



UNIVERSIDAD DE HOLGUÍN

“OSCAR LUCERO MOYA”

FACULTAD DE INFORMÁTICA Y MATEMÁTICA

Trabajo de Diploma en opción al Título de Ingeniero Informático

**Sistema informático para la gestión de la
información profesoral en los Departamentos
Docentes de la Universidad de Holguín Oscar
Lucero Moya.**

AUTORA

Yasmín Peña Godales

TUTOR:

Dr. Luis Cuevas Rodríguez

Holguín, Julio de 2010

Agradecimientos

A Dios y la Virgen de la Caridad por permitir que llegara a este día

A mis padres por su gran amor, apoyo y dedicación en todo momento

A mis abuelas, en especial a mi abuelita Consuelo por su amor y ternura, por ser mi guía hacia el camino correcto

A Miguelito mi amor, gracias por estar a mi lado en todo momento y por tu comprensión

A mis tíos, primos y toda la familia en general por estar siempre ahí cuando los necesité

A mi tutor por su ayuda y preocupación siempre

A Vidalina por sus consejos y darme fortaleza para seguir adelante

A Humberto, un amigo especial por su apoyo incondicional

A Iris por su compañía y cooperación

A todas mis amistades, compañeros de aula y todas las personas que de una forma u otra pusieron un granito de arena en la realización de esta tesis.

RESUMEN

El Sistema para la Gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya” surge por la necesidad de contar con una herramienta que apoye el desempeño y calidad de trabajo de los jefes de departamentos, pues hasta el momento estos directivos realizan el trabajo de forma manual, esto dificulta el registro, actualización y recuperación de la información, lo cual se traduce en ineficiencias.

Atendiendo a estas necesidades, se desarrolla una aplicación Web para facilitar la gestión de la información y procesamiento de los datos en esta área de forma rápida y segura, con nuevas potencialidades que lo conviertan en un producto sostenible. En este documento se expone la fundamentación del tema, la propuesta de herramientas para el desarrollo del producto así como la descripción, construcción y el estudio de factibilidad del sistema, con el propósito fundamental de gestionar la información profesoral de una manera eficiente.

En el proceso de desarrollo del producto se utiliza Iconix como metodología de desarrollo de software, Java como lenguaje de programación, con soporte de base de datos en PostgreSQL. Con el objetivo de agilizar el desarrollo de la aplicación se utilizan los frameworks ZK, para el diseño de interfaces e Hibernate para la capa de persistencia de datos. Como servidor Web el Apache Tomcat.

Abstract

The System for the Storage of information about teachers of the University of Holguin “Oscar Lucero Moya” was developed as a useful tool to help the directors of the department carry out their work more efficiently. Until now these workers had realized their difficult work manually, this includes registration, updating and recuperating information which can result in various inefficiencies.

As a means of solving these problems, this investigation has resulted in the development of an application web that facilitates the storage of information and the safe, rapid processing of the data with new functionalities, resulting in the creation of a sustainable product. In this document the basis of the topic is discussed, the proposal of tools for the development of the product, that is the description, construction and the study of sustainability of the system, with the fundamental purpose of realizing the efficient storage of the information about teachers.

The development of the product was realized using the methodology of Iconix, Java as the programming language, with support of the database PostgreSQL, the frameworks ZK was used for the designing of interfaces and Hibernate for the layer of persistence. Apache Tomcat was used as a web server.

Índice de Contenido

Introducción	1
Capítulo 1: Fundamentos teóricos sobre la gestión de la información profesoral en los departamentos docentes.	8
1.1 Gestión de la información	8
1.2 Departamentos Docentes.	9
1.2.1 Procesos en el departamento docente	12
1.3 Sistemas automatizados para la gestión de la información profesoral.	15
1.4 Tecnologías para el desarrollo de un sistema informático.	18
1.4.1 Metodologías de desarrollo de software.	21
1.4.2 Frameworks.....	29
1.4.3 Sistemas gestores de base de datos (SGDB).....	32
1.4.4 Servidores Web.....	35
Capítulo 2: Sistema para la gestión de la información profesoral.	38
2.1 Análisis de los requerimientos:.....	38
2.1.1 Modelo de dominio.	42
2.1.2 Prototipación rápida	43
2.1.3 Modelo de Casos de Uso.....	46
2.2 Análisis y diseño preliminar	47
2.2.1 Descripción de los casos de uso.....	48
2.2.2 Diagrama de robustez	49
2.3 Diseño	52
2.3.1 Diagrama de secuencia	53
2.3.2 Diagrama de clases del diseño	53
2.3.3 Modelo de datos	53
2.4 Implementación.....	54
2.4.1 Estándares de código.....	54
2.4.2 Tratamiento de errores	59
Capítulo 3: Validación de la solución.....	62
3.1 Impacto administrativo	62
3.2 Impacto Socio-Humanista.....	65
3.3 Impacto Ambiental.....	65
3.4 Impacto Tecnológico	66
3.5 Valoración del sistema por criterio de expertos.....	67
Conclusiones Generales	71
Recomendaciones	72
Referencias bibliográficas	73
Bibliografía	77

Índice de Figuras

Figura 1: Arquitectura Cliente - Servidor	1
Figura 2: Metodología ICONIX.....	24
Figura 3: Modelo del dominio	43
Figura 4: Interfaz principal	44
Figura 5: Insertar Profesor	45
Figura 6: Gestionar Profesores.....	46
Figura 7: Diagrama de los principales Casos de Uso del Sistema.....	47
Figura 8: Diagrama de robustez Insertar Profesor	50
Figura 9: Modelo de Despliegue.....	51
Figura 11: Diagrama de secuencia Insertar Profesor	53
Figura 13: Ventana de error	60

Índice de Tablas

Tabla 1: Descripción del Caso de Uso del Sistema Insertar Profesor.....	48
Tabla 2: Descripción del Caso de Uso del Sistema Eliminar Alumno Ayudante.....	49
Tabla 3: Historial de versiones del proyecto.....	55
Tabla 4: Norma de los componentes visuales.....	57
Tabla 5: Resumen	64

Índice de Anexos

Anexo 1: Descripciones de algunos Casos de Uso del Sistema.....	79
Anexo 2: Algunos diagramas de robustez asociados a los Casos de Uso del Sistema	81
Anexo 3: Diagrama de Casos de Uso del Sistema	84
Anexo 4: Diagramas de secuencia asociados a algunos Casos de Uso del Sistema	79
Anexo 5: Procedimiento COCOMO.....	87
Anexo 6: Encuesta para la determinación de expertos	95
Anexo 7: Encuesta aplicada a los expertos para la valoración del sistema.....	97
Anexo 8: Resultados de la encuesta de opinión de los usuarios aplicando el método Delphy.....	99

Introducción

El crecimiento de Internet y la consecuente digitalización de la información han acelerado el conocimiento científico y la evolución tecnológica. Las características más distintivas de las nuevas tecnologías hoy son la inmaterialidad, interactividad, instantaneidad, innovación, elevados parámetros de calidad de imagen, sonido, digitalización, automatización, interconexión y diversidad.

En las últimas décadas, con el desarrollo de la computación, los avances en la electrónica y una creciente inserción de las comunicaciones en el ámbito del ciberespacio, se ha hecho muy normal la utilización de ambos medios: la computación y las comunicaciones, para lograr un progresivo avance en la gestión de la información y la posibilidad de administrar desde cualquier punto del planeta; donde ya no se hace necesario que el grupo de trabajo esté en el mismo lugar físico y ni siquiera la información que éste gestione y administre; posibilitando rapidez y eficiencia en las diferentes operaciones que a diario se realizan donde el uso de las computadoras ha sido generalizado.

De ahí la existencia de sitios y aplicaciones Web como expresión de un sistema de información, que pretende facilitar la organización y el acceso a los contenidos y servicios disponibles en los sitios. Esto está conmoviendo a prácticamente todos los campos de la sociedad cubana, y la educación no es una excepción. Esas tecnologías se presentan cada vez más como una necesidad en el contexto de sociedad donde los rápidos cambios, el aumento de los conocimientos y las demandas de una educación de alto nivel constantemente actualizada se convierten en una exigencia permanente. Por su parte, las universidades del país, factor clave dentro de la educación, no están exentas de estos avances.(Rosario, 2006)

La Universidad de Holguín “Oscar Lucero Moya”, fundada en el año 1973, está ubicada en la Avenida XX Aniversario s/n, reparto. Piedra Blanca, Holguín, CP 80100. Tiene como función principal la formación de futuros profesionales de la sociedad cubana, concediéndole la importancia que la misma requiere y trabajando por lograr que dicha función se logre con la mayor calidad posible. Está organizada en 7 facultades: Informática y Matemática (FACINF), Ciencias Económicas y Contables (FACCE), Ingeniería Industrial y Turismo (FACII),

Ingeniería (FACING), Humanidades (FACHUM), Ciencias Sociales (FACCSOC) y Derecho (FACDER).

En ellas se pueden estudiar las siguientes carreras de Ingeniería como Informática, Mecánica, Industrial, Civil, Agrónoma; así como las Licenciaturas en Contabilidad y Finanzas, Economía, Lengua Inglesa, Derecho, Estudios Socio-Culturales, Turismo y Periodismo.

Para ello cuenta con 719 profesores, distribuidos en los Departamentos Docentes que constituyen la unidad básica estructural científico-docente, encargada de garantizar la alta calidad del proceso docente educativo, mediante el trabajo docente metodológico y de investigación científica, en disciplinas afines; así como la preparación de los cuadros científicos y científicos pedagógicos, la superación de los mismos y el trabajo educativo con los estudiantes.

Los procesos de gestión vinculados a las actividades que se desarrollan en estas unidades docentes generan una serie de tareas que son responsabilidad del jefe de departamento quien planifica, organiza, ejecuta y evalúa de manera integrada las mismas, que están vinculadas a la docencia, la investigación, el posgrado y la extensión, propias del área de conocimiento.

Como tarea principal, un jefe de departamento confecciona el plan de trabajo de todo el departamento y controla el de los subordinados. Además lleva el control sobre los datos de su personal (nombre, apellidos, sexo, edad, e-mail, categoría científica, categoría docente, entre otros), así como toda la información referente a su trayectoria en la universidad (disciplina en la que trabaja, asignatura que imparte, publicaciones, participación en eventos, proyectos a que ha estado vinculado, cuándo transitó de categoría docente y otros).

Realiza el cálculo de volumen de docencia de sus profesores y elabora la información necesaria, de acuerdo con las metodologías para el cálculo de las necesidades del personal docente y el fondo de tiempo que la Dirección del Ministerio de Educación Superior (MES) elabora al efecto, efectúa los controles a clases y evalúa a sus profesores, controla la marcha y resultados de los trabajos de investigación, planifica y supervisa la preparación e impartición de cursos de postgrado por parte de los profesores de más experiencia del

departamento. Éstas, entre otras actividades que revisten importancia para el departamento. (MES, 1977)

A partir del estudio que se realizó en la Universidad de Holguín “Oscar Lucero Moya” se determinó que no están formalmente establecidas normas ni procedimientos para el desempeño del trabajo de un jefe de departamento docente. Consecuentemente, en la mayoría de los casos el trabajo se desarrolla por la experiencia que se va adquiriendo en el transcurso del desarrollo del mismo, o por rutinas empíricas que han surgido. Para el registro y procesamiento de los datos que estos jefes realizan, existen procedimientos manuales como tomar datos por escrito, llenar una planilla o mediante una autoevaluación, lo cual genera baja calidad, pues al no tener una adecuada estrategia de organización de los datos, éstos se repiten. De esta forma se hacía difícil llevar un control centralizado de la información.

En el diagnóstico se estableció que tampoco existe una herramienta informática capaz de gestionar toda la información de los profesores y el propio departamento. Actualmente este trabajo se realiza de forma manual en algunos departamentos y en otros apoyándose de las herramientas Microsoft Office Word y Excel, por lo que la integridad de la información es mínima y el desempeño del trabajo se convierte en procesos largos y tediosos.

Además, el modo en que se guarda la información no ofrece garantía de protección, no asegura posibilidades de acceso futuro y dificulta en gran medida la corrección de los datos almacenados, por lo que no se garantiza la perdurabilidad de esta información en el tiempo.

La situación es especialmente crítica en los Departamentos Docentes Carrera, quienes tienen el protagonismo en el trabajo docente-metodológico y educativo de la carrera que les corresponde atender. En ello incide el número de profesores, cantidad de asignaturas a impartir, sus características, el número de alumnos por grupo y tipo de cursos, el grado de afinidad entre las disciplinas, el número de docentes adjuntos, el número de integrantes del movimiento de Alumnos Ayudantes, el peso específico del trabajo investigativo, los laboratorios y talleres a su cargo; lo que significa que hay tareas específicas que también deben planificarse y controlarse relacionada con estos aspectos.

Es importante destacar que los jefes de departamento quienes además de desarrollar las actividades como profesores, sistemáticamente reciben orientaciones y solicitudes de

informes lo que les obliga a procesar gran volumen de información, por lo que necesitan hacer búsquedas por diferentes criterios sobre sus profesores, consultar sus datos, estudiar su trayectoria, para realizar una adecuada toma de decisiones, ya sea realizar una evaluación, decidir quienes de ellos deben participar en una determinada actividad, entre otras. La recuperación y actualización de la información se dificultan, por lo que se presentan demoras en la entrega de informes, lo que se traduce en ineficiencias. Esto le resulta difícil y requiere de mucho tiempo, pues la información no está organizada, tampoco se encuentra almacenada de la mejor forma, no se tiene un control de los datos de cada profesor por individual sino que necesita consultar varios documentos con el riesgo de que no estén actualizados y se vuelve un proceso bastante lento.

Tras un estudio de esta situación se hizo evidente el siguiente **problema**: ¿Cómo favorecer el proceso de gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya”?, el cual se encuentra enmarcado en el **objeto de estudio**: proceso de gestión de la información profesoral en los centros de educación superior.

Para dar solución a este problema se propuso el siguiente **objetivo**: Elaboración de un sistema informático para favorecer el proceso de gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya”.

El objetivo antes planteado delimitó el **campo de acción**: Informatización del proceso de gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya”.

Para guiar la investigación se planteó la **hipótesis**: La elaboración de un sistema informático que gestione toda la información profesoral generada en un departamento docente de una forma eficiente, en un entorno amigable para el usuario, que sea rápido en las respuestas, realice búsquedas por diferentes criterios con alto grado de confiabilidad y brinde integridad en los datos, favorecerá el desempeño y la calidad del trabajo de los jefes de departamento.

Para el desarrollo de la investigación se ejecutaron las siguientes **tareas**:

1. Elaborar los fundamentos teóricos que sustentan el proceso de gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya”.
2. Diagnosticar el estado actual del proceso de gestión de la información profesoral en los departamentos docentes.
3. Estudiar las tendencias y tecnologías existentes para el desarrollo de sistemas informáticos.
4. Capturar los requerimientos de la solución propuesta.
5. Valorar la sostenibilidad del sistema informático.
6. Diseñar e implementar la solución propuesta.
7. Valorar el sistema informático con respecto al grado de satisfacción de posibles usuarios.

Entre los **métodos** que se usaron para llevar a cabo las tareas están:

Teóricos

- ❖ **Análisis - Síntesis:** se utilizó para determinar los fundamentos teóricos de la investigación, conceptos relacionados con el objeto, la comprensión de las relaciones esenciales y características generales del proceso, así como la elaboración de las conclusiones de la investigación.
- ❖ **Modelación:** se empleó durante la fase de diseño del sistema; guiado por la metodología de ingeniería de software ICONIX, usando UML como lenguaje de modelado; de manera que el conjunto de modelos de la metodología describiera desde todas las perspectivas posibles el proceso de desarrollo en sentido general.
- ❖ **Histórico - Lógico:** se llevó a cabo un análisis del proceso de gestión de información en los departamentos docentes de la Universidad de Holguín, para conocer con mayor profundidad los antecedentes y las tendencias actuales.
- ❖ **Hipotético-Deductivo:** se empleó para la elaboración de la hipótesis que la investigación se encargó de analizar su certeza.

Empíricos

- ❖ **Entrevista:** permitió la interacción directa con el personal interesado en el sistema, recoger las informaciones importantes, las necesidades del jefe de departamento para ayudar en su labor y determinar los principales requerimientos del sistema.
- ❖ **Observación:** permitió realizar un análisis detallado de las condiciones de los medios informáticos con que cuenta el departamento, para valorar el nivel de eficiencia y calidad que deberá tener el sistema acorde a las condiciones reales.
- ❖ **Encuesta:** Se realizó encuesta a expertos en materia de informática para la validación de la solución.

El presente trabajo desarrollará un sistema para el registro, organización, actualización y conservación de las informaciones relacionadas con los profesores. Permitirá tener un control preciso e individual de los resultados de cada profesor en las tareas asignadas en el plan de trabajo previsto. Contribuirá a profundizar en los procesos de evaluación y caracterización de cada uno de los profesores a partir de elementos de juicios precisos y concretos.

El documento está estructurado en introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y anexos.

El Capítulo 1, *“Fundamentos teóricos sobre la gestión de la información profesoral en los departamentos docentes”*, ofrece una breve descripción de diferentes conceptos imprescindibles que dan base a la investigación y al objeto de estudio, el flujo de los procesos en los que interviene el mismo y un análisis crítico al respecto. Se brinda además una panorámica de los sistemas automatizados existentes vinculados al campo de acción, las tendencias y tecnologías actuales a emplear para la selección de las herramientas para el modelado y construcción de la solución propuesta.

El Capítulo 2, *“Sistema para la gestión de la información profesoral ”*, explica el uso de la metodología expuesta en el capítulo inicial para el diseño de la aplicación y el desarrollo del sistema presentado en diagramas y tablas para su mejor comprensión.

El Capítulo 3, *“Valoración de Sostenibilidad del Sistema Propuesto”*, se estudian los beneficios que obtendrá el cliente con la implantación del sistema, concluyendo con una

valoración de la sostenibilidad del sistema diseñado. Además se ofrecen los resultados obtenidos luego del procesamiento de la encuesta a expertos la cual valida de alguna forma la hipótesis planteada.

Capítulo 1: Fundamentos teóricos sobre la gestión de la información profesoral en los departamentos docentes.

En el presente capítulo, con el objetivo de entender con más claridad los procesos de negocios, se realiza una revisión de los conceptos a los cuales se harán referencia en todo el contexto del trabajo. Se brinda información sobre los Departamentos Docentes y su funcionamiento, de los sistemas automatizados existentes vinculados al campo de acción, los objetivos que se persiguen y las tendencias y tecnologías a considerar para cumplir el objetivo propuesto en la presente investigación.

Las acciones de informatización o automatización de procesos, cualesquiera que estos sean, requiere previamente de un análisis exhaustivo de la gestión que se realiza de la información. Es necesario tener bien definido qué se entiende por gestión de información y cuáles son sus principales elementos.

1.1 Gestión de la información

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

La información para la gestión es la información necesaria para tomar decisiones de gestión, la cual es importante para:

- ✓ Tomar las decisiones necesarias para mejorar la gestión de prestaciones y servicios.
- ✓ Poner en práctica la planificación, implementación, supervisión y evaluación participativas.

Para poder utilizar la información para tomar decisiones de gestión, debe gestionarse la información (recabar, registrar y analizar). Aunque la gestión de la información (el proceso de recabar y guardar la información) y la información para la gestión (la información necesaria para tomar decisiones bien documentadas) son diferentes, se refuerzan entre sí y no pueden separarse en las operaciones cotidianas.

Por lo tanto, la gestión de la información implica (Vasconcelos, 2005):

- ✓ Determinar la información que se precisa.
- ✓ Recoger y analizar la información.
- ✓ Registrarla y recuperarla cuando sea necesaria.
- ✓ Utilizarla.
- ✓ Divulgarla.

En la presente investigación la información que se quiere gestionar está estrechamente vinculada con los Departamentos Docentes y todo lo relacionado con ello. Los procesos que serán controlados son de los profesores los cuales, a su vez, pertenecen a un Departamento.

1.2 Departamentos Docentes.

Un departamento docente es el nivel de dirección que rige el trabajo de los profesores y se ocupa directamente de proyectar un trabajo metodológico que priorice la formación y desarrollo de valores en los estudiantes, desde el contenido de las asignaturas hasta las disciplinas que son de su competencia. (MES, 2006)

Como resultado del estudio realizado se identificó como objetivo fundamental del departamento docente, la formación de especialistas de alto nivel profesional, mediante la planificación, organización, administración y control del trabajo docente metodológico y de investigación, la superación del personal docente y el trabajo educativo en general, a través de las disciplinas que se imparten en el mismo.

Se detectaron como funciones de los Departamentos Docentes las siguientes: (Rodríguez y Salgado, 2009)

- ✓ Analizar los métodos, técnicas, contenidos y recursos aplicados en los procesos de enseñanza y aprendizaje en el departamento.
- ✓ Planificar y poner en marcha propuestas metodológicas que tiendan a la integración vertical y horizontal de la enseñanza en el área de estudio de su competencia.
- ✓ Evaluar los resultados obtenidos.
- ✓ Realizar toda otra tarea de asesoramiento o cooperación que le fuera requerida por Dirección.

- ✓ Coordinar la labor docente con los otros departamentos y áreas.

Documentos e informaciones a controlar en un Departamento Docente:

- ✓ Estrategia del Dpto. según las prioridades.
- ✓ Proyectos Educativos de las brigadas y diagnóstico previo.
- ✓ Plan de trabajo Metodológico y las acciones realizadas, controles a clases efectuados.
- ✓ Actas de las reuniones de los colectivos de año.
- ✓ Diagnóstico y planes de intervención de los tutores.
- ✓ Expedientes de todas las asignaturas.
- ✓ Expedientes de todas las disciplinas.
- ✓ Gráfico y Horarios y el cumplimiento del plan de estudio.
- ✓ Exámenes aplicados y la instrumentación de la norma de ortografía

Las actividades del departamento docente están dirigidas en tres áreas fundamentales docencia, investigación y postgrado. El jefe del mismo cumple un papel importante en el desarrollo de ellas, pues es el encargado de controlar y dirigir todos los procesos que tienen lugar en estas estructuras y que son de relevante importancia para el cumplimiento de los objetivos trazados en cada una de las ARC que conforman la línea de trabajo.

Para poder cumplir la Misión y Visión de una Institución se definen sus objetivos estratégicos, a partir de un diagnóstico inicial. Esos objetivos se organizan a nivel de empresa, como parte de la planeación estratégica, en Áreas de Resultados Claves (ARC), a partir de las cuales se definen los elementos que deben “halar” a la organización hacia un plano superior.

En las universidades cubanas, caso particular de empresas con características especiales, se introdujeron, algunos cursos atrás, las ARC. Esta tecnología establece criterios de medidas, definidos por la Dirección de Recursos Humanos del MES, para realizar la evaluación integral del trabajo desarrollado por el personal docente. La organización por ARC, adaptación de la establecida para las empresas en el país, ha sufrido modificaciones desde su instauración. Hasta el pasado curso eran nueve pero actualmente existen solamente dos y dentro de ellas se establecen los objetivos de trabajo del presente curso (UHO, 2009).

❖ ARC 1: Perfeccionamiento de la Labor Educativa y Político-Ideológica.

Objetivo1: Alcanzar niveles superiores de compromiso con la Revolución y de participación consecuente y cumplimiento de las tareas propias de la comunidad universitaria como resultado del perfeccionamiento de la labor educativa y el trabajo político-ideológico.

❖ ARC 2: Perfeccionamiento de la Educación Superior.

Objetivo 2: Perfeccionar la Gestión Universitaria para lograr niveles superiores de calidad.

Objetivo 3: Elevar la calidad de la Educación Superior en los municipios.

Las ARC son de vital importancia, pues constituyen el criterio de medida que tiene el jefe de departamento para realizar la evaluación anual de sus profesores.

Los profesores juegan un papel fundamental en todos los procesos que se realizan en el departamento, pues tienen a su cargo la dirección del proceso de enseñanza-aprendizaje, en tanto debe planificar, organizar, regular, controlar y corregir el aprendizaje del alumno y sus propias actividades. Deben estar en constante interacción y comunicación con sus alumnos, con sus colegas y con el resto los trabajadores del departamento. (2006)

De ellos se lleva el control sobre todos sus datos entre ellos: nombre, apellidos, sexo, edad, e-mail, categoría científica, categoría docente, entre otros, así como toda la información referente a su trayectoria en la universidad: disciplina en la que trabaja, asignatura que imparte, participación en publicaciones, forum, revistas, registros, proyectos a que ha estado vinculado, alumnos ayudantes que atiende, cuándo transitó de categoría docente y otros.

Además de las tareas que realizan, se detectaron como funciones de los profesores las siguientes: (MES, 2006)

- ✓ Educar para la formación de valores éticos y morales, convicciones personales, hábitos de conducta y personalidades integralmente desarrolladas que piensen y actúen creadoramente, aptas para construir la nueva sociedad y defender las conquistas de la Revolución.
- ✓ Realizar la planificación, ejecución y control del proceso docente de pregrado y postgrado en todas sus formas, de acuerdo con su categoría docente.

- ✓ Desarrollar actividades metodológicas y de superación inherentes al proceso docente de pregrado y postgrado, hasta el nivel de actualización que requiera el desarrollo exitoso de las funciones correspondientes a su categoría.
- ✓ Elevar constantemente sus conocimientos pedagógicos, científico-técnicos y culturales.
- ✓ Realizar investigaciones, trabajos de desarrollo y de innovación tecnológica, así como servicios científico técnicos y de aplicación que contribuyan al desarrollo de las fuerzas productivas de la sociedad y al perfeccionamiento de la vida social en su conjunto y cumplir las regulaciones establecidas para el personal docente universitario.
- ✓ Dirigir procesos académicos y/o unidades organizativas universitarias.
- ✓ Desarrollar docencia de pregrado en asignaturas de la disciplina en que ejerce sus funciones, o en disciplinas afines, así como docencia de postgrado en el mayor nivel de complejidad e integridad en las mismas disciplinas.
- ✓ Dirigir y desarrollar trabajo metodológico en la formación del profesional y la educación de postgrado.
- ✓ Dirigir y participar en la formación científico educativa del personal con categorías docentes precedentes.

1.2.1 Procesos en el departamento docente

Teniendo en cuenta los objetivos del departamento docente y las funciones que cumple el mismo, su principal proceso está encaminado al control y manejo de la información relacionada con el proceso de enseñanza-aprendizaje, donde tiene un papel muy importante el colectivo de profesores, pues son ellos los encargados de ejecutar las tareas y actividades.

El jefe de departamento, para poder lograr los objetivos propuestos, primeramente verifica el personal con que cuenta para desarrollar su trabajo. De ellos, lleva sus datos personales, controla el cumplimiento del plan de trabajo con la evaluación, la auto evaluación, el parte de cumplimiento de la investigaciones, las tutorías, los controles a clases y las afectaciones, los cursos de postgrados impartidos o recibidos; entre otras cuestiones que se relacionan con el personal y que son de importancia para él.

Por otra parte, en estas áreas se manejan procesos relacionados con la educación de postgrado de forma general para todo el departamento, por lo que también se recogen los cursos que se van impartir, llevando un registro del profesor que va a impartir cada uno, la fecha en que se desarrollará y los objetivos.

Lo propio sucede con el trabajo de Ciencia e Innovación, donde la información que se recoge está relacionada con las investigaciones realizadas y sus salidas, se lleva un control del plan de los premios, participación en eventos, registros de propiedad intelectual y publicaciones, así como los reconocimientos efectuados al departamento como colectivo y a sus integrantes individualmente.

Además, el jefe de departamento debe mantener actualizado el control de todos los documentos que los profesores deben entregar relacionados con su labor educativa, investigativa y el trabajo con los estudiantes así como los que él mismo debe emitir concernientes al departamento para la facultad. Para ello debe pedir a cada profesor la documentación antes mencionada.

Los jefes de departamentos docentes de la Universidad de Holguín “Oscar Lucero Moya” (UHOLM) no cuentan con una aplicación que organice y gestione toda la información con la que ellos trabajan, diariamente estos jefes junto con el personal asistente, reciben orientaciones, por lo que tienen que procesar un gran volumen de información para realizar una adecuada toma de decisiones, independientemente de las demás responsabilidades que tienen y las funciones que cumplen.

En la investigación realizada en UHOLM se detectó que parte del procedimiento de la gestión de la información relacionada con los profesores que se mueve en un departamento docente se realiza de forma manual, lo que provoca gran consumo de tiempo y riesgos de cometer errores. También puede verse afectado por el hecho de que la información sea guardada en lugares diferentes en dependencia de quien la use, además de que cualquier persona no autorizada tenga acceso a la misma y que en muchas ocasiones el resultado obtenido no posea la calidad requerida, afectando por consiguiente el valor del conocimiento derivado de esta información; lo que repercute negativamente en la toma de decisiones y en la eficacia del proceso de gestión. Además, se manejan los procesos sustantivos de cualquier departamento docente (formación profesional, investigación y extensión).

Para su registro y procesamiento existen procedimientos manuales, generándose duplicidad, baja calidad y entorpeciendo el desarrollo de una adecuada estrategia para su gestión y desarrollo. Debido al creciente cúmulo de información, el entorno dinámico y los constantes cambios a que están sometidas los mismos, no es posible anticipar todas las necesidades del mañana.

Como alternativa de solución al problema se propone el diseño e implementación de una aplicación Web, la misma garantizará de forma automatizada el registro, gestión y protección de la información necesaria para desarrollar el trabajo del jefe del departamento docente. Este sistema trata el trabajo específico con el capital humano, la actividad metodológica y postgrado.

Se parte de definir los datos personales del profesor y permite la interacción directa entre él y su jefe de departamento, automatizando funciones de vital importancia, como la confección del currículum según ciertos estándares que se establezcan en el departamento y plan de trabajo del profesor, su evaluación, los resultados de los controles a sus actividades docentes, su actividad científica y de postgrado, asignaturas que imparten; los premios obtenidos en revistas, forum o publicaciones, registros realizados, participación en proyectos y toda información referente al profesor.

Además, garantizará la perdurabilidad de esta información en el tiempo. Por otra parte, se integran otros aspectos necesarios para la gestión de la información en un Departamento Docente, para lograr un sistema de gestión departamental más completo. Tales aspectos son: trabajo de atención de los profesores a Alumnos Ayudantes y Grupos Científicos Estudiantiles, tutorías de tesis ya sea de formación académica (FA) o de pregrado. Por otra parte, asiste en la elaboración de reportes e informes, que servirán para tomar decisiones, puesto que se almacena un gran volumen de información relacionada con esta rama, que además estará disponible para los usuarios que necesiten consultarla, con diferentes niveles de acceso.

Para el desarrollo de sistemas enfocados a los Departamentos Docentes se requiere hacer un estudio muy intenso sobre todo en la modelación del negocio o del entorno específico. En el mundo existen muchísimos sistemas y esta gran variedad está dada por las diferencias en las políticas que cada organización adopta para sí y que a la hora de diseñar e implementar un sistema informático para la gestión de los Departamentos Docentes hay que tener en

cuenta. Luego la búsqueda de un precedente adaptable debe basarse en producciones nacionales. Es importante, por lo tanto, hacer una revisión a lo que está hecho hasta hoy, para valorar qué impacto ha logrado generar, para poder analizar sus características y el alcance empresarial que han logrado.

1.3 Sistemas automatizados para la gestión de la información profesoral.

A continuación se brinda información acerca de diferentes sistemas automatizados existentes vinculados al campo de acción, los cuales fueron valorados y ayudaron en gran medida a proponer ideas al diseño del sistema requerido, a analizar y plantear una serie de necesidades existentes, a señalar algunas perspectivas de desarrollo posibles, a proponer soluciones, herramientas, ventajas y concretar lo que se desea como producto final.

Sistema de apoyo al trabajo administrativo y toma de decisiones en el departamento de Ingeniería Mecánica (DIM 2009)

Sistema creado en la Universidad de Holguín “Oscar Lucero Moya” por parte del jefe de departamento de Ingeniería Mecánica, PT Dr. Carlos Batista Rodríguez, que permite la planificación, organización y control de los principales objetivos que debe lograr un departamento docente en un curso escolar. Se basa en un conjunto de normativas y procedimientos que debe ejecutar el directivo en tres momentos en el curso, para lograr los objetivos tácticos y estratégicos trazados: mensual, trimestral y anualmente. Parte de la experiencia personal del autor y las consultas a otros directivos y profesores de la Universidad de Holguín en las temáticas que se abordan, el sistema fue desarrollado en Microsoft Office Access. (Rodríguez y Salgado, 2009)

Sistema Docente Integral (SDI)

Sistema creado en el Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE) por el Grupo de Gestión Universitaria Digital, dirigido a aumentar la eficiencia de la gestión universitaria, específicamente el control de los procesos docentes, su planificación y resultados. De forma general, el objetivo del sistema es registrar, procesar y crear mecanismos de recuperación de la información relacionada con el proceso docente, tanto de pregrado, como de postgrado, de forma automatizada.

El SDI está conformado actualmente por 8 módulos y se trabaja para aumentar sus funcionalidades. Los módulos que están diseñados son:

1. Secretaría de Facultades
2. Matrícula
3. Estadística
4. Archivo Histórico
5. Recuperación de Información
6. Pago de Estipendios
7. Planificación Docente
8. Postgrado.

Cada módulo del SDI se desarrolló, con vista a su trabajo en red, sobre una Base de Datos centralizada que almacena la información generada desde diferentes áreas como: secretarías docentes, Economía, Archivo Central, Vicerrectoría de Investigaciones y Postgrado, entre otras.

Su relevancia se encuentra en que la herramienta es capaz de hacer un completamiento de la información de forma automática, o sea, que sin necesidad de teclear nuevamente los Programas de Disciplinas, el sistema utiliza los históricos existentes y actualiza la Base de Datos. (Estevez, 2003)

Sistema Automatizado para la Gestión de la información de Ciencia e Innovación Tecnológica de una Universidad (SIGECIT)

Sistema desarrollado en la Universidad de Matanzas “Camilo Cienfuegos”, que facilita el proceso de planificación, ejecución y control de la actividad de Ciencia y Técnica en la universidad de forma de ágil y segura a través de la Web, permitiendo la familiarización con un modelo para gestionar la dirección de la Ciencia y de la información asociada, lo que resulta de gran utilidad para el personal encargado de gestionar datos de este tipo disminuyendo considerablemente el tiempo empleado en la búsqueda de información, resultados estadísticos y haciendo más dinámica la realización de los balances y la toma de decisiones. (Endrinal y Martínez, 2008)

Sistema gestor de información en Departamentos docentes del ISMM

La Aplicación Web implementada se desarrolló en un equipo de trabajo de 4 integrantes en el Instituto Superior Minero Metalúrgico de Moa. Toca aspectos tales como:

Recursos Humanos: Gestiona la información referente a los planes de trabajo mensuales, planes de resultados individuales, PC-10, sanciones, estimulaciones, política de cuadros y otras informaciones.

Formación del Profesional: Se encarga de gestionar toda la información docente estudiantil, notas y evaluaciones alcanzadas por cada estudiante, documentación de las asignaturas, horarios de clases, proyectos, planes de estudio, actas de reuniones, etc.

Ciencia-Innovación Tecnológica: Gestiona los distintos premios y reconocimientos alcanzados por el Departamento, las estrategias, así como las relaciones y colaboraciones con otras entidades. (Carcasses, 2008)

Análisis comparativo de otras soluciones existentes con la propuesta:

Luego de un estudio exhaustivo acerca del estado del arte sobre sistemas automatizados existentes relacionados con la gestión de la información en un departamento docente, se concluye que ningún sistema encontrado cumple con los requisitos previstos y funcionalidades del sistema diseñado, pues todos los Sistemas que se encontraron, usan como Gestor de Base de datos software propietario, como es el caso del SQL Server2000 lo que significa que su uso es bajo licencia, esto representa el derecho legal de instalación y uso del software (no la propiedad).

Además algunos están implementados con herramientas privativas que los hacen dependientes de la versión de realización, como es el caso de Microsoft Access, además no posee buena navegabilidad ni una interfaz agradable para el usuario, careciendo así de seguridad, portabilidad y rapidez en las respuestas. Otros por ejemplo poseen un gran número de funcionalidades, que nunca se utilizarían, esto trae como consecuencia el aumento de la complejidad de su uso. Algunos de ellos no presentan un buen diseño de interfaces, lo que le resulta complejo e incómodo al usuario.

Si bien en la actualidad existen diversas aplicaciones que trabajan sobre la base del tratamiento de la información docente-educativa en las distintas universidades del país, la mayoría, hasta el momento, no abarcan totalmente la información acerca del control de los resultados y trabajo realizado por los profesores. En el caso de aquellos que sí lo hacen, sólo una muy pequeña parte de estos sistemas abarca aspectos relacionados con algunas de las funciones previstas en la aplicación, pero no llegan a su completa integridad.

Teniendo en cuenta lo anterior se puede apreciar la utilidad de desarrollar herramientas que brinden en mayor medida una adecuada respuesta a los requerimientos y necesidades de los Departamentos Docentes de la Universidad de Holguín, brindando mayores posibilidades de adaptaciones futuras y acceso a las informaciones, todo sobre bases del uso de un lenguaje de programación sobre plataforma Web, naciendo de esta forma la propuesta final de desarrollo del presente trabajo.

La propuesta abarca todo lo referente a los profesores tanto su información personal como profesional, siendo una propuesta robusta, íntegra y segura que utiliza tecnologías y herramientas recientes capaces de satisfacer las necesidades del cliente, facilitando así el desempeño del trabajo de los jefes de departamentos docentes.

Para el desarrollo de la aplicación web que se propone es preciso realizar un análisis profundo de las tecnologías existentes para decidir cuales de estas son las que se ajustan a los requerimientos del sistema, favoreciendo así la calidad y el aumento de la velocidad de desarrollo del mismo.

1.4 Tecnologías para el desarrollo de un sistema informático.

Para el desarrollo del Sistema que lleva el nombre de la presente tesis, se hizo un análisis de las tendencias y tecnologías actuales que se utilizan en el mundo para desarrollar aplicaciones Web. A continuación se brinda un resumen sobre dos de estas tendencias: aplicaciones Web y arquitectura cliente-servidor.

Aplicaciones Web.

Los servicios Web son hoy en día muy importantes por el uso inagotable de los mismos. Esto viene dado porque no solo permiten mostrar información estática o escrita con

anterioridad a través de Internet o una Intranet, sino que son capaces de generar datos solicitados a través de consultas, debido a la dinámica sobre la que están basados.

Una Aplicación Web es un conjunto de páginas Web enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de información. (Monmany, 2008)

Estas aplicaciones ofrecen numerosas ventajas pues estas ahorran tiempo y pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa. No hay problemas de compatibilidad pues basta tener un navegador mínimamente actualizado para poder utilizarlas. Las actualizaciones son inmediatas ya que como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.

El consumo de recursos es bajo dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos nuestros porque se realizan desde otro ordenador. Es multiplataforma pues se pueden usar desde cualquier sistema operativo.

Además son portables, no ocupan espacio en el disco duro, los virus no dañan los datos y los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones web ricas. La gran ventaja de utilizarla es precisamente que las aplicaciones Web se desarrollan en un ambiente Cliente/Servidor, esto trae como beneficio que la mayor parte del procesamiento se realiza en el servidor, disponiendo de una buena interfaz gráfica, y una mejor interacción con el usuario, garantizando que la información se mantenga íntegra.

Como en los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente, que servía como interfaz de usuario que tenía que ser instalado por separado en cada ordenador personal de cada usuario, donde el cliente realizaba peticiones a otro programa -el servidor- que le daba respuesta; no podemos dejar de mencionar la arquitectura cliente-servidor dentro del contexto de las aplicaciones Web.

Arquitectura cliente-servidor.

La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema.(Zalgado, 2009)

Esta arquitectura consiste básicamente en que un programa, el Cliente informático, realiza peticiones a otro programa, el servidor, que les da respuesta. Según IBM 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 y/o, a través de la organización, en múltiples plataformas.

En esta arquitectura se pueden distinguir 3 capas o niveles:

1. Manejador de Bases de Datos (Nivel de almacenamiento),
2. Procesador de aplicaciones o reglas del negocio (Nivel lógico)
3. Interfaces del usuario (Nivel de presentación)

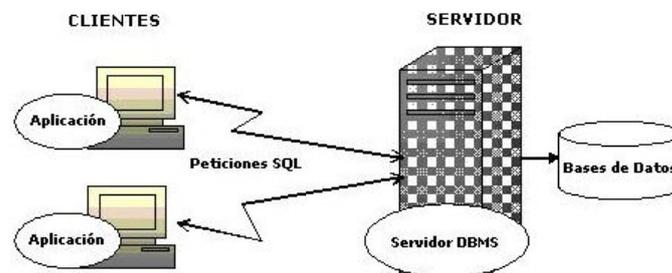


Figura 1: Arquitectura Cliente - Servidor

Una de las principales razones que conllevó a la aplicación de este modelo, es precisamente que el sistema exige integración, logrando de esta forma una buena gestión de la información profesoral por lo que es necesario tener en cuenta todas las ventajas y características que posee la arquitectura Cliente/Servidor.

Esta arquitectura presenta múltiples ventajas entre ellas los recursos centralizados pues el servidor puede administrar los recursos que son comunes a todos los usuarios. Posee seguridad mejorada gracias a que la cantidad de puntos de entrada que permite el acceso a los datos no es muy alta.

Además es escalable, fácil de mantener, presenta centralización del control pues los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. La existencia de plataformas de hardware cada vez mas baratas, permite que los costes se hayan reducido considerablemente. Además, se pueden utilizar componentes (tanto de hardware como de software) de diferentes fabricantes, lo cual favorece la implantación de soluciones. El servidor presenta a todos sus clientes una interfaz única y bien definida. El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa. Muestran información de forma sencilla a través de un hipervínculo.

El esquema Cliente/Servidor también contribuye a una disminución de los costos de entrenamiento del personal, pues favorecen la construcción de interfaces gráficas interactivas, las cuales pueden hacer que los "datos" se conviertan en "información" y además son más intuitivas y fáciles de usar.

Actualmente la gran mayoría de las aplicaciones de gestión empresarial hacen uso del modelo Cliente/Servidor, precisamente en el desarrollo de la aplicación se tendrá en cuenta por las siguientes razones:

- ✓ Petición de sistemas fáciles de usar.
- ✓ Productividad y calidad.
- ✓ Costo y rendimiento.
- ✓ Acceso a la información.
- ✓ Nuevas tecnologías y herramientas de alta productividad.

Los elementos presentados hasta el momento representan una arquitectura para el desarrollo de un sistema informático, objetivo de la investigación. Para el desarrollo de un sistema se define toda una disciplina "Ingeniería de Software", en esta se contemplan un grupo de metodologías que pueden ser utilizadas. A continuación se muestran las características de algunas de estas metodologías.

1.4.1 Metodologías de desarrollo de software.

Las metodologías de desarrollo de software surgieron a raíz de la necesidad de controlar y documentar proyectos cada vez más complejos. La implantación de una metodología es necesaria si se quieren gestionar adecuadamente los proyectos. El término de metodología

se define como un conjunto de métodos eficientes, orientados a conseguir un objetivo propuesto, son un conjunto de procesos que organizados dan una secuencia de pasos a seguir para obtener los hitos propuestos y finalmente el producto final. (FOWLER, 2003)

El Proceso Unificado de Rational

Es un proceso de desarrollo iterativo e incremental, en el sentido de que el software no es liberado de una sola vez al final del proyecto, sino que es desarrollado por partes. Cada iteración contiene todas las fases del ciclo de vida de un software: análisis, diseño, implementación y prueba.

El Proceso Unificado de Rational (RUP, por sus siglas en inglés) es en realidad un refinamiento realizado por Rational Software Corporation del más genérico Proceso Unificado, el cuál es a su vez una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza UML para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Entre sus características fundamentales están: (JACOBSON, 2000)

1. Guiado/Manejado por casos de uso.
2. Centrado en la arquitectura.
3. Iterativo e Incremental.
4. Desarrollo basado en componentes.
5. Utilización de un único lenguaje de modelación (UML).
6. Proceso Integrado.

Programación Extrema

La programación extrema (XP, por sus siglas en inglés) es un enfoque de la ingeniería de software formulado por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de

requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto y aplicarlo de manera dinámica durante el ciclo de vida del software. Las características fundamentales del método son: (ROSENBERG, 2001)

1. Desarrollo iterativo e incremental
2. Pruebas unitarias continuas
3. Programación en parejas
4. Frecuente integración del equipo de programación con el cliente
5. Corrección de todos los errores antes de añadir nueva funcionalidad.
6. Refactorización del código.
7. Propiedad del código compartida:
8. Simplicidad

ICONIX

ICONIX está entre la complejidad de RUP y la simplicidad y pragmatismo de XP, sin eliminar las tareas de análisis y diseño que XP no contempla. Al igual que RUP, es una metodología conducida por casos de uso, pero no incorpora tantos artefactos UML. Utiliza UML en sus etapas, de modo que se pueden seguir los requerimientos y adaptarse a nuevos cambios. El objetivo de esta metodología es buscar un subconjunto mínimo, pero suficiente de artefactos UML para hacer un buen trabajo de Ingeniería del Software en poco tiempo. (STEPHENS, 2007)

Las tres características fundamentales de ICONIX son:

- ✓ Iterativo e incremental: varias iteraciones ocurren en el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- ✓ Trazabilidad: cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos producidos.

- ✓ Dinámica del UML: la metodología ofrece un uso dinámico del UML, como los diagramas de casos de uso, diagramas de secuencia y de colaboración.

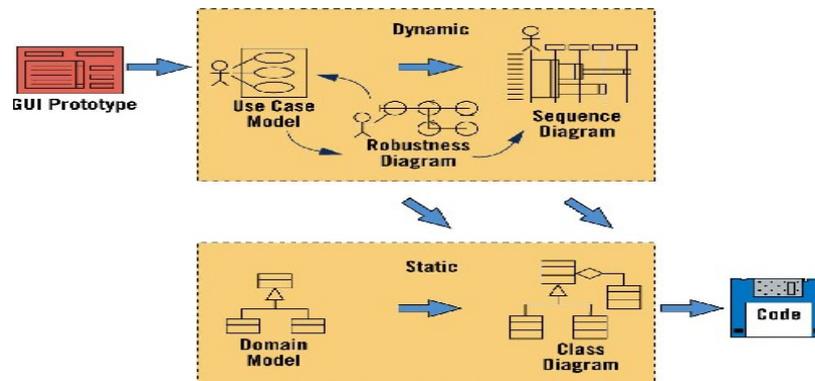


Figura 2: Metodología ICONIX

En la anterior Figura 2 se presenta el esquema general de la metodología ICONIX. En ella se muestran los principales modelos implicados y las fases generales. Las disciplinas que establece ICONIX para el desarrollo de un software son:

- ❖ Análisis de los Requerimientos
 - ✓ Modelo del Dominio
 - ✓ Prototipación rápida
 - ✓ Modelo de Casos de Uso

- ❖ Análisis y Diseño Preliminar
 - ✓ Descripción de Casos de Uso
 - ✓ Diagrama de Robustez

- ❖ Diseño
 - ✓ Diagrama de Secuencia

- ❖ Implementación
 - ✓ Escribir /Generar código

Entre sus principales ventajas se pueden definir las siguientes: (STEPHENS, 2005)

- ✓ El método utiliza un núcleo subconjunto de UML, pues se necesitan aprender 4 diagramas en vez de 14 que tiene RUP.
- ✓ Es rastreable de un paso al próximo.
- ✓ Tiene en cuenta los cursos básico y alterno.
- ✓ Funciona bien en un entorno ágil (iteración corta, pequeño incremento).
- ✓ Es una metodología práctica que ha probado funcionar en el mundo real, en miles de proyectos.
- ✓ Es adaptable a posibles cambios.

Resumiendo lo fundamental de ICONIX:

La metodología ICONIX resuelve el 80% de los desarrollos de software usando solo un 20% de los modelos definidos en UML. No descarta la utilización de ninguno de los modelos, en los casos que sea necesario, sino que define un conjunto mínimo de modelos y un proceso dinámico de desarrollo, utilizable en la mayoría de los casos.

La participación y el compromiso de los usuarios finales es uno de los pilares fundamentales de las metodologías ágiles, que permite verificar la completitud y el cumplimiento de los requisitos. Esto se logra en ICONIX con la participación de los usuarios en la prototipación temprana, en la descripción de los casos de uso y en las pruebas del sistema.

Además de lo planteado anteriormente, se escogió la metodología ICONIX porque es la que más se adapta a los requerimientos, pues siendo una metodología ágil, está menos orientada al documento y más centrada en el código fuente, exigiendo menor documentación y ahorrando en 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.

Al determinar la metodología de desarrollo de software para realizar el proceso de Ingeniería de Software se está listo para hacer una revisión de los lenguajes existentes para la web. Existe una multitud de lenguajes concebidos o no para Internet, cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para desarrollar

distintas aplicaciones. A continuación se presenta un breve resumen de algunos de estos lenguajes.

Lenguajes de programación Web.

En la programación para la Web se habla de dos tipos de lenguajes. Los lenguajes del lado servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Por otro lado, los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML), son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento.

La versatilidad de un lenguaje está íntimamente relacionada con su complejidad. Un lenguaje complicado en su aprendizaje permite en general el realizar un espectro de tareas más amplio y más profundamente. Es por ello que a la hora de elegir el lenguaje que queremos utilizar tenemos que saber claramente qué es lo que queremos hacer y si el lenguaje en cuestión nos lo permite o no. (Worsley, 2008)

PHP

Es un lenguaje para programar scripts del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. PHP es un lenguaje encapsulado dentro de los documentos HTML de forma que se pueden introducir instrucciones PHP dentro de las páginas, debido a esto, el diseñador gráfico de la Web puede trabajar de forma independiente al programador. Una Web dinámica con PHP contiene una serie de documentos de este tipo que el servidor Apache interpreta proporcionando al cliente documentos HTML.(Quiñones, 2009)

Este lenguaje presenta algunas desventajas entre ellas se puede mencionar que Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número. Se necesita instalar un servidor web. Además la legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP. Otra de sus limitaciones es que la programación orientada a objetos es aún muy deficiente para aplicaciones grandes y dificulta la modularización.

JavaScript

Brendan Eich, un programador que trabajaba en Netscape, adaptó otras tecnologías existentes (como ScriptEase) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995 y denominó inicialmente a su lenguaje LiveScript. Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. (Rumbaugh, 2009)

JavaScript se utiliza fundamentalmente para el desarrollo de páginas web dinámicas; es un lenguaje interpretado, es decir, que no requiere compilación y su sintaxis es semejante a la de los lenguajes Java y C.

El lenguaje está basado en objetos y no orientado a estos, el código se integra junto a HTML brindando diferentes efectos para la interacción con los usuarios. Los navegadores más populares como Internet Explorer, Mozilla Firefox, Opera, Netscape, entre otros interpretan el código JavaScript integrado dentro de las páginas web.

Java

Fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas WEB. Con la programación en Java, se pueden realizar distintos tipos de aplicaciones, como son los applets, las cuales son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor WEB y aplicaciones de escritorio que son programas que se ejecutan en forma independiente. (Zukowski, 2006)

Las características fundamentales que conllevaron a seleccionar como lenguaje de programación a Java se enumeran a continuación:

- ✓ Amplio respaldo de la comunidad Open Source: A diferencia de otros lenguajes de programación puramente comerciales, el código fuente de todas las APIs de Java está a la disposición de los programadores. Esta característica, potenciada por la política de abundante y buena documentación llevada a cabo por Sun Microsystems, hace que cualquier duda o proyecto de envergadura que se quiera afrontar en Java cuente con mayores facilidades.

- ✓ Programación completamente orientada a objeto: La concepción de Java de tratar cualquier componente como un objeto permite que el código, una vez generado, pueda ser empleado en otras aplicaciones, por lo que posee un nivel de reutilización bastante alto.
- ✓ Distribuido: Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- ✓ Interpretado y compilado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time).
- ✓ Seguro: Se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real.
- ✓ Potable: Especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.
- ✓ Multihebra: Soporta sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.

El uso de Java para el desarrollo de este proyecto es propicio, ya que supone un significativo avance en el mundo de los entornos software pues ofrece la potencia del diseño orientado a objetos con una sintaxis fácilmente accesible y un entorno robusto y agradable. Proporciona un conjunto de clases potentes y flexibles. Además pone al alcance de cualquiera la utilización de aplicaciones que se pueden incluir directamente en páginas Web. Fue diseñado para crear aplicaciones altamente fiables, una de las características fundamentales que debe tener la aplicación que se propone. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Además el framework que se utiliza está implementado en Java, el cual permite desarrollar aplicaciones ricas en interfaces con poca programación.

AJAX

AJAX es una técnica de desarrollo web para crear aplicaciones interactivas (RIA, por sus siglas en inglés). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de

los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax se ha convertido en una de las innovaciones tecnológicas con más crecimiento en los últimos tiempos. Este crecimiento ha traído la proliferación de una gran cantidad de frameworks que permiten hacer uso de este conjunto de tecnologías de forma fácil. (Richardson, 2008).

Con la intención de hacer la programación de la propuesta más fácil, cómoda y ágil se hizo una revisión de algunos frameworks implementados en Java. A continuación se mencionan algunos de estos con sus características más representativas.

1.4.2 Frameworks

Los Frameworks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Ellos ayudan en el desarrollo de software, proporcionan una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez. También ayudan a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación. Además simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. Un framework proporciona estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener.

ZK: Framework para el desarrollo de aplicaciones Web con Java. (Mitchel, 2009)

ZK es un framework de aplicaciones web en AJAX, completamente en Java de Código abierto que permite una rica interfaz de usuario para aplicaciones web y con poca programación.

El núcleo de ZK es un mecanismo conducido por eventos basado en AJAX, sustentado sobre 70 componentes XUL y 80 componentes XHTML, y un lenguaje de marcación para diseñar interfaces de usuario. Los programadores diseñan las páginas de su aplicación en componentes XUL/XHTML ricos en características, y los manipulan con eventos disparados

por la actividad del usuario final. Es similar al modelo de programación encontrado en las aplicaciones basadas en GUI de escritorio.

Además de la programación basada en componentes y orientación a eventos, de manera similar a Swing, ZK soporta un lenguaje de marcación para la definición de una potente interfaz de usuario llamada ZUML (Lenguaje de Marcación de Interfaz de Usuario ZK), el cual brinda las siguientes ventajas:

- ✓ Permite a los no expertos diseñar eficientemente interfaces de usuario.
- ✓ Embeber script en Java ayuda al prototipado rápido y personalizaciones.
- ✓ No hay requisitos de Ajax o JavaScript para los programadores.
- ✓ Modelo basado en componentes conducido por eventos intuitivo.
- ✓ Permite centrar toda la lógica de programación en el lado del servidor.

ZK posee un mecanismo que le permite al desarrollador construir aplicaciones usando el patrón Modelo Vista Controlador (MVC). Este mecanismo separa por capas el modelo, la vista y el controlador de una manera realmente genial.

La vista puede crearse haciendo uso de los lenguajes java o ZUML, presentando el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario. El modelo usa beans que contienen información de diversas fuentes de datos, siendo la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

El controlador responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Se crea al heredar de diferentes clases útiles que implementan de la interfaz, estas clases permiten un acceso directo a los componentes, objetos embebidos, y variables externas creadas en la vista (página zul), y la creación de métodos intuitivos.

Para el desarrollo de esta aplicación, el uso del framework ZK es favorable, pues al incluir un lenguaje de marcado como ZUML, permite a los no expertos diseñar eficientemente interfaces de usuario, y aunque está basado en Ajax no es necesario que el desarrollador tenga conocimientos de este. Posee un modelo basado en componentes intuitivo dirigido por eventos, y al permitir centrar toda la lógica de programación en el servidor no hay necesidad

de programar manualmente código para ejecutarse del lado del cliente, por lo tanto no se expone al cliente a la lógica del negocio de la aplicación.

Hibernate (Albiol, 2008)

Hibernate es un framework únicamente para la capa de persistencia de datos, es una herramienta de mapeo relacional de objeto para entornos java. El término ORM se refiere a la técnica de mapear una representación de datos desde un modelo de objetos a un modelo de datos relacional con un esquema basado en SQL. Es software libre, distribuido bajo los términos de la licencia GNU LGPL. La velocidad de desarrollo aumenta al utilizar un framework de este tipo, ya que los desarrolladores sólo deben preocuparse por escribir las clases que se mapearán y el framework se ocupa de hacer todas las operaciones de inserción, modificación, eliminación y consulta.

Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objeto (POO). Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hay un conjunto de factores que hacen de Hibernate una potente herramienta:

- ✓ **Productividad:** Usado conjuntamente con otras herramientas reducirán significativamente el tiempo de desarrollo.
- ✓ **Mantenibilidad:** Al tener menos líneas de código hace que el sistema sea más comprensible por lo los desarrolladores se centran en la lógica de negocio no en la conexión. Un sistema con menos código es más fácil de refactorizar.
- ✓ **Rendimiento:** Disponiendo de un ORM es más sencillo las optimizaciones, así algunos aspectos como consultas son más fáciles de optimizar que con código manejado vía SQL/JDBC.
- ✓ **Flexibilidad:** Hibernate puede integrarse en una arquitectura J2EE. Su flexibilidad permite configurarlo no sólo con JDBC, sino también con JNDI, JTA y JMX.

- ✓ Versatilidad en el acceso a BD relacionales: Las consultas a la Base de Datos se puede hacer de cuatro formas.
- ✓ Lenguajes de Consultas de Hibernate (HQL, por sus siglas en inglés): Es un lenguaje de acceso a base de datos donde las consultas se hacen a los objetos, no a las tablas, sólo vale para Hibernate.
- ✓ Ejemplos por Consulta (QBE, por sus siglas en inglés): Crea un objeto, rellena un campo con datos, y se crea una sentencia SQL con ese criterio.
- ✓ Consultas por Criterios (QBC, por sus siglas en inglés): Para criterios no sólo de igualdad. Ideal para buscadores pues se ahorra cantidad de tiempo y código.

El uso de este framework resulta favorable para la aplicación, pues la Base de Datos que se usa es relacional, el modelo relacional trata con relaciones, tuplas, conjuntos y es muy matemático por naturaleza. Sin embargo, el paradigma orientado a objetos trata con objetos, sus atributos y relaciones. Cuando se quiere hacer que los objetos sean persistentes utilizando para ello una base de datos de este tipo, hay una desavenencia entre estos dos paradigmas, la también llamada diferencia objeto-relacional, por tanto se hace uso de Hibernate para trabajar directamente con objetos simplificando enormemente la programación lógica de persistencia. Además permite integración con PostgreSQL que es el gestor de Base de datos que se utiliza. En resumen ofrece funcionalidades generales o de uso específico que contribuyen a un desarrollo más rápido y de una calidad superior.

La propia evolución de la informática en todos sus aspectos fue abaratando costes y simplificando procesos que de forma manual resultaban muy engorrosos. Los sistemas gestores de Base de Datos son una muestra de uno de estos trabajos para evitar el desgaste del hombre, este nos permite tener almacenada nuestra información manteniendo su integridad y protección para la manipulación de la misma, por lo cual se hace necesario realizar un análisis de lo antes expuesto para llegar a futuras conclusiones.

1.4.3 Sistemas gestores de base de datos (SGDB)

Un Sistema Gestor de Base de Datos es el software que posibilita la gestión de la información por parte de los usuarios, almacenada en una o varias Bases de Datos que constituyen un conjunto exhaustivo y no redundante de datos, estructurados de forma

independiente a su implantación o utilización en la máquina, accesible en tiempo real y compatible con usuarios recurrentes y sus respectivas necesidades de información. Su objetivo fundamental es el de suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, de manera que no sea necesario conocer el modo de almacenamiento de ellos en la PC, ni el método de acceso empleado. A continuación se mencionan algunos sistemas de gestión de base de datos:

MySQL

MySQL, es un sistema para la administración de bases de datos relacional (RDBMS) rápido y sólido creado por la empresa sueca MySQL AB. Está disponible desde 1996, pero su nacimiento se remonta a 1979. Ha obtenido el galardón Choice Award del Linux Journal Readers en varias ocasiones.

Es un sistema gestión de base de datos multiusuario y de subprocesamiento múltiple. Utiliza SQL el lenguaje estándar para la consulta de bases de datos utilizado en todo el mundo. MySQL es licenciado bajo la GNU GPL (del inglés, GNU General Public License). MySQL AB distribuye una versión comercial, que en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

Se puede utilizar en una gran cantidad de sistemas Unix diferentes así como en Microsoft Windows. Presenta APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl. El sistema de privilegios y contraseñas es muy flexible y seguro, permite verificación basada en el host. La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC. (Hillegas, 2008)

PostgreSQL

PostgreSQL es un sistema gestor de bases de datos objeto-relacionales (ORDBMS) libre, desarrollado originariamente en el Departamento de Ciencias de Computación de la Universidad de California en Berkeley.

Soporta casi toda la sintaxis SQL y ofrece muchas características modernas tales como: consultas complejas, integridad referencial (foreign keys), triggers, vistas (views), integridad

transaccional, control de concurrencia multi-versión. También soporta almacenamiento de objetos grandes (imágenes, sonido y video). (Quiñones, 2006)

Ofrece otras funcionalidades importantes tales como:

- ✓ Savepoints: permite hacer un roll back sin tener que repetir la transacción entera.
- ✓ Point in Time Recovery: permite salvar el estado de la DB en momentos concretos, para su posterior recuperación.
- ✓ Tablespaces: permite destinar discos físicos a un índice o a una tabla concreta.
- ✓ Improved Memory and I/O: optimizaciones en la velocidad de ejecución y en el consumo de memoria de la aplicación.

Cuenta con herramientas gráficas como PgAdmin, phpPgAdmin, las cuales hacen que la administración de la base de datos sea sencilla. Debido a su licencia libre PostgreSQL puede ser utilizado, modificado y distribuido por todo el mundo de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

Son múltiples las ventajas de este gestor de Base de Datos, entre ellas están:

- ✓ Modelos de negocios más rentables con instalaciones a gran escala.
- ✓ No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- ✓ Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
- ✓ Poder instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias, la principal preocupación de muchos proveedores de bases de datos comerciales
- ✓ Velocidad y rendimiento excepcionales
- ✓ Confiabilidad a toda prueba
- ✓ Seguridad de primera clase
- ✓ Flexibilidad para extenderse según se requiera
- ✓ Bajo Costo Total de Operación (TCO)

1.4.4 Servidores Web

Un servidor Web es un programa que implementa el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés). Este protocolo está diseñado para transferir hipertextos, páginas Web o páginas HTML, textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Un servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP o navegador. El navegador realiza una petición al servidor y este le responde con el contenido que el cliente solicita. Al solicitar una dirección de una página Web en un navegador, este realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página. El cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla.

Apache Tomcat

Existen varios servidores web, pero se mencionará el Apache Tomcat por ser el que mejores características posee. Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan: (Ditzel, 2008)

- ✓ Multiplataforma.
- ✓ Es un servidor de web conforme al protocolo HTTP/1.1.
- ✓ Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ✓ Basado en hebras en la versión 2.0.
- ✓ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ✓ Se desarrolla de forma abierta.

- ✓ Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

A continuación se mencionan las plataformas que se utilizaron para modelar e implementar la solución propuesta.

Plataforma Eclipse

Eclipse es una plataforma de programación usada para crear entornos integrados de desarrollo del (IDE, por sus siglas en inglés) de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". El IDE de Eclipse emplea módulos (del inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Eclipse posee características poderosas como editor de texto, resaltado de sintaxis, compilación en tiempo real, control de versiones con CVS y muy importante permite integración con Hibernate.

El uso de esta plataforma para el desarrollo de esta aplicación brinda numerosas ventajas pues se le puede incorporar gran número de plug-ins que se adaptan a las necesidades de este proyecto, como visual editor, plug-in oficial de Eclipse para la creación de interfaces gráficas e hibernate tools que proporciona a los desarrolladores la creación de scripts, facilitando el desarrollo web, y un entorno ampliado. Además trabajar con varios espacios de trabajo es agradable, de tal manera que logras organizar la carpeta del servidor en proyectos según la perspectiva. Otra de las ventajas es la compilación en tiempo real, pues conforme se escribe el código, avisa de los errores que se van cometiendo.

Visual Paradigm para UML

Visual Paradigm es una plataforma de modelado poderosa y fácil de usar que soporta el ciclo de vida completo del desarrollo de software de: análisis y diseño orientados a objetos, construcción, prueba y despliegue. Posee una intuitiva interfaz de usuario y permite dibujar todos los tipos de diagramas de clases, posibilitando que la creación de los diagramas o representaciones sea mucho más rápida que en otras herramientas CASE. Visual Paradigm posee elementos que la posicionan entre las herramientas más usadas en el mundo entre ellos: el soporte para UML 6.1, el modelado de procesos de negocio, la administración de requisitos, el modelado de bases de datos con la creación de diagramas Entidad Relación (ER), entre otros. Además es capaz de integrarse con diferentes plataformas de desarrollo de aplicaciones como Eclipse, así como con distintos gestores de bases de datos como: PostgreSQL.

Conclusiones parciales

Una vez expuestos los problemas actuales del objeto de estudio y las características de las aplicaciones relacionadas con el proceso que se desea automatizar, se evidencia la necesidad de desarrollar una aplicación web ya que es lo que más concuerda con las necesidades del sistema y por la necesidad inminente de realizar una eficiente gestión de la información profesoral, de forma tal que permita favorecer la toma de decisiones. Al desarrollo de la aplicación web se le incorpora la metodología ICONIX, para la posterior implementación del sistema se recomienda el uso del lenguaje de programación Java utilizando los frameworks ZK e Hibernate, con soporte de Base de Datos en PostgreSQL y como servidor Web el Apache Tomcat. Además se utilizan herramientas como la plataforma de modelado Visual Paradigm para UML 3.4 Enterprise Edition y la plataforma de programación Eclipse.

Capítulo 2: Sistema para la gestión de la información profesoral.

En el presente capítulo se presenta la solución propuesta por este trabajo a través de la metodología seleccionada: ICONIX. Para implantar esta metodología, primeramente se realiza un levantamiento de los principales requerimientos funcionales y no funcionales del sistema y seguidamente se diseña el modelo del dominio. Luego se muestra el diagrama de casos de uso del sistema, los cuales son descritos y se muestran algunas pantallas del sistema para su mejor comprensión. Se ilustran además los diagramas de robustez y secuencia para mostrar el desarrollo de las etapas de Análisis y Diseño y por último el estándar de código que se utilizará en la posterior etapa de implementación.

2.1 Análisis de los requerimientos:

Esta fase inicial tiene como objetivo principal el realizar la revisión de requerimientos del sistema. Incluye varias fases entre las que se encuentra el proceso de captura de requisitos, el cual se realiza mediante la comunicación con los clientes y los usuarios. Pues, los requisitos son las condiciones o capacidades que un sistema debe satisfacer, así como las especificidades de sus acciones. Además, se hace el análisis de requisitos para determinar si estos son confusos, incompletos, ambiguos, o contradictorios. Para obtener una lista de requerimientos funcionales, esta se enmarca en el contexto de pequeños párrafos donde se describen cada uno de ellos:

El sistema permitirá:

1. Visualizar los datos personales de cada profesor (nombre, apellidos, CI, teléfono, correo, dirección particular, municipio, provincia, sexo, cargo, disciplina, asignaturas impartidas, idiomas que domina, organización(si pertenece al PCC y CDR), la categoría docente (Instructor, Asistente, Auxiliar, Titular, Consultante) y categoría científica (M.Sc. y Dr.).
2. Gestionar profesores con todos sus datos.
3. Visualizar al jefe de departamento todas las actividades programadas la cual tiene asociada un nombre, fecha, hora lugar.
4. Gestionar las actividades.

5. Controlar la participación de cada profesor en cada una de las actividades.
6. Visualizar al jefe de Dpto. las publicaciones (la cual tiene asociada un título, una fecha reportada y fecha publicada) por cada profesor. Además cada publicación tendrá una categoría publicación, la cual puede ser: (En Cuba, En el extranjero, En BD, Web of Science, Monografía, Libro editado, Libro electrónico).
7. Gestionar las publicaciones.
8. Visualizar al jefe de Dpto. los Forum (título, fecha y curso) en que haya participado cada profesor. El forum tendrá un nivel forum que podrá ser: municipal, provincial o nacional y tendrá una categoría forum la cual podrá ser: mención, destacado o relevante.
9. Permitirá gestionar los Forum.
10. Visualizar al jefe de Dpto. los Premios (título y fecha) obtenidos por cada profesor. El premio tiene además un tipo (CITMA provincial, Otros Premios Nacionales, CITMA Joven Tecnólogo, CITMA Joven investigador, Innovación Tecnológica, Academia de Ciencias de Cuba). El premio tendrá asociado además un nivel premio (municipal, provincial y nacional).
11. Gestionar los Premios.
12. Visualizar al jefe de Dpto. los registros (título y fecha) de cada profesor. Los registros tienen asociado una categoría registro (informáticos, no informáticos, patente)
13. Gestionar los Registros.
14. Visualizar al jefe de Dpto. en cual Revista cada profesor tiene alguna publicación, la cual tiene asociada (nombre, título del artículo, y fecha). La revista tiene también un lugar de edición (extranjero o nacional). Además tendrá una categoría revista (indexada, no indexada, web of Science).
15. Gestionar las Revistas.
16. Visualizar al jefe de Dpto. los alumnos ayudantes que atiende cada profesor, de los cuales debe conocerse (nombre, apellidos, CI, brigada, carrera, índice académico y evaluación)
17. Gestionar los Alumnos Ayudantes.
18. Visualizar a qué tipo de plan pertenece cada carrera.
19. Visualizar al jefe de Dpto. las tesis diploma que ha tutorado cada profesor, de las cuales debe conocerse (título y tipo).
20. Gestionar las Tesis de Diploma.
21. Visualizar al jefe de Dpto. las tesis Postgrado que ha tutorado cada profesor, de las cuales debe conocerse (título y tipo).

22. Gestionar las Tesis de Postgrado.
23. Visualizar al jefe de Dpto. el GCE (nombre, tema de investigación, estudiantes que participan) al cual pertenece cada profesor.
24. Gestionar los GCE.
25. Visualizar al jefe de Dpto. a qué proyecto (nombre) pertenece cada profesor.
26. Gestionar los proyectos.
27. Visualizar las ARC por las cuales se evaluará a cada profesor.
28. Visualizar en forma de aviso el cumpleaños de cada profesor.
29. Generar reportes a CV de cada profesor.
30. Realizar búsquedas por diferentes criterios:
 - CI
 - Registros
 - Publicaciones
 - Revistas
 - Categoría científica
 - Categoría docente
 - Tutoría de tesis (pregrado o postgrado)
 - Premios
 - Forum
 - Proyecto
 - AA que atiende un profesor
 - Idiomas que domina
 - GCE

Luego de listar los requerimientos funcionales, deben tenerse en cuenta también los requerimientos no funcionales que se refieren a las características con las que debe contar el sistema, siendo estos de gran importancia para el mismo.

Apariencia o interfaz externa

- ✓ Diseño sencillo e intuitivo, permitiendo a personas de incluso poca experiencia, una mayor claridad y comodidad a la hora de trabajar con el sistema.

Navegabilidad y Usabilidad

- ✓ Interfaz amigable con facilidades para su uso por usuarios de cualquier nivel, reduciendo el tiempo de entrenamiento a los mismos.
- ✓ Funcionalidades visibles en todo momento que faciliten la navegación.
- ✓ Elementos organizativos para las funcionalidades como iconos y botones.
- ✓ El sistema brindará al usuario constantemente información referente al sitio en que se encuentra y posibilitará el acceso a otras páginas sin mucho esfuerzo ni consumo de recursos tanto humanos como computacionales.
- ✓ Familiarización con el entorno web.

Seguridad

- ✓ Acceso restringido. (Jefe de Departamento y sus profesores)
- ✓ Hacer copia de seguridad de la Base de Datos del Sistema

Ayuda y documentación en línea

- ✓ Inclusión de Manual de Usuario y Ayuda para el trabajo con el sistema.

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

- ✓ Multiplataforma

Hardware

- ✓ El servidor debe tener como mínimo un procesador Pentium III a 1 GHz, con 256 MB de memoria RAM, una tarjeta de red Protocolo Ethernet 10/100 MB/s.
- ✓ Los clientes deben tener un procesador gráfico de 16 bits o superior con una resolución de 800x600 píxeles o superior, procesador a 600 MHz y memoria RAM de 64 MB o superior. Dado que su principal objetivo es ser utilizado como una aplicación cliente-servidor, estas máquinas clientes deben estar conectadas en una red local de 10/100 MB

o deben tener un módem de línea telefónica con una conexión a 128 kbps, para facilitar el acceso al sistema y algunas funcionalidades de éste.

Software

- ✓ El sistema se desarrollará usando como lenguaje de programación Java y como gestor de Bases de Datos PostgreSQL, publicado en un servidor Web (Apache Tomcat).
- ✓ Todas las máquinas tienen que tener un navegador instalado, preferentemente de la versión Internet Explorer 5.0 o superior, pueden ser además: Mozilla, Mozilla FireFox, Netscape y Opera (versiones recientes). Para no entrar en contradicción con la política informática del país, debe recomendarse que el navegador sea libre.
- ✓ Para el funcionamiento del sistema en el servidor será necesario el S.O. Windows 95 o superior, o los sistemas de tipo UNIX (GNU/Linux) cuyo kernel sea estable.

Una vez analizados los requerimientos, se escogen las palabras claves (sustantivos) de cada uno de ellos, se conforma una lista de palabras, que más tarde serán utilizadas en el proyecto. Con estas palabras que se convertirán en clases y determinando las relaciones entre ellas se conforma el modelo del dominio.

2.1.1 Modelo de dominio.

El modelo del dominio es una parte esencial del proceso de ICONIX. Es la tarea de descubrir o identificar objetos (clases) en el mundo real y todas las relaciones de agregación y generalización entre ellos. Para ello se utiliza un diagrama de clases de alto nivel definido como modelo de dominio. Este es un artefacto vivo, colaborativo. El modelado del dominio es la tarea de construir un glosario o un diccionario de términos utilizados en el proyecto. Define el alcance para un proyecto, y forma la base en la cual se construyen los casos de uso. También proporciona un vocabulario común que posibilita una comunicación clara entre los miembros de un equipo del proyecto. (STEPHENS, 2007) A continuación se muestra lo antes expuesto a través de la figura 3 que representa el modelo del dominio.

La figura 4 representa la interfaz principal la cual le da la bienvenida al usuario explicando brevemente en qué consiste el sistema y su objetivo. Además le muestra los menuces por los cuales puede navegar luego de loguearse.



Figura 4: Interfaz principal

A continuación la figura 5 representa la interfaz Insertar Profesor la cual muestra la forma en que el usuario realiza la entrada de datos de un profesor. Se destaca que las demás interfaces de este tipo tienen un diseño similar.

Figura 5: Insertar Profesor

A continuación la figura 6 representa la interfaz Gestionar Profesores la cual permite visualizarle al usuario los profesores existentes hasta el momento en la Base de Datos. Muestra además los iconos que representan las operaciones que puede realizar en la interfaz (insertar, eliminar, actualizar) en que se encuentra y el icono de reportes que le permite mostrar todos los datos del profesor hasta el momento.



Figura 6: Gestionar Profesores

Con la lista de requerimientos y con las interfaces diseñadas según el usuario, el próximo paso es realizar el modelo de casos de uso del sistema (CUS).

2.1.3 Modelo de Casos de Uso.

El modelo de casos de uso comprende los actores, sistema y los propios casos de uso. Estos permiten a los usuarios estructurar y articular sus deseos, les obligan a definir la manera como querrían interactuar con el sistema, a precisar que informaciones quieren intercambiar y a describir qué debe hacerse para obtener el resultado esperado. Los casos de uso brindan una forma estructurada de capturar los requerimientos de comportamiento de un sistema, así que razonablemente se puede crear un diseño de ellos. Tienen como objetivo realizar un buen diseño orientado a objeto, lo que significa que permiten llegar rápidamente a un código de alta calidad. (Rodríguez y Torres, 2008)

A continuación la figura 7 representa el diagrama de CUS con sus principales requerimientos. En el se observa su principal actor que es el jefe de departamento

interactuando con las diferentes funcionalidades que el sistema ofrece. Para ver el diagrama general de CUS (Ver Anexo 3)

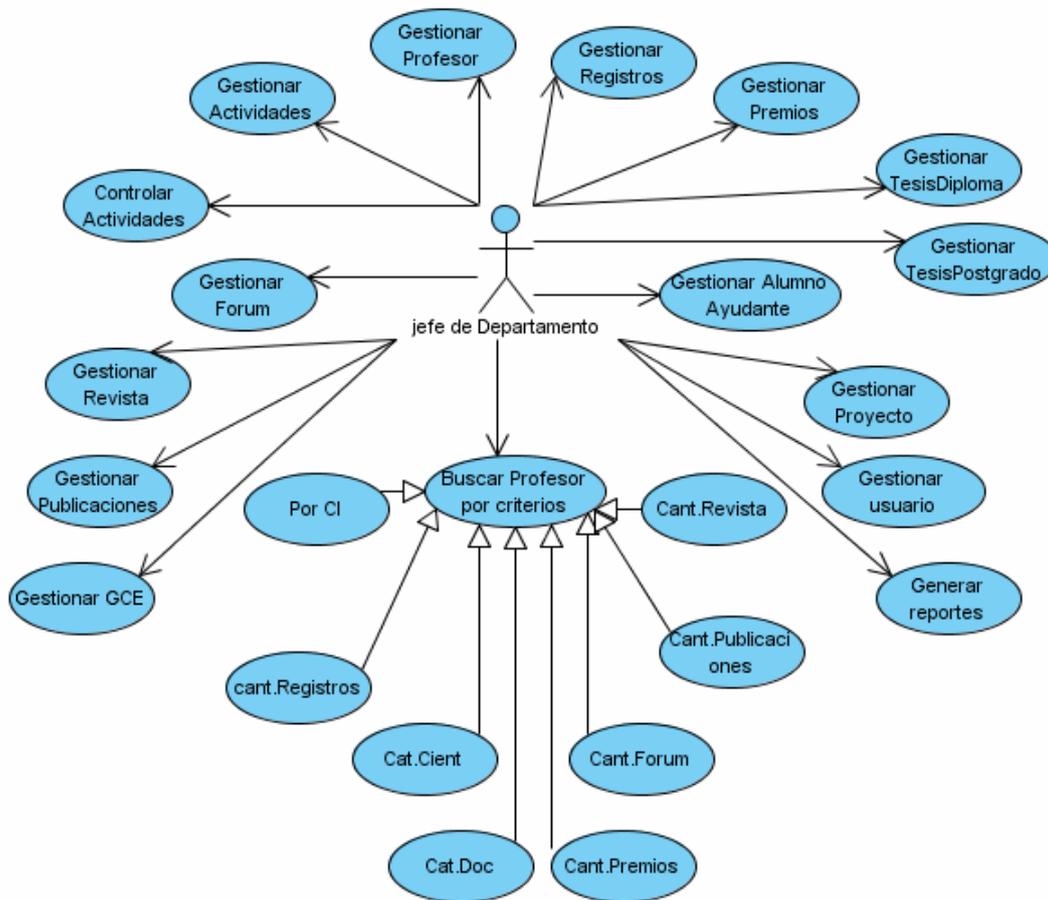


Figura 7: Diagrama de los principales CUS

Luego de culminar la primera fase de este proceso con el diagrama de CUS, se pasa a la siguiente fase de análisis y diseño preliminar.

2.2 Análisis y diseño preliminar

En la presente fase se describen para su mejor comprensión, los principales casos de uso del sistema, estos son ilustrados en forma de tablas y tienen en cuenta los cursos básicos y alternos donde se hace referencia a las verificaciones que realiza el sistema. Para culminar la fase se muestran algunos diagramas de robustez asociados a los principales casos de uso.

2.2.1 Descripción de los casos de uso

Estos deben guardar estrecha correlación con el manual de usuario y la interfaz grafica de usuario y deben corresponderse a lo que efectivamente hacen los usuarios. En resumen, los casos de uso permiten describir la utilización del sistema en el contexto del modelo de objeto. Los casos de uso deben ser descritos siguiendo la regla de los dos párrafos, en voz activa, usando un flujo respuesta/evento entre el usuario y el sistema, y utilizando una estructura de oración sustantivo-verbo-sustantivo. A continuación se describirán los CUS Insertar Profesor y Eliminar Alumno Ayudante ilustrados en las tablas 1 y 2 para su mejor comprensión. Para ver todas las descripciones de los CUS (Ver Anexo 1) y fichero adjunto.

Caso de Uso	Insertar Profesor
Curso básico	El jefe de departamento da un click en el icono Insertar de la interfaz Gestionar Profesores y el sistema muestra la Ventana Insertar Profesor con una serie de datos a llenar (nombre, apellidos, CI, fecha de nacimiento, correo, teléfono, sexo, dirección, idioma, asignatura que imparte, disciplina a la que pertenece, organización, categoría docente, categoría científica, cargo), al ser llenados da click en el Botón Insertar Profesor y el sistema almacena sus datos.
Curso alterno	<p>Ci no válido: El sistema verifica el formato de entrada del CI proporcionado (cantidad de caracteres, tipo de dato), en caso contrario muestra un mensaje alertando que el CI no es valido.</p> <p>Profesor repetido: El sistema verifica que el CI proporcionado no esté repetido, en caso contrario muestra un mensaje alertando que el profesor especificado ya existe en el sistema.</p>

Tabla 1: Descripción del CUS Insertar Profesor

Caso de Uso	Eliminar Alumno Ayudante
Curso básico	El jefe de departamento da un click en el icono Eliminar de la interfaz Gestionar Alumno Ayudante y el sistema muestra la ventana Eliminar Alumno Ayudante, el jefe de departamento selecciona de una lista de Alumnos Ayudantes el que desea eliminar, da click en el botón eliminar y el sistema lo elimina.
Curso alterno	No confirmación de eliminar: El jefe de departamento ya no desea eliminar el Alumno Ayudante seleccionado y da click en el botón cancelar y la ventana de mensaje se oculta.

Tabla 2: Descripción del CUS Eliminar Alumno Ayudante

2.2.2 Diagrama de robustez

Según la metodología ICONIX, para llegar desde los casos de uso a un diseño detallado (y luego al código), los mismos se necesitan vincular a los objetos. Aquí es donde entran los diagrama de robustez que se realizan con el objetivo de revisar las ideas de los casos de uso, verificar que tengan un comportamiento razonable y asegurar que en los casos de uso se cubran los caminos básicos y los posibles alternos. Los estereotipos del diagrama de robustez son: objeto interfaz, objeto entidad y controlador. (Rosenberg y Stephens, 2007)

A continuación se muestra la figura 8 que representa el diagrama de robustez asociado al CUS Insertar profesor. Para ver todos los diagramas de robustez asociados a los CUS (Ver Anexo 2) y fichero adjunto.

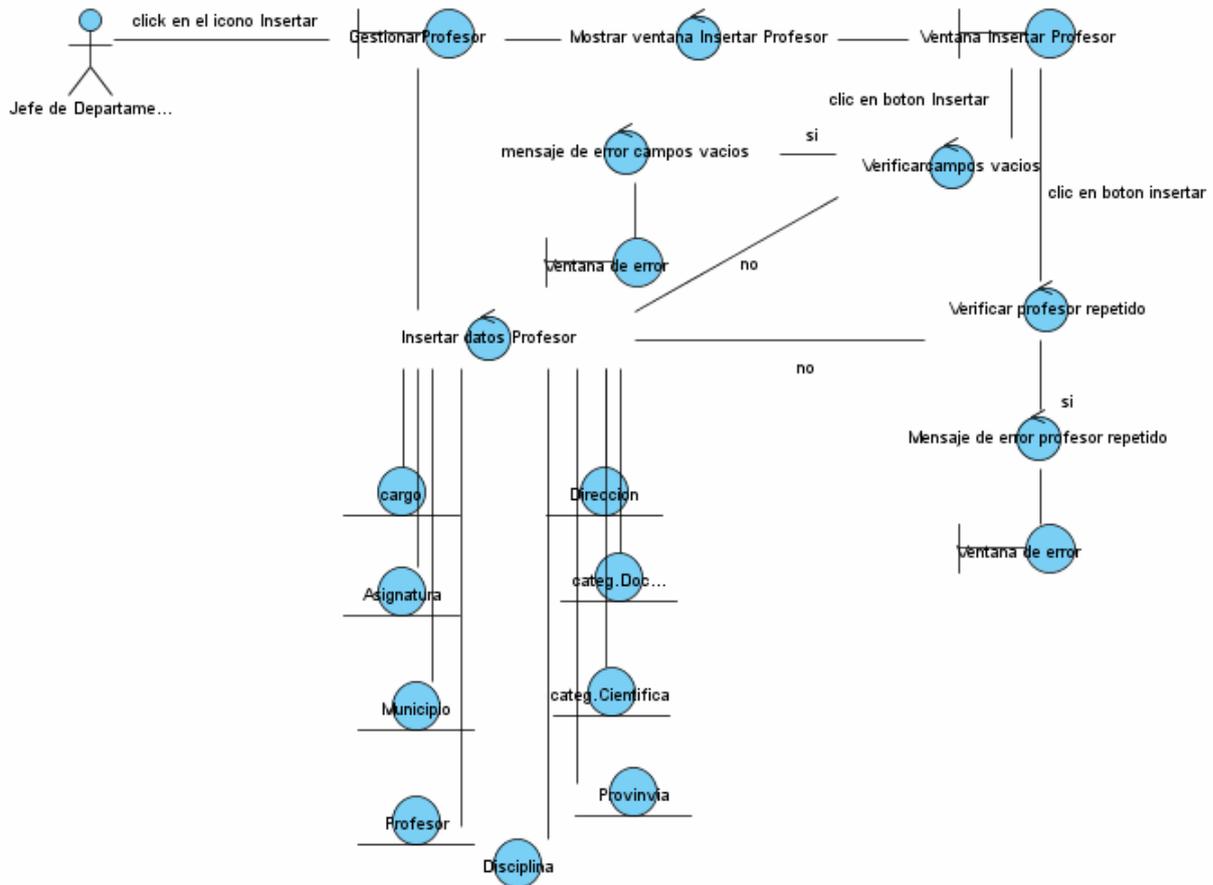


Figura 8: Diagrama de robustez Insertar Profesor

Al culminar esta fase no está de más dar una idea de cómo está estructurado el sistema, cómo funciona, a través de cuáles tecnologías, paquetes entre otros.

Arquitectura técnica

La arquitectura técnica describe el sistema que se está intentando construir en términos de estructura, es la concepción de las tecnologías utilizadas en el desarrollo de la aplicación, ya que a partir de ello se tiene una consideración general de lo que se está desarrollando. Incluye la topología del sistema (los nodos del servidor, localización física en la red, la elección del servidor de aplicación. (Rosenberg y Scott, 2001)

Dentro de la arquitectura técnica un paso muy importante es el diagrama de despliegue, el cual representa las principales capas que conforman la aplicación, siendo un modelo de objetos que describe la distribución física del sistema, además permite distribuir las funcionalidades entre nodos. Conjuntamente es una colección de nodos y arcos; donde cada

nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

A continuación se muestra la figura 9, la cual muestra el navegador web que es con quien el cliente interactúa, esto se hace a través de una página web desarrollada con el framework ZK. Todas las interfaces de la aplicación, es decir los ficheros.zul tienen un controlador asociado el cual es el encargado de capturar las diferentes acciones o peticiones del usuario. Posteriormente instancia el correspondiente objeto de acceso a datos (dao) quien mediante el empleo del framework Hibernate es quien actualiza, inserta, elimina y realiza todas las operaciones en la Base de datos PostgreSQL del sistema.

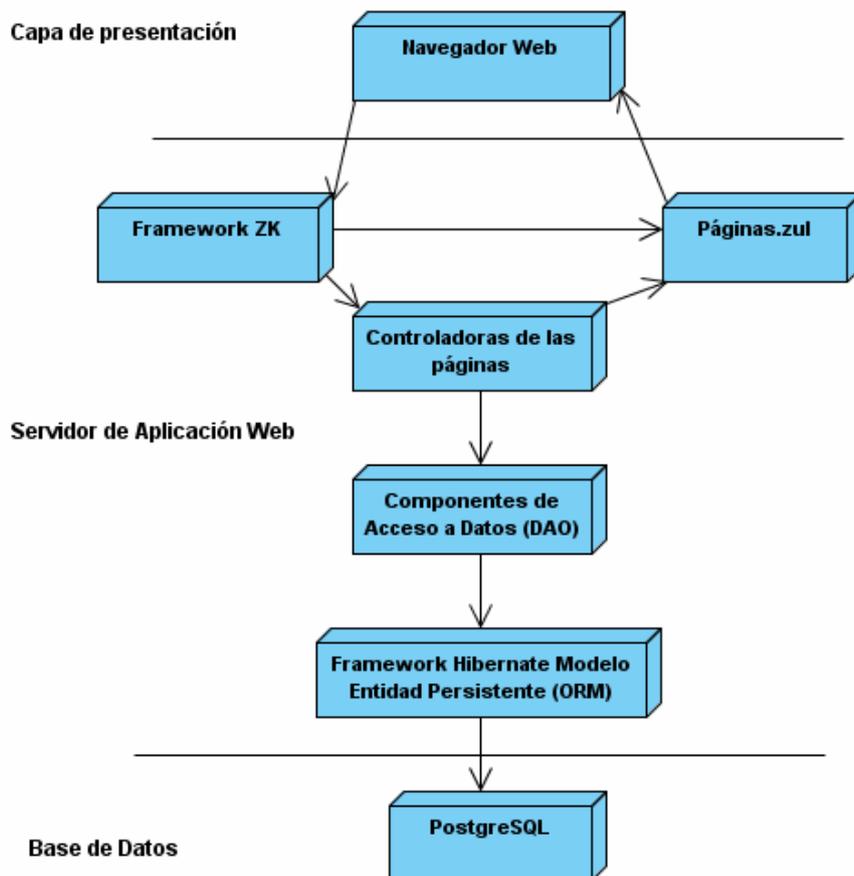


Figura 9: Modelo de Despliegue

En el desarrollo del sistema propuesto se tuvo en cuenta los principios de una arquitectura multicapa, lo cual aumenta la mantenibilidad y flexibilidad del sistema, pues a la hora de corregir, perfeccionar o adaptar el software resulta sencillo detectar en que lugar han de realizarse las modificaciones necesarias.

En la capa de presentación se tiene el navegador el cual ejecuta las páginas web. En el servidor de aplicaciones radica la lógica del negocio, la cual se distribuye en el framework ZK el cual es el encargado de gestionar los controladores correspondientes a las vistas de usuarios. Todas las peticiones que ocurren entre el navegador y el servidor de aplicaciones son asincrónicas, mediante el empleo de AJAX, lo cual implica que las páginas web en el navegador no se recarguen, aumentando el rendimiento y velocidad de la aplicación. Para separar la lógica del negocio de la presentación se empleó el patrón de diseño MVC (Model View Controller por sus siglas en inglés), para lograr separar las vistas del sistema y el código de implementación facilitando así su mantenimiento, un código más legible y fácil de entender.

Para el manejo del modelo de datos, se empleó el patrón de diseño DAO (Data Access Object por sus siglas en inglés), el cual aumenta la reusabilidad del código y establece un único punto de entrada para el acceso a las clases persistentes. La gestión de los datos almacenados en la BD se realiza mediante el framework ORM Hibernate, el cual reduce el tiempo de desarrollo de una aplicación pues abstrae al programador de utilizar sentencias SQL y gestionar las conexiones a las BD.

La próxima fase de Diseño muestra el comportamiento que tendrán las clases y cómo el usuario interactúa con ellas.

2.3 Diseño

El análisis y diseño preliminar solo era una teoría de cómo funcionarían las clases de la aplicación, al contrario que con el diseño detallado se logra asignar funciones a cada una de las clases que fueron detectadas. Además es donde se empiezan a ver que métodos llevarán las clases del sistema. Esto se debe que hasta ahora solo se ha interactuando con los objetos de las clases con los actores y con otros objetos de manera dinámica, por lo que se tiene suficiente información como para poder empezar a especificar los métodos de nuestras respectivas clases. En la presente fase se muestran los diagramas de secuencia asociados a los principales casos de uso del sistema, los cuales reflejan el comportamiento de las clases.

2.3.1 Diagrama de secuencia

Este diagrama es el núcleo del modelo dinámico y muestra todos los cursos alternos que pueden tomar los casos de uso. Se compone de 4 elementos: curso de acción, objetos, mensajes y métodos. Tiene tres objetivos elementales: Asignar comportamiento a las clases, mostrar en detalle como las clases interactúan entre sí durante el tiempo de vida del caso de uso y terminar la distribución de las operaciones entre clases. (Rosenberg y Stephens, 2007) Para ver todos los diagramas de secuencia asociados a los CUS (Ver Anexo 4) y fichero adjunto.

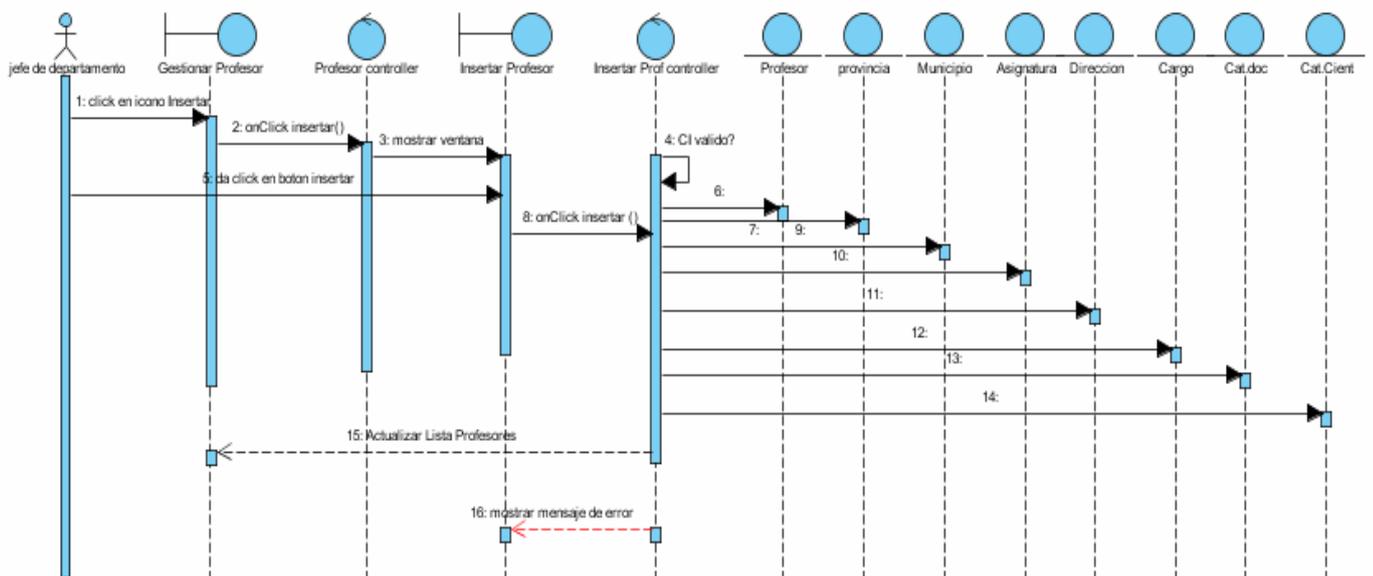


Figura 10: Diagrama de secuencia Insertar Profesor

2.3.2 Diagrama de clases del diseño

A partir de la evolución que ha ido sufriendo el modelo de dominio a lo largo de las fases anteriores se construye el diagrama de clases. Este artefacto permite encontrar errores en los diagramas de secuencia, métodos en clases equivocadas u otro tipo de anomalías. (Ver fichero adjunto)

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 datos. Para

determinar el modelo lógico de datos se hace necesario primero analizar cuales son las clases persistentes del componente a desarrollar, por lo se realiza el diagrama de clases persistentes, que modela la información que trasciende en el tiempo, incluso después de cerrada la aplicación. (Pressman, 1997). El diagrama de clases persistentes fue generado a partir del diagrama de entidad_ relación el cual fue utilizado para generar la Base de Datos del sistema.

Luego de finalizar la presente etapa de diseño se está listo para pasar a la posterior fase de implementación, en la cual se define el estándar de código utilizado y se tiene en cuenta además el tratamiento de errores. (Ver fichero adjunto)

2.4 Implementación

La implementación se llevó a cabo para lograr el éxito del sistema propuesto, en él se describen los elementos resultantes del diseño, además está enmarcado en la fase de construcción, y tiene como propósito principal desarrollar la arquitectura y el sistema como un todo. Ayuda a definir la organización del código, planificar las integraciones necesarias al sistema en cada iteración e implementar las clases durante el diseño. Una cuestión importante que influyó en la reducción del tiempo de implementación fue la estrategia de reutilización de código proveniente del mismo proyecto.

2.4.1 Estándares de código

Uno de los factores que definen si un software posee calidad es que el software sea fácil de mantener, el mantenimiento del software resulta ser la etapa más compleja y costosa en el proceso de desarrollo del mismo. Una de las tareas que implica el mantenimiento del software en su fase inicial es la comprensión del código que fue escrito por otras personas. Si el código está poco documentado y no se utilizó un estándar para desarrollarlo, implica que aumente la complejidad y el tiempo de mantenimiento.

Debido a lo antes expuesto para el desarrollo de la aplicación se hizo uso de un estándar de código.

Historial de versiones del documento:

Fecha	Versión	Descripción	Autor
18/01/20010	1.0	Estándar de Proyecto GIPDD	Yasmín Peña Godales

Tabla 3: Historial de versiones del proyecto

✓ Organización del código

Aspectos generales

Idioma: El idioma que se va a emplear para nombrar los distintos elementos va a ser el español.

Identación: La indentación va ser fija para todos elementos a 5 espacios.

Anidamiento: El anidamiento va a ser absoluto para cualquier tipo de instrucción a no más de 5 niveles (es).

Tamaño máximo de líneas: Las líneas de código no deben exceder los 70 caracteres. Por su parte, las de ruptura (continuación de una línea de código que excedió los 70caracteres) no deben exceder a 30 caracteres.

Módulos: Los módulos no deben contener más de 100 líneas de código.

Apertura y cierre de ámbito (begin-end, {-}, etc): Los ámbitos van a ser abiertos y cerrados en una línea aparte a la sentencia que los precede.

✓ Líneas y espacios en blanco

Líneas en blanco: Se deben usar 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: Se deben usar espacios en blanco antes y después de:

- Operadores lógicos.
- Operadores matemáticos.
- ✓ Organización del código (Ficheros)

La carpeta raíz del proyecto se va a llamar GIPDD. El sistema va a quedar subdividido en subsistemas. A continuación se informa de cada uno de estos subsistemas:

Comentarios: Los comentarios estarán escritos en español.

Comentarios de una línea: Se deben usar este tipo de comentarios cuando lo crea necesario.

Comentarios en bloque: Estos comentarios se aplicarán en los casos que se listan a continuación, y con la estructura mostrada:

Nombres

Nombre para Paquetes: Indicar propósito, que contienen letras, y la longitud de este no debe exceder los 25 caracteres.

Nombre para Unidades: Indicar propósito, que contenga letras, y la longitud de este no debe exceder los 12 caracteres.

Nombre para Tipos: Las letras de las palabras estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 12 caracteres.

Nombre para Interfaces : Las letras de la palabra estarán en Notación Camel. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 12 caracteres.

fIns : Formularios de insertar

fAct : Formularios de actualizar

fG : Formulario para la gestión

Los componentes visuales deberán cumplir la siguiente norma:

Componente	Prefijo	Ejemplo
Button	Btn	Button bt Button btnEliminar;
Textbox	tBox	Textbox tBoxName;
Listbox	lBox	Listbox lBoxUserList;

Tabla 4: Norma de los componentes visuales

Se destaca que los íconos empleados a la derecha representan las diferentes operaciones a realizar por el usuario como Insertar, Actualizar y Eliminar, y los empleados a la izquierda representan el nombre de la interfaz en que está situado.

Nombre para Clases: Los nombres de las clases deben indicar propósito, estarán escritas siendo el estándar de la notación Camel de java, y la longitud de estas no debe exceder los 12 caracteres.

Nombre para Objetos: Van a mantener el mismo nombre de la clase de la que son tipo, quitándole el sufijo de clase.

Nombre para Funciones de Acceso: Las letras de la palabra estarán en Notación Camel. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 12 caracteres.

Nombre para Métodos: Las letras de la palabra estarán en Notación Camel. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 12 caracteres.

Nombre para Constructores: Las letras de la palabra estarán en Notación Camel. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 12 caracteres.

Nombre para Parámetros: Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 8 caracteres.

Nombre para Excepciones y sus objetos: Notación Camel. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Constantes: Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 8 caracteres.

Nombre para Arreglos: Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 8 caracteres.

Nombre para Variables locales: Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 8 caracteres.

Nombre para Contadores de ciclo: Se van a utilizar letras, comenzando por la i

Declaraciones e inicializaciones

Orden de declaración

Por niveles de visibilidad

Los niveles de visibilidad van a ser declarados en orden ascendente, por lo que, primero se declarará `private`, luego `protected`, y por último `public`).

Dentro de un nivel de visibilidad deben declarar primero los atributos y luego los métodos.

Grupos de elementos a ser declarados sin importar el orden:

Constructores y destructores: No importa el lugar donde serán declarados en el segmento de los métodos

Se debe:

- No redeclarar parámetros para que contenga un valor por defecto.
- Evitar muchos parámetros simples para los constructores.

Funciones sobrecargadas

Para estas funciones no va a importar el orden de declaración.

Declaración de funciones

Parámetros: El orden de declaración de los parámetros que no son por defecto va a ser en orden decreciente de importancia. Por otro lado, en caso de que los parámetros sean por defecto, serán declarados primero los más propensos a ser utilizados.

Adicionalmente para los parámetros se debe:

- Proveer nombres formales en la declaración.
- Mismo nombre en la declaración y en la instancia.
- Evitar paso de parámetros por valor.
- Usar parámetros por defecto en lugar de sobrecarga de funciones.

Aspectos de complejidad: Tiene que ser menor que 8. Mientras que la complejidad ciclomática extendida que 12. El número máximo de líneas que puede contener el cuerpo de una función es de 80, y la cantidad máxima de puntos de retorno va a ser 1.

✓ Base de Datos

El nombre de la BD será el mismo que el del proyecto GIPDD. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Campos: las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 12 caracteres.

Además de hacer referencia el estándar de código a utilizar en esta fase, también se tuvo en cuenta el tema de tratamiento de errores, lo cual le brinda al usuario información de qué está haciendo incorrecto y le permite rectificarlo.

2.4.2 Tratamiento de errores

En cada página de la aplicación se garantiza que cuando se produzca alguna excepción el usuario del sistema sea advertido mediante ventanas de mensajes que se muestran en la página. Estos mensajes también son mostrados en las interfaces de entrada de datos,

donde se valida que el usuario no entre datos incorrectos y que no deje campos en blanco entre otros. En la figura 13 se presenta una ventana de alerta que se genera cuando el usuario intenta acceder a un menú del sistema sin antes loguearse.

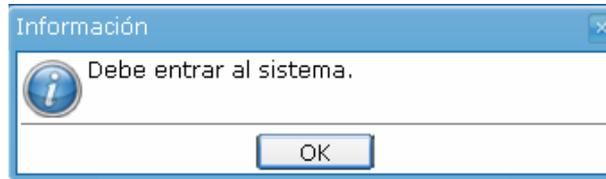


Figura 11: Ventana de error

Prueba

El modelo de prueba es la garantía de la calidad del software, en cual se desarrollan comprobaciones de manera sistemática a los componentes ejecutables en el modelo de implementación con pruebas de integración de sistema. La metodología ICONIX plantea que el proceso de prueba se debería comenzar mucho antes que el de codificación, incluso recomienda que se tenga en cuenta en la etapa de análisis identificando los casos de prueba a partir de los diagramas de robustez, de esta forma es posible eliminar una gran cantidad de errores (inclusive antes de que existan) . Es por ello que probar un sistema debe verse como una parte importante dentro del ciclo iterativo e incremental de desarrollo, que verifica que el producto cumple con el propósito específico para el que fue creado. (STEPHENS, 2007)

Teniendo en cuenta lo antes expresado se desarrollaron comprobaciones que permitieron probar de manera sistémica que el comportamiento del sistema descrito en cada caso de uso estaba implementado correctamente, básicamente las funciones identificadas durante el análisis de robustez. Estas pruebas se realizaron considerando en todo momento que los requerimientos estuvieran íntimamente relacionados con las mismas, se trató que al menos hubiera una prueba para verificar cada requerimiento. Los problemas detectados como resultado de las pruebas realizadas se tomaron en cuenta y a la vez fueron corregidos en iteraciones posteriores.

Conclusiones parciales

Se realizó claramente el flujo de trabajo de análisis y diseño del sistema, lo que posibilita el dominio de la solución que se propone y mejora la preparación para la implementación y prueba, presentado una vía completa y coherente por paquetes de los elementos específicos de la modelación Web. Se construyó el diagrama de clases persistentes, a partir del que se obtuvo el modelo de datos y se definieron todos los elementos que describen el modelo de implementación de la aplicación. Terminado el capítulo se pretende hayan quedado claros los conceptos manipulados y las funcionalidades del sistema GIPDD, de esta forma se preparan las bases para continuar con la construcción de la solución propuesta.

Capítulo 3: Validación de la solución.

Para llevar a cabo un buen proyecto de desarrollo de software, debemos comprender el ámbito del trabajo a realizar, los recursos requeridos, las tareas a ejecutar, las referencias a tener en cuenta, el esfuerzo (COSTE) a emplear y la agenda a seguir." (De la Fuente Moya, 1997) Dada su importancia en este capítulo se realizará un estudio de las sostenibilidad del producto informático y la planificación del costo del mismo, además de la validación de la hipótesis planteada mediante el método estadístico Delphy

3.1 Impacto administrativo

La confección de un Producto Informático trae consigo un Impacto Administrativo, el cual no es más que cambios producidos en la economía y gestión de recursos de una entidad. Cuando se diseña un producto informático se debe estar convencido de que éste repercutirá de forma positiva o negativa en el personal que esté en intercambio con él, por esto es preciso tener en cuenta las implicaciones en todos los órdenes que tendrá dicha aplicación. Aspectos como: tiempo de uso de los recursos, características físicas, económicas y mentales del personal sobre el cual el software incidirá de forma directa. Lograr obtener un software novedoso que satisfaga las expectativas del cliente es importante, pero más aún lo es hacerlo teniendo en cuenta siempre la premisa de no comprometer a la naturaleza y hacer un uso eficiente de los recursos.

Para determinar el costo se utilizó el modelo COCOMO II. (Ver Anexo 5)

✓ Valores calibrados utilizados:

A = 2,94; B = 0,91; C = 3,67; D = 0,28

✓ Cálculo del esfuerzo

El esfuerzo de desarrollo está representado por las siglas PM y se expresa en hombres por mes. La fórmula para obtenerlo se muestra a continuación:

$$PM = A * Size^E \prod_{i=1}^n EM_i \quad \text{Donde} \quad E = B + 0.01 * \sum_{j=1}^5 SF_j$$

$$E = 1.0698$$

PM \approx 44 hombres por mes.

- ✓ Cálculo del tiempo de desarrollo (TDEV)

El tiempo de desarrollo se obtuvo a partir de la fórmula:

$$TDEV = C * PM^F \quad \text{Donde} \quad F = D + 0.22 * (E - B)$$

TDEV \approx 13 meses.

- ✓ Cálculo de cantidad de hombres (CH)

La cantidad de hombres es el resultado de la división del esfuerzo y el tiempo estimado de desarrollo:

$$CH = PM / TDEV$$

CH \approx 4 hombres.

- ✓ Costo total (CT)

El costo total es el resultado de multiplicar el costo de hombres por mes (CHM) y el tiempo de desarrollo. Por tanto:

$$SP = \$ 100 \text{ Salario Promedio}$$

$$CHM = \text{cantidad de hombres} * SP$$

$$CHM = \$400 \quad \text{Costo de hombres por mes para 4 trabajadores}$$

Costo total: \$5200

$$CT = CHM * TDEV \quad CT \approx 5200.00 \$$$

Resumen

Cálculo	Valor
Esfuerzo (PM)	44
Hombres x mes	
Tiempo de desarrollo (m)	13
Costo(pesos)	\$ 5200

Tabla 5: Resumen

Análisis de costos y beneficios

El costo que puede ocasionar un producto informático es, sin dudas, uno de los más seguidos por la entidad que lo demanda. En este ámbito cabe destacar que el diseño de este sistema informático no es con fines comerciales, tanto el diseño como la posterior elaboración e implantación del producto resultante no originarán gastos notables, puesto que se ha diseñado y será implementado como parte de un trabajo de diploma. Su confección se realizará en función del equipamiento técnico con que se cuenta y no es necesario para su uso que se contrate un personal capacitado, lo que evita que se incurra en otros gastos por concepto de renovación o mejoramiento del equipamiento existente y contratación de personal.

El resultado que se espera con la implantación del sistema es el ahorro de electricidad asociado al funcionamiento del PI, este consumo no incrementará el ya existente en la empresa, debido a que el PI será implantado en máquinas ya existentes y de actual explotación. La calidad de la gestión de la información profesoral y el flujo de trabajo mejorará considerablemente una vez puesto en marcha el PI, debido a que la solución que brinda está orientada a dotar de confiabilidad e integridad a la gestión de información del proceso. La Universidad de Holguín contará con un sistema que le facilitará el flujo de información y los jefes de departamento podrán minimizar el tiempo de trabajo.

Como todo sistema de información, el PI apoyará la toma de decisiones por parte de los directivos. Además disminuye el tiempo de búsqueda de información profesoral. Finalmente para el desarrollo del PI se utilizarán herramientas que en su totalidad son libres, de modo que no es necesario pagar licencia por su uso.

Analizando el costo de desarrollar la aplicación contra los beneficios se puede concluir que su implementación es factible para la Universidad de Holguín.

3.2 Impacto Socio-Humanista

Cuando se hace alusión al Impacto Socio-Humanista, se hace referencia a cualquier alteración o cambio que se produce en los encargados del uso del producto informático. La implantación del sistema propuesto tiene un impacto social sustancial porque mejoraría las condiciones de trabajo de las personas involucradas en el sistema, dígase: Jefes de Departamentos (el mayor beneficiado) pues se benefician al contar con una herramienta que les permita realizar, con ayuda de la computadora, lo que antes realizaban manualmente.

Este sistema no prevé la racionalización de ninguna plaza, sino que permitirá agilizar el trabajo de los jefes de departamentos, que son los encargados de gestionar la información relacionada con los procesos que tienen lugar en el departamento docente, lo cual incurre en ahorro de tiempo. Se prevé la reducción del consumo de recursos como hojas y materiales de impresión, puesto que los informes creados podrán ser guardados digitalmente y por lo tanto pueden ser transferidos por correo electrónico o en soporte magnético. En cuanto a las posibilidades de generalización del producto informático, se puede plantear que puede ser explotado en cualquier institución universitaria donde se gestione información profesoral.

3.3 Impacto Ambiental

En cuanto al **Impacto Ambiental** se puede definir como cualquier alteración que se produzca en el medio ambiente al realizarse un proyecto o cualquier actividad humana. Para el diseño de la aplicación web no se utilizó ningún recurso perjudicial para el Medio Ambiente. La interfaz gráfica propuesta para la aplicación se concibió para que fuera sencilla, amigable, intuitiva y con contrastes agradables, los colores empleados son suaves y a la vez representan a la Universidad.

Se evitó el uso en exceso de colores cálidos, de manera que el diseño no es agotador y refresca la vista, debido a que el color predominante es el azul claro. Los sonidos empleados son escasos, agradables y presentan un volumen bajo, su principal función es apoyar determinada acción realizada por el usuario o mensaje emitido por el sistema, por lo que la molestia por ruido o sonido desagradable queda descartada. La interfaz está caracterizada por rasgos curvos y suaves, lo que transmite una sensación de tranquilidad y paz. Los colores empleados no son saturados, al contrario, predominan los colores claros evitando el cansancio o daño en la vista al usuario que visita el sitio.

Se tuvieron en cuenta las exigencias fisiológicas del ser humano, es decir, tamaño de letra, espaciamiento entre caracteres, tipografía, etc. Una buena navegabilidad limita el tiempo de búsqueda de información y evita daños en la columna vertebral de los usuarios. El nivel de reutilización es alto, pues los componentes pueden ser empleados en otros productos dirigidos a la gestión de la información profesoral.

3.4 Impacto Tecnológico

No es necesario que los usuarios tengan grandes conocimientos informáticos o reciban una preparación profesional para la implantación y uso del producto informático, pues cuenta con un manual de usuario que facilitan el trabajo con el mismo. La universidad cuenta con la infraestructura electrónica necesaria para la implantación y uso de la aplicación. Este producto está diseñado para admitir cambios tecnológicos, como mejoras en la red y hardware, así como cambios de plataforma.

La decisión tecnológica más apropiada asimilable con el usuario y el contexto para elaborar y aplicar el producto informático se describe a continuación:

- El servidor debe tener como mínimo un procesador Pentium IV a 2 GHz o más, con 512 MB de memoria RAM, una tarjeta de red Protocolo Ethernet 10/100 MB/s.
- Los clientes deben tener un procesador gráfico de 16 bits o superior con una resolución de 800x600 píxeles o superior, procesador Pentium III a 1 GHz y memoria RAM de 256 MB o superior. Dado a que su principal objetivo es ser utilizado como una aplicación cliente-servidor, estas máquinas clientes deben estar conectadas en una red local de

10/100 MB o deben tener un módem de línea telefónica con una conexión a 128 kbps, para facilitar el acceso al sistema y algunas funcionalidades de éste.

- El sitio se desarrollará usando como lenguaje de programación Java y PostgreSQL como gestor de Bases de Datos, publicado en un servidor Web (Apache Tomcat).
- Todas las máquinas tienen que tener un navegador instalado, preferentemente de la versión Internet Explorer 5.0 o superior, pueden ser además: Mozilla, Mozilla FireFox, Netscape y Opera (versiones recientes).
- Para el funcionamiento del sistema en el servidor será necesario el sistema operativo Windows XP o superior, o los sistemas de tipo UNIX (GNU/Linux) cuyo kernel sea estable.
- Entre los factores tecnológicos que constituyen un riesgo de vulnerabilidad se encuentran la ruptura o mal funcionamiento del servidor en que se encuentra hospedada la aplicación, la ruptura del equipamiento de las vías de comunicación digitales (cables de red, switches, hubs, etc.). Todo esto puede ser evitado mediante el cumplimiento de las medidas de seguridad informática y aplicando el mantenimiento periódico del hardware y software con que cuenta la organización. Además, se debe hacer hincapié en el hecho de que todas las herramientas concebidas para el desarrollo del sistema son software libre, así que no existen preocupaciones sobre legalidad de utilización.

A continuación se realiza un análisis de los resultados arrojados por la encuesta aplicada a los diferentes especialistas encargados de valorar el sistema. Para el procesamiento de estos fue empleado el Método Delphy, el cual permitió la emisión de un pronóstico que ratificó la validez de la solución.

3.5 Valoración del sistema por criterio de expertos.

El método Delphy es considerado como uno de los métodos subjetivos de pronósticos más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución estadística de las opiniones de expertos o usuarios en un tema tratado.

Fue creado con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política. Es la utilización

sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas o también la utilización sistemática del criterio práctico de un grupo de usuarios para conseguir un consenso de opiniones sobre la posible aplicabilidad de una propuesta determinada.(Rodríguez y Concepción, 2003)

Este método emplea una serie de variables que, a través de cálculos, permiten obtener un resultado final en cuanto a la relevancia de la solución.

La variable K_c (coeficiente de competencia) se calcula teniendo en cuenta la opinión del experto sobre su nivel de conocimiento acerca del problema que se está resolviendo y con las fuentes que le permiten argumentar sus criterios y se obtiene por la siguiente fórmula:

$$K_c = \frac{1}{2} (k_c + k_a)$$

Donde

K_c : Es el coeficiente de competencia.

k_c : Es el coeficiente de conocimiento o información que tienen el experto acerca del problema, calculado sobre la valoración del propio experto en una escala de 0 a 10 y multiplicado por 0,1.

k_a : Es el coeficiente de argumentación de los criterios del experto, obtenidos como resultado de la suma de los puntos obtenidos a través de una tabla.

El coeficiente de competencia se valora cualitativamente de la siguiente manera:

- $0,8 \leq K_c \leq 1$ Coeficiente alto
- $0,5 \leq K_c < 0,8$ Coeficiente medio
- $K_c < 0,5$ Coeficiente bajo

Tras seleccionar los encuestados con coeficiente de competencia alto (pueden seleccionarse, también, aquellos expertos cuyo coeficiente calculado haya sido medio) se construye una tabla donde se reflejan el total de respuestas por aspectos consultados y categorías señaladas, obteniéndose la tabla de frecuencia absoluta. Con los resultados de esta última se obtiene la tabla de frecuencia acumulada.

El tercer paso consiste en construir la tabla de inverso de la frecuencia absoluta acumulada, culminando con la tabla del inverso de la frecuencia absoluta de corte.

El procesamiento estadístico anteriormente descrito permite arribar a conclusiones sobre cada uno de los aspectos abordados en las encuestas.

Análisis de los resultados obtenidos en la encuesta a los posibles usuarios del sistema.

Para evaluar el grado de relevancia de las posibilidades que brinda la aplicación Web “Sistema para el control del trabajo y resultados de los profesores de la UHo” propuesto en esta investigación, se aplicó el método estadístico Delphy.

El primer paso fue seleccionar una cantidad de profesionales con experiencia en el tema informatizado por la aplicación propuesta, de forma que constituyeran el grupo de experto que la evaluaría. Esto expertos fueron debidamente elegidos pues para ello se realizó una encuesta y a partir del procesamiento de la misma se obtuvo el coeficiente de competencia de cada experto seleccionándose sólo los de alta competencia para un total de 17 expertos.

Una vez seleccionado los expertos se procedió a la realización de la encuesta de evaluación del sistema. La misma se confeccionó a partir de los siguientes indicadores generales de evaluación:

1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?

8. ¿Como evalúa la protección de la información manipulada en el sistema?

Después de haberse realizado el procesamiento de la encuesta se obtuvo que todos los aspectos resultaron muy adecuados. Teniendo en cuenta que el criterio de experto se empleó para validar la hipótesis, podemos afirmar que esta ha sido cumplida.

Conclusiones parciales

En este capítulo se ha calculado un estimado del costo para el desarrollo del producto informático que se ha diseñado, además de realizarse un detallado estudio de factibilidad y valoración de sostenibilidad del mismo. El producto informático que se ha diseñado, teniendo en cuenta sus características, queda clasificado como sostenible, y de acuerdo con los resultados ofrecidos por el estudio de factibilidad correspondiente, su implementación resultará factible para la Universidad de Holguín. Teniendo en cuenta que el criterio de experto se empleó para validar la hipótesis, podemos afirmar que esta ha sido cumplida siendo este un elemento más a favor del impacto que tendrá la implantación del mismo.

Conclusiones Generales

Con el desarrollo del Sistema para la gestión de la información profesoral en los departamentos docentes de la UHO se da cumplimiento al objetivo de la investigación, obteniéndose un software que favorece el desempeño y calidad de trabajo de los jefes de departamento docente. Los resultados más relevantes se mencionan a continuación:

- ✓ La aplicación web desarrollada mejora la calidad del trabajo y satisfacción de los jefes de departamentos, agilizando a la vez el proceso de control a los profesores.
- ✓ Según todo lo planteado en este trabajo, se considera que la hipótesis inicial fue cumplida, donde se planteó la elaboración de una aplicación Web que gestione toda la información profesoral generada en un departamento docente de una forma eficiente, rápido en las respuestas, realice búsquedas por diferentes criterios con alto grado de confiabilidad y brinde integridad en los datos. Se valoró el cumplimiento de la hipótesis a través del criterio de experto.
- ✓ El sistema es sostenible según las dimensiones administrativa, socio-humanista, ambiental y tecnológica, por lo que resulta factible para la Universidad y trae consigo el mejoramiento de la calidad de vida del ser humano.
- ✓ El sistema puede ser empleado en cualquier institución donde se gestione información profesoral.
- ✓ El diagrama de despliegue de la aplicación, permitió conocer la arquitectura del sistema, aumentando la mantenibilidad y flexibilidad del mismo.
- ✓ La metodología y tecnologías empleadas en la presente investigación, por los resultados obtenidos, demuestra su validez para el desarrollo de sistemas de este tipo.

Recomendaciones

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

- ✓ Continuar el desarrollo de este sistema incorporándole nuevas funcionalidades.
- ✓ Trabajar en la implantación del mismo en los departamentos docentes de la Uho.
- ✓ Incorporar al sistema consultas estadísticas por diferentes criterios por ejemplo: cálculo de promedio de publicaciones por profesor, participación en eventos, premios obtenidos entre otros.
- ✓ Trabajar en la integración del sistema propuesto con otros sistemas que se desarrollan en el marco del proyecto de gestión integral de la Uho.

Referencias bibliográficas

[1] Albiol, F. R. "Hibernate". URL: <http://www.javahispano.org>. 2008. febrero del 2010.

[2] Andrade, R. "Sistemas Gestores de Bases de Datos". URL: <http://www.de.sarrolloweb.com>.

[3] Carcasses, F. G. P. "Gestor de información en departamentos docentes del ISMM,". 2008.

[4] Davis, G. "Web Oficial de PHP". URL: <http://www.php.net>.

[5] De la Fuente Moya, A. "Modelo de estimación de costes para proyectos de software". 1997.

[6] Ditzel, C. "Apache." URL: < <http://www.apache.org> 2008. febrero del 2010.

[7] Endrinal, Y. H. y Martínez, Y. G. "Sistema automatizado para la gestión de la información de ciencia e innovación tecnológica de una universidad." 2008.

[8] Estevez, R. G. "Sistema de Gestión Académica Instituto Superior Politécnico José Antonio Echeverría (CUJAE)". URL: <http://www.cujae.edu.cu/eventos/convencion/cittel/Trabajos/CIT005.pdf> 2003. Febrero del 2010.

[9] FOWLER, M. "La nueva metodología". URL: <http://www.programacionextrema.org/articulos/newMethodology.es.html>. 2003. febrero del 2010.

[10] Hayes, D. "Javascrip_Aspectos y características generales" URL: www.htmlpoint.com.

[11] Hillegas, R. "MySQL.es". URL: <http://www.mysql.com> 2008. febrero del 2010.

[12] JACOBSON, I. "El Proceso Unificado de Desarrollo de Software. " 2000.

Referencias bibliográficas

- [13] Lago, R. "Diseño y desarrollo de aplicaciones web". URL: http://www.bab_soft.com/disenio_desarrollo_aplicaciones_web.php.htm
- [14] MES. "Ministerio de la educación superior. Sobre la organización y funcionamiento del DEPARTAMENTO DOCENTE". 1977.
- [15] MES. "Ministro de Educación Superior por Acuerdo del Consejo de Estado. RESOLUCIÓN No. 128/2006 ". 2006.
- [16] MES. "RESOLUCIÓN No. 128/2006". 2006.
- [17] MES. "RESOLUCIÓN No. 210/07". 2006.
- [18] Mitchel, M. "FrameworkZK". URL: <http://www.developer.com> 2009. Febrero del 2010.
- [19] Monmany, J. "Aplicaciones Web". URL: <http://www.webvillage.info> 2008. 20 de febrero del 2010.
- [20] Pressman, R. "Ingeniería de Software. Un enfoque práctico". 1997.
- [21] Quiñones, E. "Característica de PostgreSQL. " 2006. URL: <http://www.apesol.org>.
- [22] Quiñones, E. "Web Oficial de PHP". 2009. URL: <http://www.php.net>.
- [23] Richardson, C. " Ajax:Un nuevo acercamiento a las aplicaciones web." URL: <http://www.maestrosdelweb.com/editorial/ajax/>. 2008. febrero del 2010.
- [24] Rodríguez, C. B. y Salgado, R. U. "La Gestión en los departamentos docentes del MES. ¿Ciencia o Empirismo?". " URL: 2009. febrero del 2010.
- [25] Rodríguez, F. y Concepción, R. "El Método Delphy para el procesamiento de los resultados de encuestas a expertos o usuarios en estudios de mercado y en la investigación educacional. " URL: 2003.

Referencias bibliográficas

- [26] Rodríguez, M. y Torres, M. "Sistema de gestión de información editorial. Semanario ¡ahora!" Tesis de Diploma. Universidad "Oscar Lucero Moya. Holguín, 2008.
- [27] Rosario, J. "TIC: Su uso como Herramienta para el Fortalecimiento y el Desarrollo de la Educación Virtual". URL: <http://www.cibersociedad.net/archivo/articulo.php?art=22> 2006. 2 de febrero de 2010.
- [28] Rosenberg, D. y Scott, K. "Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example". 2001.
- [29] Rosenberg, D. y Stephens, M. "Use Case Driven Object Modeling with UML. Theory and Practice". URL: 2007.
- [30] Rosenberg, D. "Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example". 176 p. 2001.
- [31] Rumbaugh, J. "Javascrip_Aspectos y características generales". URL: www.htmlpoint.com 2009. febrero del 2010.
- [32] Stephens, D. "Agile Development with ICONIX Process". 2005.
- [33] Stephens, D. "Use Case Driven Object Modeling with UML — Theory and Practice. " 2007.
- [34] UHO. "Objetivos de trabajo del curso 2009–2010, FACINF". 2009.
- [35] Vasconcelos, G. "Herramienta para la Extracción de Información de los Programas de Disciplinas ". URL: 2005. febrero del 2010.
- [36] Worsley, J. "Lenguajes para la web". URL: http://www.lenguajes_de_programacion.com 2008. febrero del 2010.
- [37] Zalgado, E. "Conceptos de arquitectura Cliente/Servidor". URL: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap6-1.HTML> 2009. febrero del 2010.

- [38] Zukowski, J. "Que es java ". 2006.

Bibliografía

1. ALVAREZ, S. Administración informática. Estimación de proyectos y estrategias. COCOMO II. En: Conferencia de Ingeniería de Software. Febrero 2002.
2. Astigarraga, E. El método Delphy. Universidad de Deusto. San Sebastián. España, 2002?
3. Bases de datos. Microsoft MSDN Estudiantes. <http://www.microsoft.com/spanish/MSDN/estudiantes/ssii/> (10/4/2007).
4. BOEHM, B. "An Overview of the COCOMO 2.0 Software Cost Model", Software Technology Conference, 1995. En
5. BOEHM, B. "Software Cost Estimation with COCOMO II" Prentice Hall PTR. 2000
6. Brito M., Salinas S. Actualización tecnológica e Internet en servicios de información. Revista Ciencias de la Información. Vol 34, No. 2, agosto, 2003.
7. Brito M., Salinas S. Actualización tecnológica e Internet en servicios de información. Revista Ciencias de la Información. Vol 34, No. 2, agosto, 2003.
8. Carro, J., Vilaragut L. Las tecnologías de información: soporte de la inteligencia corporativa. El caso de Internet. Instituto de Información Científica y Tecnológica (IDICT). La Habana, Mayo, 2000.
9. Cascante P., Fonseca M. Intranet: Una ventana a la comunicación interna de las organizaciones. Revista de Derecho y Tecnologías de la Información. No. 4-2006. UNED, Costa Rica, 2006.
10. Escalona, M., Koch, N. Ingeniería de Requisitos en Aplicaciones Web - Un estudio comparativo. Universidad de Sevilla. Sevilla, España, Diciembre de 2002.
11. File: //sunset.usc.edu/research/COCOMOII/Docs/stc.pdf, recuperado el 15-2-2000.
12. Jacobson Ivar, Booch Grady, Rumbaugh James. El proceso unificado de desarrollo del software. Addison Wesley editorial de Pearson Education.sa, 1999.

13. Jacobson, Ivar; Booch, Grady; y, Rumbaugh, James. "El proceso unificado de desarrollo de software". Editorial Félix Varela. La Habana, 2004. Tomos I, II.
14. Joyanes, Luis. Cibersociedad. Los retos sociales ante un nuevo mundo digital. s.l.: McGrawHill, 1997.
15. Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Primera Edición por Prentice Hall, Hispanoamericana S.A. 1999.
16. Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición por Prentice Hall.
17. MARRERO, D. Modelado de aplicaciones Web con UML. En: Conferencia de Ingeniería de Software, Diciembre 2002, ISPJAE (CEIS).
18. Nakhimovsky A., Myers T. XML Programming: Web Applications and Web Services with JSP and ASP. Apress. 2002.
19. Quatrani, Terry. Visual Modeling with Rational Rose 2000 and UML, Publisher Addison Wesley, Second Edition October 19, 1999
20. Wallace, Doug et al. 2002. Extreme Programming for Web Projects. Addison Wesley.

Glosario de Términos

Applets: Programas escritos en lenguaje Java que se incluyen en una página HTML. Son ejecutados por la Máquina Virtual de Java.

Base de Datos: Una Base de Datos consta de una colección de tablas que contienen datos y otros objetos, como vistas, índices, procedimientos almacenados y desencadenadores, que se definen para poder llevar a cabo distintas operaciones con datos. Los datos almacenados en una Base de Datos suelen estar relacionados con un tema o un proceso determinados, como por ejemplo, la información de inventario para el almacén de una fábrica.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación. Se distinguen de los programas (software) porque éstos le indican a los componentes mencionados lo que deben hacer.

Herramienta CASE: Herramientas utilizada para el desarrollo de proyectos de Ingeniería de Software.

HTML (Hypertext Markup Language): Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.

HTTP: Hypertext Transport Protocol: Conjunto de especificaciones para el intercambio de ficheros (texto, gráfico, imagen, sonido, vídeo) en la WEB

Interfaz de usuario: Interfaz a través de la cual un usuario interactúa con un sistema.

Interfaz: Parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Plug-ins: Programas de software que extienden las capacidades de los navegadores de Internet.

Php: Ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts con el objetivo de crear aplicaciones potentes. Se caracterizan por ser tecnología open source, y por su fácil desarrollo y mantenimiento. Existe mucha documentación en Internet.

RUP: Rational Unified Process (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

SQL: Structured Query Language. Lenguaje de preguntas estructurado, lenguaje que utiliza bases de datos para pedir información de las mismas.

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.

WEB (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

APACHE: Herramienta software libre para servir aplicaciones Web en PHP y MySQL.

Browser: Navegador. Herramienta para “Visitar” o visualizar el contenido de las páginas Web y sitios FTP en Internet.

COCOMO: Modelo para la estimación de costos de productos informáticos.

GNU: Proyecto que reside en la Fundación del Software Libre para la organización, control y difusión del software libre.

Open Source: Código abierto. Manera de nombrar también a las aplicaciones desarrolladas bajo el amparo de un producto con licencia GPL.

XP: Extreme Programming, metodología ágil para el desarrollo de software utilizada en proyectos de corto plazo y corto equipo.

Multiplataforma: Indica la capacidad o característica de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas. Por ejemplo la posibilidad de utilizar un programa o software determinado en sistemas Windows y Linux.

Framework: Un framework, es una estructura de soporte definida mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Ajax: Acrónimo de Asynchronous JavaScript And XML (javascript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications)

Cliente/Servidor: Arquitectura de sistemas de información en la que los procesos de una aplicación se dividen en componentes que se pueden ejecutar en máquinas diferentes. Modo de funcionamiento de una aplicación en la que se diferencian dos tipos de procesos y su soporte se asigna a plataformas diferentes.

Anexos

Anexo 1: Descripciones de algunos CUS

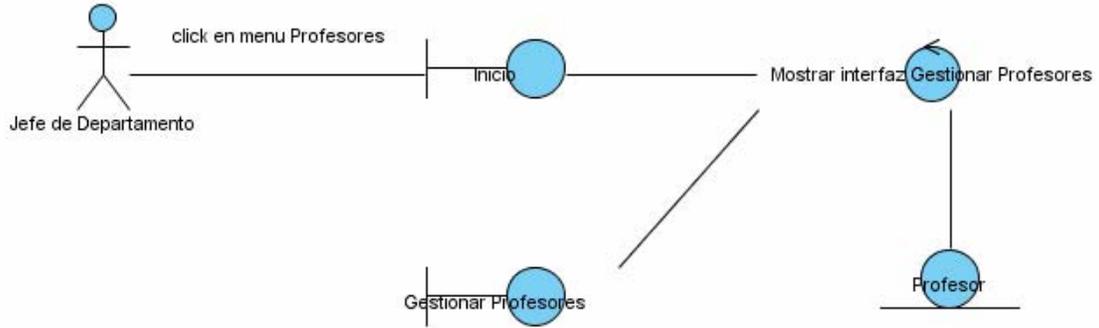
Caso de Uso	Visualizar Profesor
Curso básico	El jefe de departamento da un click en el menú Profesores de la interfaz Inicio y el sistema muestra una lista de todos los profesores existentes en la interfaz Gestionar Profesores.

Caso de Uso	Actualizar Profesor
Curso básico	El jefe de departamento selecciona el profesor que desea actualizar, da un click en el icono Actualizar de la interfaz Gestionar Profesores y el sistema muestra la ventana Actualizar Profesor con los datos del Profesor seleccionado, el jefe de departamento realiza los cambios a los datos del Profesor, da click en el botón Actualizar y el sistema guarda los cambios realizados.

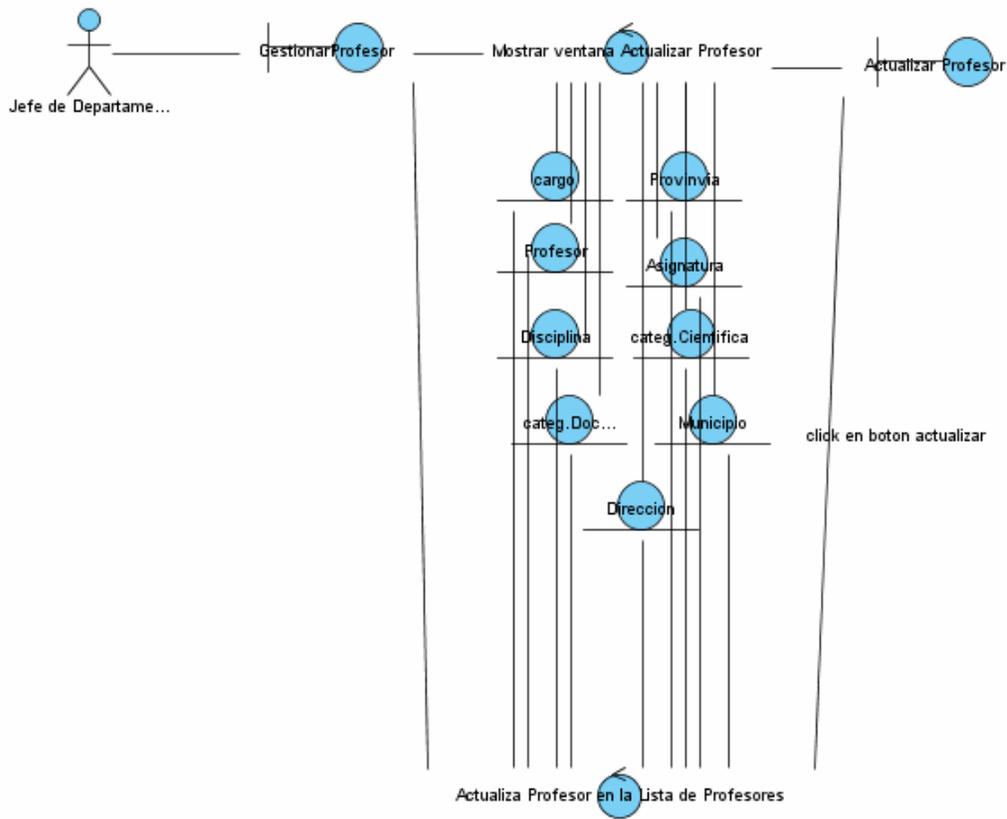
Caso de Uso	Eliminar Profesor
Curso básico	El jefe de departamento selecciona el profesor que desea eliminar, da click en el icono Eliminar de la interfaz Gestionar Profesores, y el sistema muestra un mensaje para confirmar la eliminación del profesor, el jefe de departamento da click en el botón aceptar de la ventana mensaje y el sistema elimina al Profesor seleccionado.
Curso alterno	No confirmación de eliminar: El jefe de departamento ya no desea eliminar al profesor seleccionado y da click en el botón cancelar y la ventana de mensaje se oculta

Caso de Uso	Buscar Profesor
Curso básico	El jefe de departamento da click en el menú Buscar Profesor de la interfaz Inicio y el sistema muestra la interfaz Buscar Profesor con una serie de criterios de búsqueda. El jefe de departamento selecciona el criterio por el cual desea realizar la búsqueda, da click en el botón buscar y el sistema muestra los resultados de la búsqueda.

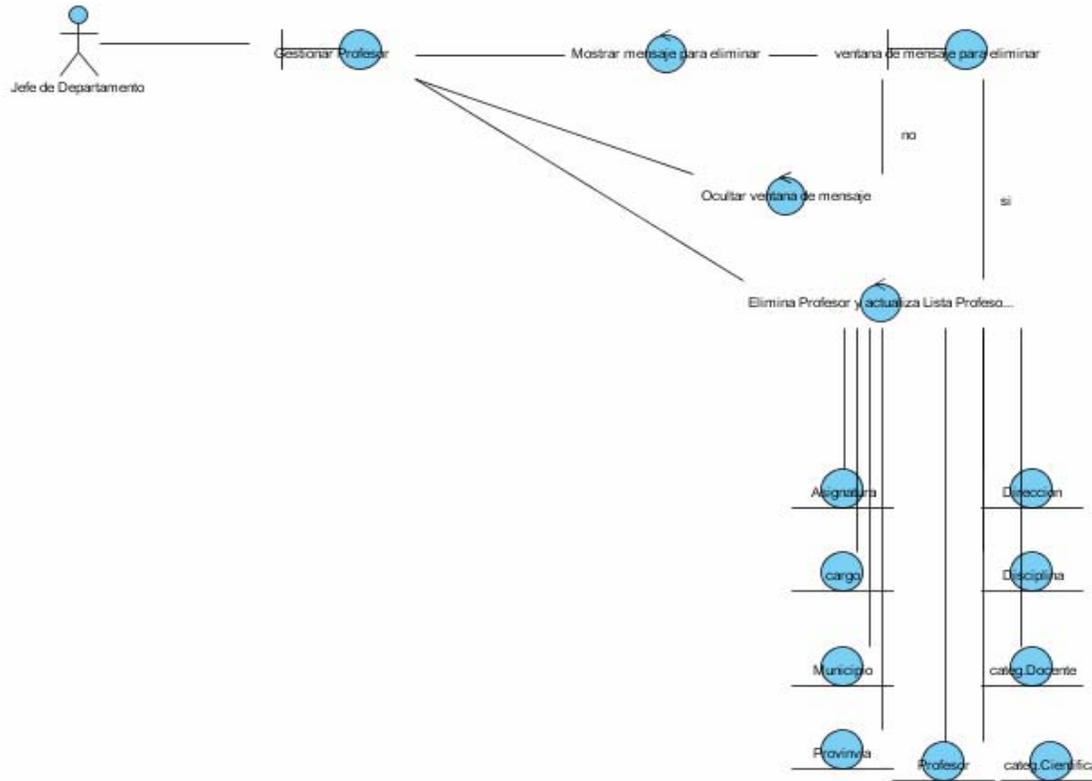
Anexo 2: Algunos diagramas de robustez asociados a los CUS



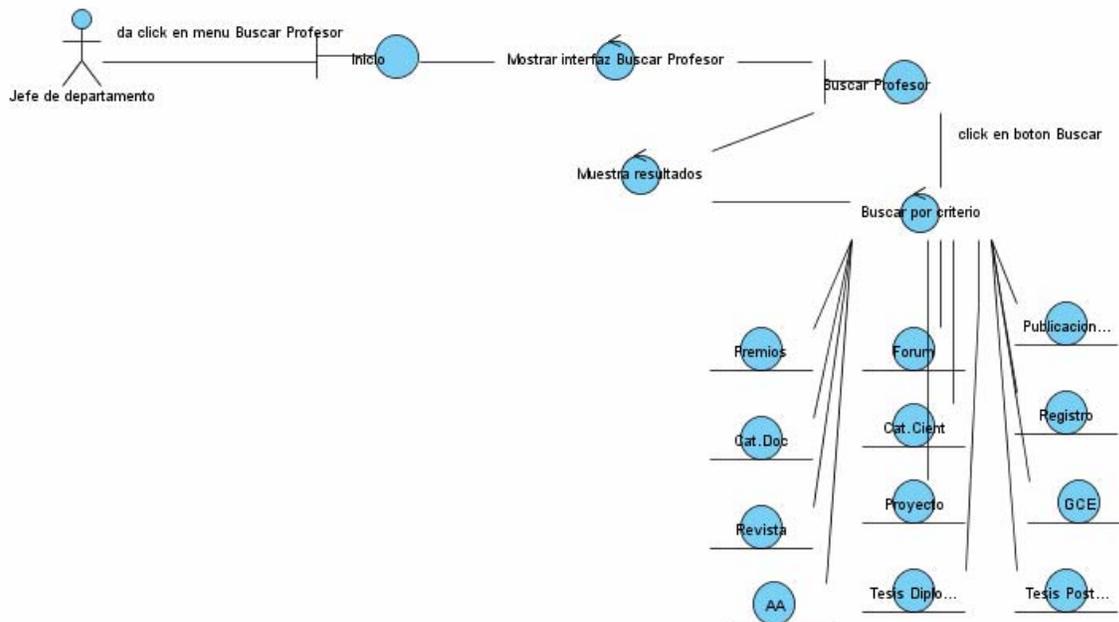
selecciona Profesor de la Lista y da clic en el icono actualizar

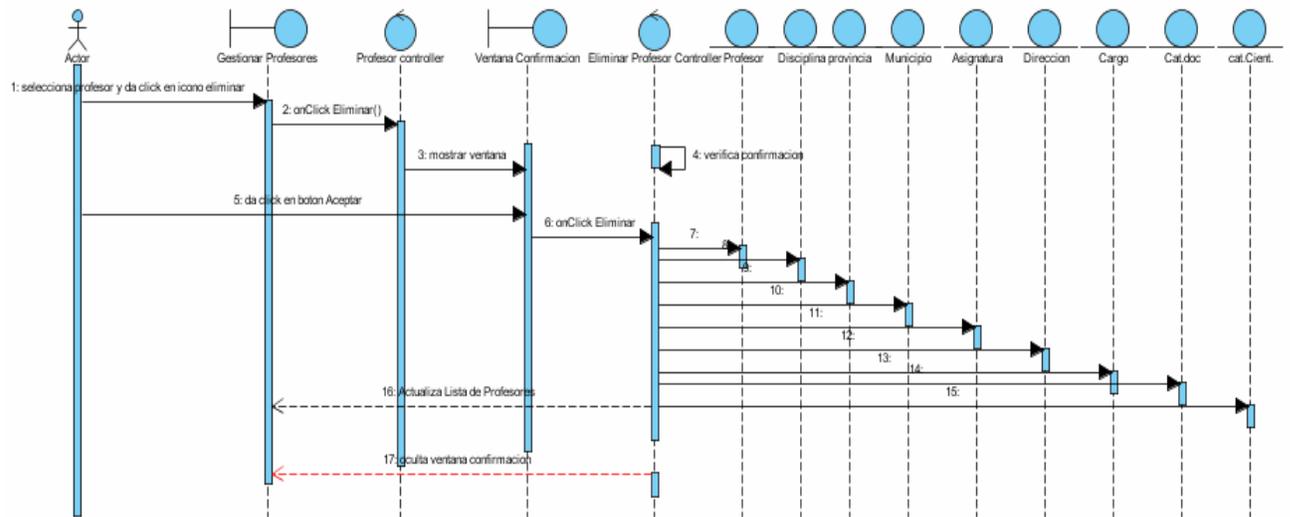


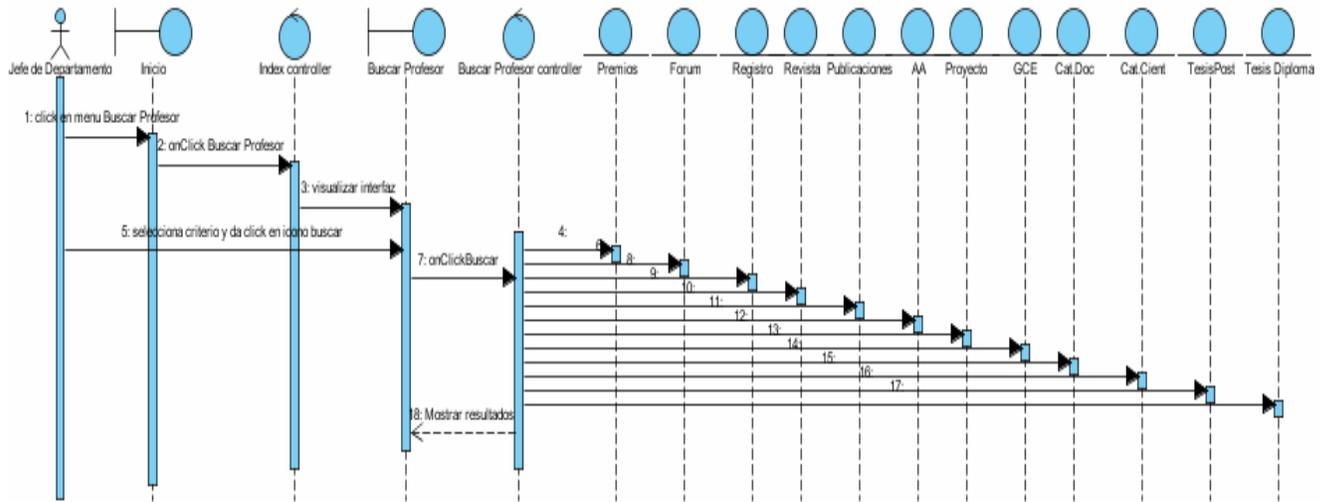
selecciona profesor de la Lista y da click en icono Eliminar



da click en menu Buscar Profesor







Anexo 5: Procedimiento COCOMO

Nombre de las entradas externas	Cantidad de ficheros	Cantidad elementos	Clasificación
Insertar profesor	12	25	Alto
Actualizar profesor	12	25	Alto
Eliminar profesor	12	25	Alto
Insertar Actividad	3	7	Medio
Actualizar Actividad	3	7	Medio
Eliminar Actividad	3	7	Medio
Controlar Actividad	2	3	Bajo
Insertar Tesis Diploma	4	6	Alto
Actualizar Tesis Diploma	4	6	Alto
Eliminar Tesis Diploma	4	6	Alto
Insertar Tesis Postgrado	4	6	Alto
Actualizar Tesis Postgrado	4	6	Alto
Eliminar Tesis Postgrado	4	6	Alto
Insertar Forum	5	7	Alto
Actualizar Forum	5	7	Alto
Eliminar Forum	5	7	Alto
Insertar Publicación	4	8	Alto
Actualizar Publicación	4	8	Alto
Eliminar Publicación	4	8	Alto
Insertar Registro	4	6	Alto
Actualizar Registro	4	6	Alto
Eliminar Registro	4	6	Alto
Insertar Revista	5	9	Alto
Actualizar Revista	5	9	Alto
Eliminar Revista	5	9	Alto

Insertar Premio	5	7	Alto
Actualizar Premio	5	7	Alto
Eliminar Premio	5	7	Alto
Insertar Proyecto	5	7	Alto
Actualizar Proyecto	5	7	Alto
Eliminar Proyecto	5	7	Alto
Insertar AA	7	10	Alto
Actualizar AA	7	10	Alto
Eliminar AA	7	10	Alto
Insertar GCE	4	5	Alto
Actualizar GCE	4	5	Alto
Eliminar GCE	4	5	Alto

✓ Salidas externas (EO)

Nombre de las salidas externas	Cantidad de ficheros	Cantidad de elementos	Clasificación
Visualizar listado Profesores	1	2	Bajo
Visualizar listado de Actividades	1	1	Bajo
Visualizar listado de Tesis de Diploma	2	1	Bajo
Visualizar listado de Tesis de Postgrado	2	1	bajo
Visualizar listado de Publicaciones	1	1	bajo
Visualizar listado de Forum	1	1	bajo
Visualizar listado de Registros	1	1	bajo

Visualizar listado de Revistas	1	1	Bajo
Visualizar listado de Premios	1	1	bajo
Visualizar listado de Proyectos	1	1	bajo
Visualizar listado de Alumnos Ayudantes	2	2	Bajo
Visualizar listado de GCE	1	1	bajo
Listar profesores por CI	1	3	bajo
Listar profesores por Carrera	2	3	bajo
Listar profesores por Disciplina	2	3	Bajo
Listar profesores por Cat.Cient	2	3	Bajo
Listar profesores por Cat. Doc	2	3	Bajo
Listar profesores por Idioma	3	3	Bajo
Listar profesores por Revista	2	3	Bajo
Listar profesores por Proyecto	2	3	Bajo
Listar profesores por GCE	2	3	Bajo
Listar profesores por AA	2	3	Bajo
Listar profesores por Registro	2	3	Bajo
Listar profesores por Tesis Diploma	3	3	Bajo
Listar profesores por Tesis Postgrado	3	3	Bajo
Listar profesores por Publicaciones	2	3	Bajo
Listar profesores por Premios	2	3	Bajo
Listar profesores por Forum	2	3	Bajo

✓ Ficheros internos (ILF)

Ficheros internos	Cantidad de ficheros	Cantidad de elementos de datos	de de	Clasificación
Actividad	1	6		Bajo
actividad_profesor	1	5		Bajo
Alumno_ayudante	1	3		Bajo
Arc	1	3		Bajo
Arc_profesor	1	6		Bajo
Asignatura	1	5		Bajo
Asignatura_profesor	1	3		Bajo
Brigada	1	2		Bajo
Cargo	1	2		Bajo
Cargo_profesor	1	4		Bajo
Carrera	1	2		Bajo
Categoría_cientifica	1	2		Bajo
Categoría_cientifica_profesor	1	3		Bajo
Categoría_docente	1	2		Bajo
Categoría_docente_profesor	1	3		Bajo
Categoría_forum	1	2		Bajo
Categoría_publicacion	1	2		Bajo
Categoría_publicacion_publicacion	1	2		Bajo
Categoría_registro	1	2		Bajo
Categoría_revista	1	2		Bajo
Clasificación_proyecto	1	2		Bajo
Curso	1	2		Bajo
Dirección	1	8		Bajo
Financista_proyecto	1	2		Bajo

Gce	1	3	Bajo
Gce_estudiante	1	2	Bajo
Gce_profesor	1	4	Bajo
Idioma	1	2	Bajo
Idioma_profesor	1	4	Bajo
Lugar_edicion_revista	1	2	Bajo
Municipio	1	3	Bajo
Nivel_forum	1	2	Bajo
Nivel_idioma	1	2	Bajo
Nivel_premio	1	2	Bajo
Nivel	1	2	Bajo
Org	1	2	Bajo
Plan	1	4	Bajo
Profesor	1	9	Bajo
Profesor_alumno_ayudante	1	5	Bajo
Profesor_org	1	2	Bajo
Provincia	1	2	Bajo
Proyecto	1	6	Bajo
Proyecto_profesor	1	3	Bajo
Publicación	1	5	Bajo
Publicaciones_profesor	1	3	Bajo
Registro	1	4	Bajo
Registro_profesor	1	4	Bajo
Responsabilidad	1	2	Bajo
Revista	1	5	Bajo
Revista_profesor	1	2	Bajo
Role	1	2	Bajo

Tesis	1	5	Bajo
Tesis_profesor	1	3	Bajo
Tipo_estudio	1	2	Bajo
Tipo_premio	1	2	Bajo
Tipo_premio_trabajo_premio	1	3	Bajo
Tipo_tesis	1	2	Bajo
Trabajo_forum	1	4	Bajo
Trabajo_forum_nivel_forum	1	4	Bajo
Trabajo_forum_profesor	1	3	Bajo
Trabajo_premio	1	3	Bajo
Trabajo_premio_profesor	1	4	Bajo
Usuario	1	3	Bajo

✓ Puntos de función

Elementos	Bajos	x Peso	Medios	x Peso	Altos	x Peso	Subtotal
ILF	64	7	0	0	0	0	448
ELF	0	0	0	0	0	0	0
EI	1	3	3	4	32	6	213
EO	28	4					112
EQ	0	0	0	0	0	0	0
						Total	773

✓ Características

Características	Valor
Puntos de función desajustados	773

Lenguajes	Java (75%), SQL (15%) SGML (10%)
Instrucciones fuente x puntos de lenguaje x lenguaje	Java:55, SQL:31 SGML: 58
Instrucciones fuente x lenguaje	Java: 31886.25 SQL: 3594.45 SGML: 4483.4
Total de instrucciones fuente	39.9641
Reducción 20% (Reutilización)(miles de instrucciones fuentes)	31.97128

Cálculo de esfuerzo, tiempo de desarrollo, cantidad de horas y costo.

✓ Multiplicadores de esfuerzo

Multiplicadores de Esfuerzo	Descripción	Valor
RELY	Confiabilidad	3
DATA	Tamaño de la BD	5
CPLX	Complejidad del Producto	4
DOCU	Necesidades de Documentación	4
RUSE	Reutilización	3
TIME	Tiempo de Ejecución	4
STOR	Almacenamiento	3
PVOL	Volatilidad de la plataforma	4
ACAP	Capacidad de los analistas	4
PCAP	Capacidad de los programadores	3

PCON	Continuidad del Personal	2
APEX	Experiencia de los Analistas	4
PLEX	Experiencia con la plataforma	3
LTEX	Experiencia con Lenguajes y Herramientas	3
TOOL	Uso de Herramientas de SW	3
SITE	Desarrollo en diferentes Sitios	3
SCED	Requerimientos de Cronograma	3

✓ Factores de escala

Factores de escala	Descripción	Valor
PREC	Precedencia	3.72
FLEX	Flexibilidad	2.03
RESL	Riesgos	5.65
TEAM	Cohesión del equipo	5.48
PMAT	Madurez de las capacidades	7.8

Anexo 6: Encuesta para la determinación de expertos

Usted ha sido seleccionado como posible experto para ser consultado respecto al grado de relevancia de las posibilidades que brinda la aplicación Web “Sistema para la gestión de la información profesoral en los departamentos docentes de la Universidad de Holguín.

Se necesita determinar su coeficiente de competencia en el tema, a los efectos de reforzar la validez del resultado de la consulta que se realizarán. Por esta razón se le solicita que responda las siguientes preguntas de la forma más objetiva que le sea posible.

1. Marque con una cruz (X), en la tabla siguiente, el valor que se corresponde con el grado de conocimientos que usted posee sobre el tema. Considere que la escala que se presenta es ascendente, es decir, el conocimiento sobre el tema referido va creciendo desde 0 hasta 10.

1	2	3	4	5	6	7	8	9	10

2. Realice una autovaloración del grado de influencia que cada una de las fuentes que le presentamos a continuación, ha tenido en su conocimiento y criterio sobre los indicadores que se han puesto a su consideración. Para ello marque con una cruz (X), según corresponda, en **A** (Alto), **M** (Medio) o **B** (bajo).

Fuentes de argumentación.	Grado de influencia de cada una de las fuentes.		
	A	M	B
Análisis teóricos realizados por usted.			
Su experiencia obtenida.			
Trabajo de autores nacionales.			
Trabajo de autores extranjeros.			
Su propio conocimiento del estado del problema en el extranjero.			
Su intuición.			

Anexo 7: Encuesta aplicada a los expertos para la valoración del sistema.

ENCUESTA DE OPINIÓN DE LOS USUARIOS:

Estimado compañero(a), con vistas a lograr la informatización del proceso de gestión de la información profesoral en los departamentos docentes de la UHo y por su experiencia, habilidades profesionales y conocimiento en el procesamiento de los datos referidos al proceso de gestión de información profesoral, se solicita su colaboración para hacer una valoración respecto el grado de relevancia de las posibilidades que brinda la aplicación Web “Sistema gestión de información profesoral en los departamentos docentes” por lo que la información que brinde será crucial para estos objetivos; rogamos que al responder esta pregunta lo haga de la manera más explícita posible. De antemano gracias.

1. A continuación se presentan diferentes indicadores de los cuales se requiere que usted evalúe. Para esta valoración se proponen las siguientes categorías:

MR: Muy relevante. **BR:** Bastante relevante. **R:** Relevante.

PR: Poco relevante **NR:** No relevante.

Marque con una X la categoría que considera adecuada para cada criterio:

Información que se puede registrar y controlar en el sistema.	Relevancia de la información				
	MR	BR	R	PR	NR
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?					
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?					
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?					
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?					
5. ¿Qué tan útil es el sistema como					

herramienta para la realización del trabajo de un jefe de departamento docente?					
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?					
7. ¿Como evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?					
8. ¿Como evalúa la protección de la información manipulada en el sistema?					

1. Otros aspectos que por su importancia quisieras plantear sobre el sistema.

Anexo 8: Resultados de la encuesta de opinión de los usuarios aplicando el método Delphy.

Tabla de frecuencia absoluta						
Criterios	MA	BA	A	PA	NA	TOTAL
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?	4	8	5	0	0	17
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?	14	3	0	0	0	17
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?	9	6	2	0	0	17
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?	10	5	2	0	0	17
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?	17	0	0	0	0	17
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?	8	9	0	0	0	17
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?	5	9	3	0	0	17
8. ¿Cómo evalúa la protección de la información manipulada en el sistema?	13	4	0	0	0	17

Criterios	MA	BA	A	PA	NA
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?	4	12	17	17	17
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?	14	17	17	17	17
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?	9	15	17	17	17
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema	10	15	17	17	17
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?	17	17	17	17	17
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?	8	17	17	17	17
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?	5	14	17	17	17
8. ¿Cómo evalúa la protección de la información manipulada en el sistema?	13	17	17	17	17

TABLA DEL INVERSO DE LA FRECUENCIA ABSOLUTA ACUMULADA				
	MA	BA	A	PA
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?	0,2353	0,7059	1	1
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?	0,8235	1	1	1
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?	0,5294	0,8824	1	1
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?	0,5882	0,8824	1	1
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?	1	1	1	1
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?	0,4706	1	1	1
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?	0,2941	0,8235	1	1
8. ¿Cómo evalúa la protección de la información manipulada en el sistema?	0,7647	1	1	1

TABLA DE DEREMINACIÓN DE LOS PUNTOS DE CORTES							
	MA	BA	A	PA	Suma	Promedio	N - Prom.
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?	-0,72	0,54	3,49	3,49	6,8	1,7	-1,7
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?	0,93	3,49	3,49	3,49	11,4	2,85	-2,85
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?	0,07	1,19	3,49	3,49	8,24	2,06	-2,06
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?	0,22	1,19	3,49	3,49	8,39	2,1	-2,1
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?	3,49	3,49	3,49	3,49	13,96	3,49	-3,49
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?	-0,07	3,49	3,49	3,49	10,4	2,6	-2,6
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?	-0,54	0,93	3,49	3,49	7,37	1,84	-1,84
8. ¿Cómo evalúa la protección de la información manipulada en el sistema?	0,72	3,49	3,49	3,49	11,19	2,8	-2,8

Conclusiones generales					
	MA	BA	A	PA	NA
1. ¿Se logró, en el sistema, incluir toda la información profesoral generada en un departamento docente?	Si	-			
2. ¿Cómo evalúa la utilización de los recursos para gestionar la información profesoral generada en un departamento docente?	Si	-			
3. ¿Cómo evalúa en cuanto a facilidad y comodidad el uso del sistema?	Si	-			
4. ¿Cómo evalúa los criterios de búsqueda implementados en el sistema?	Si	-			
5. ¿Qué tan útil es el sistema como herramienta para la realización del trabajo de un jefe de departamento docente?	Si	-			
6. ¿Cómo evalúa la forma en que la información es tratada respecto a uniformidad e integridad?	Si	-			
7. ¿Cómo evalúa que el sistema ayuda a no cometer errores en la captura y manipulación de la información profesoral?	Si	-			
8. ¿Cómo evalúa la protección de la información manipulada en el sistema?	Si	-			