	Eficiencia del usuario final	1.00	1.00	3.00
TCF04	Procesamiento interno complejo	1.00	1.00	1.00
TCF05	El código debe ser reutilizable	1.00	4.00	1.00
TCF06	Facilidad de instalación	0.50	4.00	1.00
TCF07	Facilidad de uso	0.50	5.00	3.00
TCF08	Portabilidad	2.00	0.00	1.00
TCF09	Facilidad de cambio	1.00	3.00	2.00
TCF10	Concurrencia	1.00	2.00	1.00
TCF11	Incluye objetivos especiales de seguridad	1.00	4.00	3.00
TCF12	Provee acceso directo a terceras partes	1.00	5.00	3.00

TCF13	Se requieren facilidades especiales de entrenamiento a usuarios	1.00	1.00	1.00
			Total:	22.00

Tabla 3: Factores de Complejidad técnica.

El Factor de complejidad técnica se calcula de la siguiente manera:

Factor	Valor
Valor desajustado de TCF (UTV)	22.00
Pesos TCF (TWF)	0.01
Constante TCF (TC)	0.60
TCF = TC + (UTV * TWF)	0.82

Tabla 4: Cálculo del TCF.

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del ECF. En la siguiente tabla se muestra los factores que se tienen en cuenta en el cálculo del mismo.

Factor	Descripción	Peso	Valor	TCF
ECF01	Familiaridad con el modelo de proyecto utilizado	1.50	4.00	6.00
ECF02	Experiencia en la aplicación	0.50	3.00	1.50
ECF03	Experiencia en orientación a objetos	1.00	5.00	5.00
ECF04	Capacidad del analista líder	0.50	3.00	1.50
ECF05	Motivación	1.00	5.00	5.00
ECF06	Estabilidad de los requerimientos	2.00	4.00	8.00
ECF07	Personal parte de tiempo	-1.00	0.00	0.00
ECF08	Dificultad del lenguaje de programación	-1.00	0.00	0.00
			Total:	27.00

Tabla 5: Factor ambiental.

El Factor de complejidad ambiental resulta de la siguiente manera:

Factor	Valor
--------	-------

Valor desajustado ECF (UEV)	27.00
Peso ECF (EWF)	-0.03
Constante ECF (EC)	1.40
ECF = EC + (UEV * EWF)	0.59

Tabla 6: Cálculo del ECF.

Para determinar la cantidad de hombres por caso de uso se contabilizan cuántos factores de los que afectan al ECF están por debajo del valor medio (3), para los factores ECF1 a ECF6. Luego se contabilizan cuántos factores de los que afectan al ECF están por encima del valor medio (3), para los factores ECF7 y ECF8. Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

Una vez calculado todos los factores que se tienen en cuenta para el Análisis de Puntos de Casos de Uso se obtuvieron los resultados siguientes:

Pasos	Valor
Total de Casos de Uso	72
UUCP	375.00
TCF	0.82
ECF	0.59

UCP = (UUCP * TCF * ECF)	181.00
CF	20.00
Esfuerzo en hora-hombres (E = CF * UCP)	3620.00
Costo Total	7240.00

Tabla 7: Estimación del esfuerzo y costo.

Analizando los resultados mostrados en la tabla 6 se puede concluir que se emplearan 20 horas-hombres para realizar un caso de uso, obteniéndose un esfuerzo de 3620 horas-hombres (15 meses aproximadamente), el cálculo real es de 724 horas-hombres, disminuye en un 20% por uso de herramientas visuales, para un costo total de 7240 pesos o 182.00 cuc. Este valor está dado fundamentalmente por el tiempo dedicado al mismo el cual disminuyó considerablemente como resultado del empleo del framework Grails el cual aporta un ciclo de desarrollo muy ágil.

El gasto en recursos informáticos es mínimo pues la infraestructura que sirve de soporte al proyecto ya estaba creada. El sistema más que minimizar la mano de obra la optimiza, pues agiliza todo el proceso de control por parte de los directivos lo cual le permitirá al personal involucrado disponer de más tiempo para realizar otras tareas además aporta mejoras significativas para la gestión del trabajo metodológico en la universidad, disminuyendo el trabajo manual de los trabajadores y favoreciendo el flujo de trabajo de los usuarios que interactúan con el software. Proporciona además una disminución considerable en los tiempos de búsqueda y garantiza la perdurabilidad de información valiosa. No se incurren en gastos de electricidad asociado al uso de las computadoras y el servidor porque esta situación ya se presentaba. El sistema cuenta con la funcionalidad de generar reportes por períodos lo cual favorece la toma de

decisiones por parte de los directivos de la entidad, pues en los mismos se centraliza información a partir de la cual resulta más fácil arribar a conclusiones.

Las herramientas utilizadas para el desarrollo del software en su totalidad son libres, con lo que se absuelve a la Universidad de Holguín “Oscar Lucero Moya” de incurrir en gastos para llevar a cabo el desarrollo del proyecto.

A partir de lo analizado y los beneficios que proporciona el software, se arribó a la conclusión que éste es sostenible desde la dimensión administrativa.

2.4.2.2. Dimensión Socio-Humanista

Esta dimensión se sustente en el modo de vida, desarrollo de un grupo social, satisfacción de necesidades sociales, formación ético humanista de los gestores del sistema informático, así como en la ciencia y la tecnología como procesos sociales.

Con vista a garantizar la calidad del proceso se consolida el factor humano. Lograr la comodidad de los trabajadores y que estos realicen sus labores con eficacia, seguridad, y rapidez, constituye uno de los principales motivos de implantar soluciones factibles en las empresas actuales. Por ello, en muchos casos surge la necesidad de informatizar los procesos, para optimizar las labores y obtener buenos resultados en el menor tiempo posible.

El sistema informático propuesto dará solución a la situación problemática existente en la Universidad de Holguín “Oscar Lucero Moya” y brindará facilidades de operación y uso para resolver tareas simples que permitan llevar a cabo el control del proceso referente al trabajo metodológico en dicho centro, logrando optimizar un proceso que hasta el momento se hace de forma manual. El sistema informático ahorra tiempo y esfuerzo a los trabajadores de la entidad, brindando facilidades de ejecución para llevar a cabo el control del proceso, favoreciendo a todo el personal de la empresa que de una u otra forma interactúa en cada uno de los procesos que lo componen, de manera que se

sientan satisfechos, ya que contarán con acceso a información segura e íntegra y con formas prácticas de tratarla.

Estas mejoras en las condiciones de trabajo lograrán saciar necesidades propias del personal implicado y garantizará como elemento de primer orden la calidad en el control del proceso. El sistema no generará empleo ni desempleo en la organización, su influencia está orientada, por el contrario, a favorecer el trabajo de las personas que ya son empleadas por la empresa. Una vez implantado el sistema las consultas serán más rápidas, la información será más precisa y organizada, se podrán realizar salvadas de información y las posibilidades de actualización serán mucho mayores. Para evitar rechazo por parte de los usuarios ante el cambio, el sistema se implantará de forma tal que el personal se pueda ir adaptando al mismo fácilmente, se confeccionará un manual de usuario donde se expliquen las características, funcionalidades y ventajas del sistema, y de ser necesario, se impartirán cursos para adiestrar al personal que lo utilizará, logrando así una mayor aceptación.

El sistema puede ser generalizado, ya que la necesidad que hace que surja el mismo no es sólo de la Universidad de Holguín, sino de cualquier institución análoga del país. Por tales razones, dentro de las concepciones iniciales del sistema se pensó en su flexibilidad, versatilidad y las generalidades, pero también las posibles particularidades que pueden tener las organizaciones de este ámbito o dominio.

A partir de lo analizado y plasmado con anterioridad se arribó a la conclusión que el sistema desarrollado es sostenible desde la dimensión socio-humanista.

2.4.2.3. Dimensión Ambiental

Esta dimensión se sustenta en determinar las condiciones favorables o no a las personas o cosas, minimizando daños e impactos negativos al medio ambiente.

En la actualidad, luchar por el cuidado y preservación del medio ambiente constituye una de las principales causas de lucha de cada nación. Cada vez más, el derroche excesivo se incrementa a partir de soluciones no razonables ambientalmente. Por eso toda solución de un problema no puede repercutir en daños y alteraciones del medio ambiente.

Al desarrollar el sistema, se tuvo en cuenta utilizar colores claros y agradables a la vista, todo en la gama de marrones concebidos como colores corporativos de la entidad, manteniendo una interfaz entendible, amigable y funcional haciendo uso de íconos que indican al usuario la función que realizan, además la posición simétrica de los elementos en pantalla brinda sensación de equilibrio al usuario final. Todo esto permite lograr una mayor identificación del usuario con la aplicación y favorecer la navegación e intercomunicación, así como evitar estados de estrés de los usuarios del sistema.

No se contribuye de modo alguno al deterioro del medio ambiente tras desarrollar el sistema, ya sea por causa de contaminación por ruido, interferencia, u otros factores, sino todo lo contrario, su implantación ayuda en gran medida al cuidado del mismo pues la explotación del sistema contribuye además a una disminución considerable del uso de materiales de oficina tales como papel, cartón, y tinta contribuyendo a preservar el entorno donde opera la organización con el fin de reducir los impactos ambientales. Por otra parte, se tuvo en cuenta las exigencias fisiológicas del ser humano al seleccionar el tamaño de letra, y tipografía para la cómoda visualización de los datos.

El sistema facilita el acceso rápido a la información por lo que no será extenso el tiempo necesario de intercambio con el usuario, lo que provoca que no existan riesgos de daños a la postura de los implicados, efectos provocados por un tiempo prolongado de trabajo sin las condiciones óptimas (correcta iluminación, protectores de pantalla, asientos cómodos y con medidas correctas, ubicación correcta de la computadora).

A partir de lo analizado y plasmado anteriormente, se arribó a la conclusión que el sistema es sostenible desde la dimensión ambiental.

2.4.2.4. Dimensión Tecnológica

Esta dimensión se sustenta en el uso de tecnología adecuada y asimilable con el usuario de acuerdo a las necesidades existentes.

Debido a que los trabajadores del centro, en su mayoría, al realizar su labor diaria hacen uso de computadoras, no necesitan ser capacitados pues ya cuentan con los conocimientos y habilidades necesarias, además el sistema cuenta con un manual de usuario que facilita el entendimiento y manipulación del software. En cuanto a la infraestructura informática, la Universidad de Holguín “Oscar Lucero Moya” cuenta con los recursos necesarios para la explotación del sistema, pues consta de una red principal que brinda servicios tan utilizados como correo electrónico (e-mail), Internet, transferencia de ficheros (FTP), resolución de nombres de dominio (DNS), publicación de páginas web y que fundamentalmente comunica a la universidad con el resto del mundo, posibilitando el acceso al sistema desde la web a los usuarios del sistema, además tiene los principales enrutadores, *switchs*, *módems* y servidores necesarios para establecer la comunicación. Las computadoras utilizadas por los usuarios del sistema cuentan con las prestaciones requeridas para usar el sistema y se encuentran conectadas a la red de la universidad. El sistema fue modelado según el patrón de diseño Modelo Vista Controlador por lo que garantiza la separación de capas, así su interfaz gráfica se muestra independiente a la lógica del negocio y se hace posible variar el flujo de información entre los elementos que actúan en dependencia de las necesidades de la organización donde se utilice. La flexibilidad de la tecnología utilizada permite que el sistema perdure en el tiempo garantizando su evolución pues permite agregarle mas funcionalidades e incorporarle librerías, *plugins* y componentes que se utilicen para optimizar el trabajo, además el código se

encuentra comentado de forma tal que realizar cualquier cambio sea mucho más fácil en el futuro.

A partir de lo analizado y plasmado anteriormente, se arribó a la conclusión que el sistema es sostenible desde la dimensión tecnológica.

2.5 Conclusiones de la VSPI

La valoración de sostenibilidad de la solución propuesta reconoce al sistema propuesto como un producto sostenible desde sus cuatro dimensiones. Desde su concepción, el diseño del sistema informático se basó fundamentalmente en respetar el medio ambiente, la tecnología utilizada fue *software* libre, y es asequible a todo tipo de usuarios, no se requieren nuevas inversiones para la implementación del sistema, además se dispone del personal profesional calificado para el mantenimiento del sistema.

2.6 Valoración de los resultados obtenidos

Una vez concluida la implementación del sistema propuesto, se procedió a la realización de encuestas con el objetivo de evaluar la satisfacción de los usuarios respecto al sistema, así como la opinión de especialistas en el tema. Los especialistas seleccionados fueron los jefes de departamentos, jefes de disciplinas y vicedecanos de la Facultad de Informática y Matemática.

La encuesta aplicada (Ver Anexo VI) solicitaba la opinión del usuario, sobre cuestiones esenciales referentes al sistema informático realizado, una vez obtenidos los resultados de respuestas, en cada una de las preguntas por cada uno de los encuestados, se llevó a cabo el cálculo del promedio de la cantidad de preguntas por evaluación dada. Posteriormente se realizó el cálculo del porcentaje para saber que representa ese dato numérico con respecto al total de encuestados, obteniéndose como resultado que el 46% de los encuestados opinan que el sistema para la gestión de información del trabajo metodológico en La Universidad de Holguín "Oscar Lucero Moya" es muy adecuado, el 32% opina

que es bastante adecuado, el 22% opina que es adecuado y el 0% opina que es poco adecuado y nada adecuado, definiendo así el grado de aceptación de los usuarios.

2.7 Conclusiones del capítulo

1. La utilización de la metodología ICONIX para modelar y construir el sistema propuesto, permitió resolver una gran parte de las exigencias, logrando realizar eficientemente el diseño del problema planteado en la investigación.
2. A través de la valoración de sostenibilidad en sus cuatro dimensiones se concluyó que la solución propuesta es factible y el sistema es sostenible y perdurable en el tiempo, respondiendo así a la necesidad que existe de una herramienta informática que permita gestionar de forma rápida y fiable el flujo de información referente a la gestión del trabajo metodológico.
3. La valoración de la satisfacción de los usuarios respecto al sistema, realizada a través de encuestas a especialistas, arrojó como resultado que el producto informático implementado satisface las necesidades que lo originaron.

Conclusiones

El “Sistema para la gestión de información del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”, es una nueva herramienta con la que los usuarios pueden contar para gestionar las actividades que desarrollan los profesores como parte de su preparación docente y como parte esencial del proceso formativo.

Con este sistema fue posible unificar la información necesaria para que el personal encargado de trabajar con la información referente al trabajo metodológico pueda gestionar la información de manera más eficiente y segura. Se puede contar con la integridad y confiabilidad de la información sin limitar a los usuarios del sistema en su uso.

Es posible actualizar los datos desde cualquier lugar siempre que físicamente haya acceso a la información. Esto implica que el proceso, no solo de actualización de datos, sino también de su obtención, se haga de una forma más factible.

El sistema obtenido es multiplataforma lo que permite que pueda utilizar en diversos sistemas operativos. Fue desarrollado a partir de un estudio realizado sobre la gestión del trabajo metodológico en la Universidad de Holguín “Oscar Lucero Moya”, de forma tal que sea sencillo para cuantos lo usen y que esté presente la información que realmente se necesita.

Se cumplió el objetivo trazado en esta investigación con ayuda del Framework Groovy & Grails, el cual ofreció la infraestructura adecuada para la creación de la aplicación Web, pues mantiene una lógica de implementación que disminuyó considerablemente el tiempo de desarrollo.

Recomendaciones

1. Realizar el mantenimiento de este sistema para desarrollar nuevos requerimientos dentro de las funcionalidades existentes, como por ejemplo llevar el control estadístico de todas las actividades apoyado en el uso gráficos para ayudar a la empresa en la toma de decisiones.
2. Generalizar este sistema a otras facultades de la universidad para facilitar la gestión de este proceso.

Glosario de Términos

Software: Conjunto de programas que puede ejecutar el *hardware* para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

XML: *Extensible Markup Language* o lenguaje extensible de marcas que permite el intercambio de información en un formato entendible para distintas aplicaciones.

Servlets: Son objetos que corren dentro del contexto de un contenedor web Java. Su uso más común es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

JSP: *Java Server Pages* es una tecnología Java que permite a los programadores generar dinámicamente *HTML*, o algún otro tipo de página web y permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático.

Script: Es un programa, que por lo regular se almacena en un archivo de texto plano.

LAMP: es un acrónimo usado para describir un sistema de infraestructura de Internet que utiliza Linux como sistema operativo, *Apache* como servidor web, *MySQL* como gestor de base de datos y *Perl*, *PHP*, *Python* como lenguajes de programación.

GPL: *General Public License* o Licencia Pública General de GNU, es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios

finally la libertad de usar, estudiar, compartir y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre.

Wizard: Es una herramienta dentro de las aplicaciones que ayuda al usuario a realizar una tarea de manera rápida y eficaz.

JavaBean: Son un modelo de componentes creado por *Sun Microsystems* para la construcción de aplicaciones en Java.

PDF: *Portable Document Format* (formato de documento portable).

Portlets: Son componentes modulares de interfaz de usuario gestionadas y visualizadas en un portal Web. Los *portlets* producen fragmentos de código de marcado que se agregan en la página de un portal. Típicamente, siguiendo la metáfora de escritorio, la página de un portal se visualiza como una colección de ventanas de *portlet* que no se solapan, donde cada una de estas muestra un *portlet*. Por lo tanto, un *portlet* (o colección de *portlets*) se asemeja a una aplicación Web que está hospedada en un portal. Por ejemplo, un *portlet* de aplicación puede ser para el correo, el parte meteorológico, un foro, noticias, etc.

SVG: *Scalable Vector Graphics* o Gráficos Vectoriales Redimensionables son una especificación para describir gráficos vectoriales bidimensionales, tanto estáticos como animados en formato *XML*.

XLS: Es la extensión para los archivos de hoja de cálculo de Microsoft Excel.

Hibernate: Marco de trabajo, para el Mapeo Objeto-Relacional (ORM, por sus siglas en inglés) desarrollado por *JBoss*.

GORM: *Grails Object Relational* o Mapeador Objeto Relacional de Grails basado en Hibernate.

Plugin: Programa adicional que puede ser añadido a una aplicación para aumentar la funcionalidad de esta.

Scaffolding: Es una técnica soportada por varios marcos de trabajo MVC.

MVC: Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón *MVC* se ve frecuentemente en aplicaciones web, donde la vista es la página *HTML* y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

SQL: *Structured Query Language* o Lenguaje Estructurado de Consultas

Referencias bibliográficas

- [1] M. D. E. E. SUPERIOR, “210- 2007 Reglam Docente-Metod MES,” p. 52, 2007.
- [2] C. Y. Z. Purón, “SISTEMA PARA LA GESTIÓN DE LA CAPACITACIÓN EN LA EMPRESA ‘COMANDANTE ERNESTO CHE GUEVARA’. SUBSISTEMAS DE IMPLEMENTACIÓN Y EVALUACIÓN,” Universidad de Holguín “Oscar Lucero Moya”, 2012.
- [3] F. E. Fuentes, “Sistemas de información para la gestión de la empresa.”
- [4] Y. A. Parra, “Módulo para la Gestión de Trámites de la Dirección de Relaciones Internacionales de la Universidad de Holguín ‘Oscar Lucero Moya’,” Universidad de Holguín “Oscar Lucero Moya”, 2013.
- [5] D. E. Rivera, “Módulo de Configuración del Sistema de Gestión Empresarial Medioambiental (GEMA), para el Sistema Integrado de Gestión (SIG) de la Oficina Territorial de Normalización de Holguín,” Universidad de Holguín “Oscar Lucero Moya”, 2011.
- [6] T. R. Almaguer, “Sistema para la gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín ‘Oscar Lucero Moya’,” Universidad de Holguín “Oscar Lucero Moya”, 2010.
- [7] J. J. Garrett, P. Jesse, and J. Garrett, “Ajax : Un Nuevo acercamiento a las Aplicaciones Web,” 2005.
- [8] S. Ciirm, “Arquitectura y diseño de sistemas web modernos,” pp. 1–6, 2004.

- [9] Y. R. C. Yoanny Torres Rubio, Dasiel Cordero Morales, “SISTEMA EXPERTO DE MITIGACIÓN DE RIESGOS EXPERT SYSTEM FOR RISK MITIGATION.”
- [10] D. König, *Groovy in Action*. .
- [11] J. J. Gutiérrez, “¿ Qué es un framework web ?,” pp. 1–4.
- [12] Rogers S. Pressman, *Ingeniería del Software: Un enfoque práctico*. p. 614.
- [13] G. K. Rocher, *The Definitive Guide to Grails*. 2006, p. 383.
- [14] B. Abdul-jawad, *Groovy and Grails Recipes*. 2009, p. 407.
- [15] P. F. Dodino, “Diseño de aplicaciones web en Grails Índice,” pp. 1–43, 2013.
- [16] N. Brito, *Manual de desarrollo web con Grails*. 2009, p. 167.
- [17] Metodología Iconix Metodología ICONIX.htm, “Metodología Iconix Metodología ICONIX.” 2014.
- [18] R. S. and Corporation, *Rational Unified Process Best Practices for Software Development Teams*. 1998.
- [19] J. E. Miros, *ICONIX. Notas del método con ampliaciones y mejoras*. .
- [20] and M. C.-C. D. Rosenberg, M. Stephens, “Agile Development with ICONIX Process,” 2009.
- [21] J. E. G. Cornejo, “Lenguaje de Modelado Unificado. Consultado el 17 de marzo, 2011 en,” <http://www.docircs.cl/uml.htm>. .
- [22] J. Conallen, *Building Web Applications with UML*, 2nd Editio. 2009.

- [23] G. Á. Tang, "NeuroGest: Sistema para la Gestión de Información de Pacientes en el Servicio de Neurología Pediátrica en Holguín.," Universidad de Holguín "Oscar Lucero Moya", 2011.
- [24] D. Pecos, "PostGreSQL. Consultado el 21 de diciembre, 2010, en," http://danielpecos.com/docs/mysql_postgres/x15.html. .
- [25] M. Kruckenberg and J. A. Y. Pipes, *Pro MySQL*. 2005.
- [26] C. Darie and E. Balanescu, *Beginning PHP and MySQL E-Commerce*. .
- [27] B. Bulguer, *MySQL/PHP Database Applications*. 2010.
- [28] Z. P. Canelón, "SISTEMA INFORMÁTICO PARA LA LEGALIZACIÓN DE TRÁMITES EN LA DIVISIÓN DE REGISTRO, CONTROL Y EVALUACIÓN DE ESTUDIOS DE LA ZONA EDUCATIVA YARACUY," Universidad de Holguín "Oscar Lucero Moya."
- [29] C. W. and R. Breidenbach, "Spring in Action," 2006.
- [30] M. Rebeca, C., San, P. D., & Oliva, "Metodología ICONIX, 4-10.," 2009.
- [31] M. Rosenberg, D. y Stephens, *Use Case Driven Object Modeling with UML. Theory and Practice*, vol. 1, no. published. 2008.

Anexos:

Anexos I

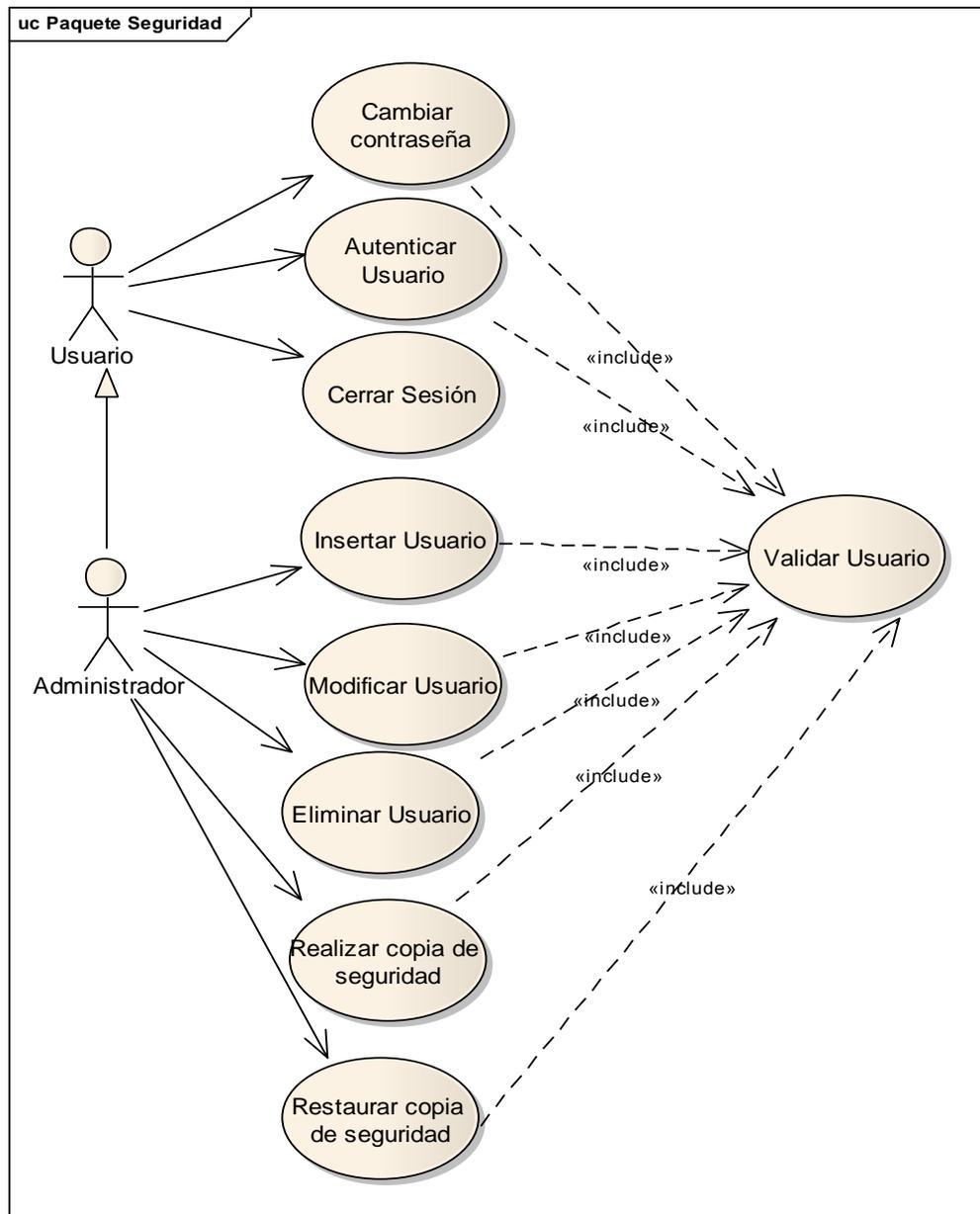


Figura 12: Diagrama de Casos de Uso: Paquete Seguridad

Fuente: Elaboración propia

Anexo II

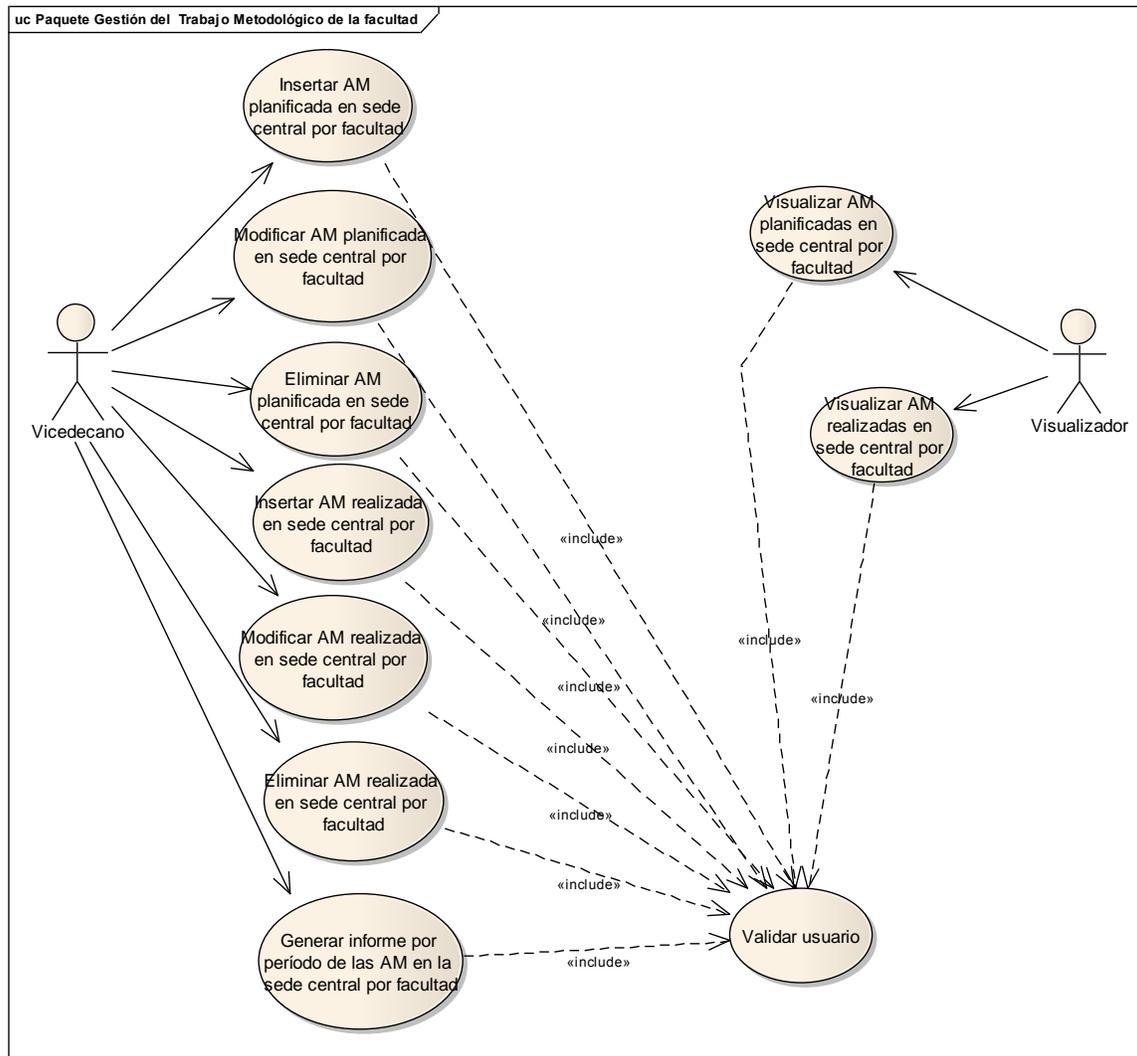


Figura 13: Paquete Gestión del trabajo metodológico de la facultad

Fuente: Elaboración propia

Anexo III

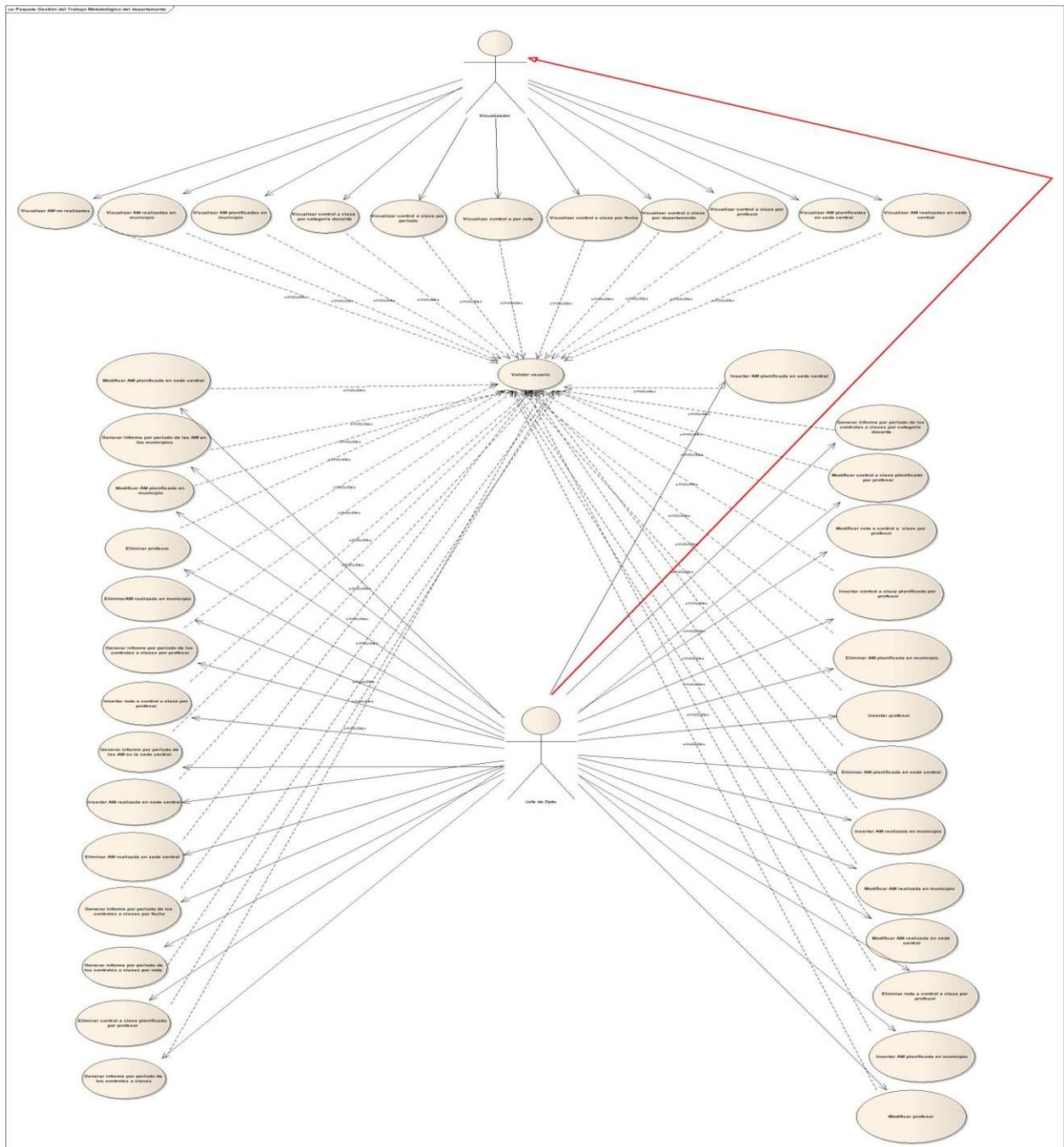


Figura 14: Paquete Gestión del trabajo metodológico del departamento.

Fuente: Elaboración propia

Anexo IV

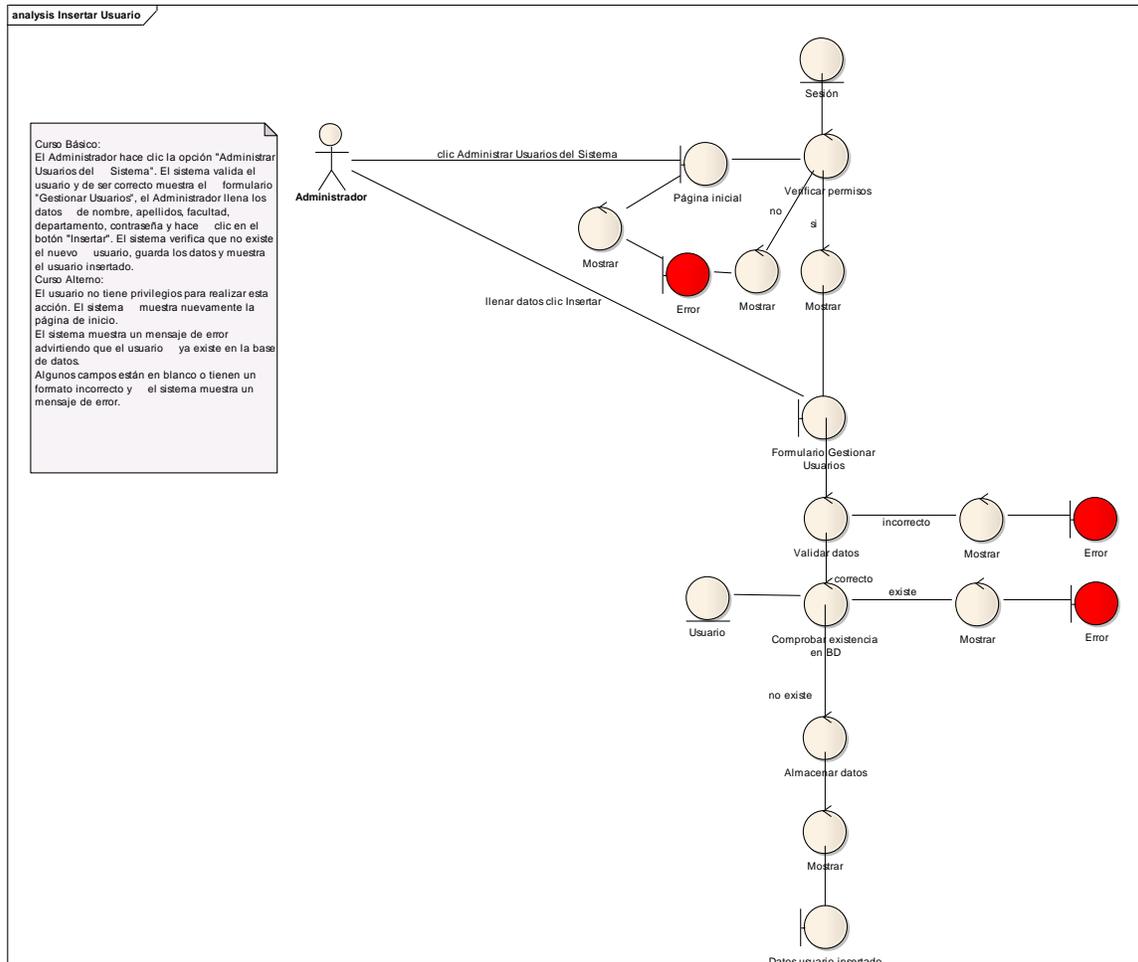


Figura 15: Diagrama de robustez. Caso de Uso: Insertar usuario

Fuente: Elaboración propia

Anexo V

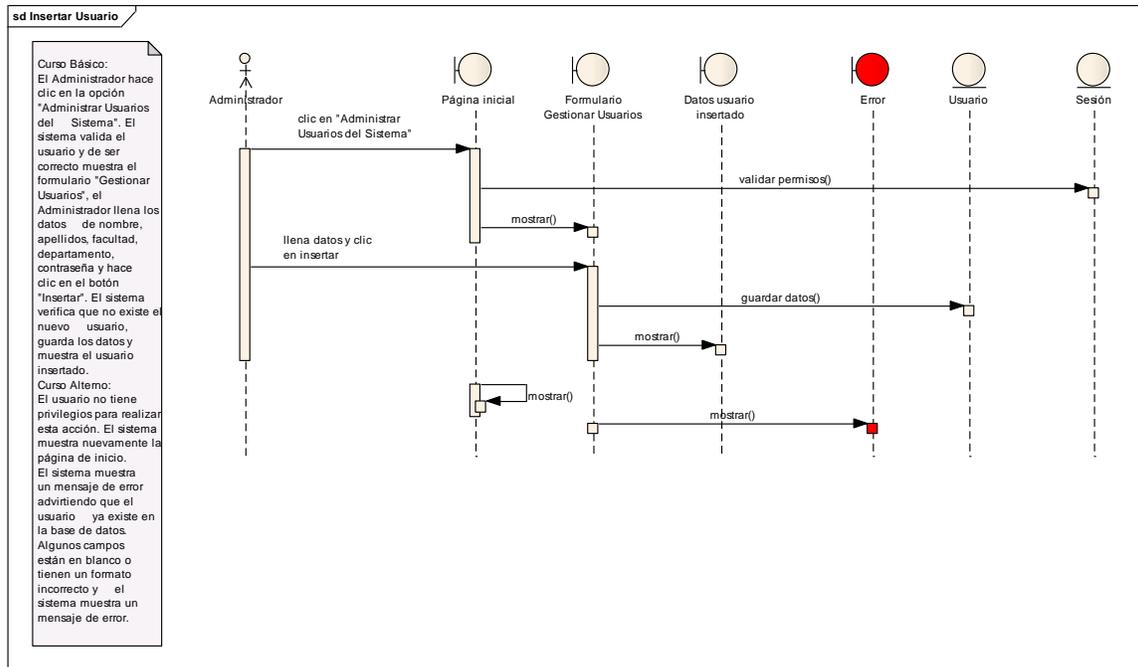


Figura 16: Diagrama de secuencia. Caso de Uso: Insertar usuario

Fuente: Elaboración propia

Anexo VI

ENCUESTA A ESPECIALISTAS

Nombre y apellidos: _____

Institución a la que pertenece: _____ Cargo actual: _____

Calificación profesional, grado científico o académico:

Profesor: _____ Licenciado: _____ Especialista: _____ Máster: _____ Doctor: _____

Años de experiencia en el cargo: _____

Años de experiencia docente y/o en la investigación: _____

Como parte de un tema de tesis en la Universidad de Holguín “Oscar Lucero Moya” se elaboró el “**Sistema para la gestión de información del trabajo metodológico**”.de la propia universidad para la gestión y control de toda la información referente al proceso en cuestión. Se requiere su opinión con relación a:

- Grado de relevancia de los aspectos “**Sistema para la gestión de información del trabajo metodológico**”.
- ¿Qué aspectos cree Ud. que deben ser incluidos o eliminados?

- Sugerencias de cambios de denominación de los aspectos propuestos, cuyo grado de relevancia, sometemos a su consideración.

Indicaciones:

A continuación le presentamos una tabla que contiene los aspectos del Sistema para la gestión del trabajo metodológico. A la derecha aparece la escala:

MA: Muy adecuado. **BA:** Bastante adecuado. **A:** Adecuado.

PA: Poco adecuado **NA:** No adecuado.

- Marque con una cruz (X) en la celda que se corresponda con el grado de relevancia que usted otorga a cada aspecto del sistema.
- Le agradecemos anticipadamente el esfuerzo que sabemos hará para responder, con la mayor fidelidad posible a su manera de pensar la presente encuesta.

Sobre el “Sistema para la gestión de información del trabajo metodológico”					
	MA	BA	A	PA	NA
Criterios					
¿Cómo aprecia la forma en que se estructura la distribución de los elementos en pantalla?					
¿Cómo evalúa la forma de insertar, modificar y eliminar los diferentes tipos de actividades metodológicas?					

¿Cómo aprecia el uso de los iconos empleados de acuerdo a las acciones que se realizan?					
¿Cómo evalúa las facilidades que brinda el sistema para la gestión del trabajo metodológico?					
¿Cómo valora la correspondencia entre el sistema y el flujo de trabajo que se realiza en los departamentos?					
¿Cómo aprecia la facilidad del uso del sistema?					
¿Cómo evalúa la generación de informes realizados mediante el sistema?					
¿Cómo evalúa la forma en que se gestiona el resto de la información referente al trabajo metodológico?					

MA: Muy adecuado **BA:** Bastante adecuado **A:** adecuado.

PA: Poco adecuado **NA:** No adecuado

➤ Escriba a continuación qué aspectos considera que deben ser incluidos o eliminados en esta propuesta:

Aspectos que se proponen ser incluidos	Aspectos que se proponen ser eliminados

- Otra sugerencia que usted desee hacer sobre la propuesta el Sistema para la gestión de información del trabajo metodológico de la Universidad de Holguín que estamos sometiendo a su consideración.

Fuente: Elaboración propia