

FACULTAD DE INFORMÁTICA Y MATEMÁTICA

**MÓDULO WEB PARA EL CONTROL DE PROYECTOS
DEL PORTAL CEPRONÍQUEL.**

Trabajo de diploma para optar por el título de Ingeniero Informático

Autor: Yaser Morera Rivera

Tutores: Ing. Mario Aballe Rodríguez

Dra.C. Rosa Isabel Urquiza Salgado

Holguín, Cuba

2011

Declaración de autoría

Declaro que soy el único autor de este trabajo, y que el mismo pertenece a la Facultad de Informática y Matemática de la Universidad de Holguín “Oscar Lucero Moya” para que haga el uso que se estime pertinente con él.

Firma del Autor

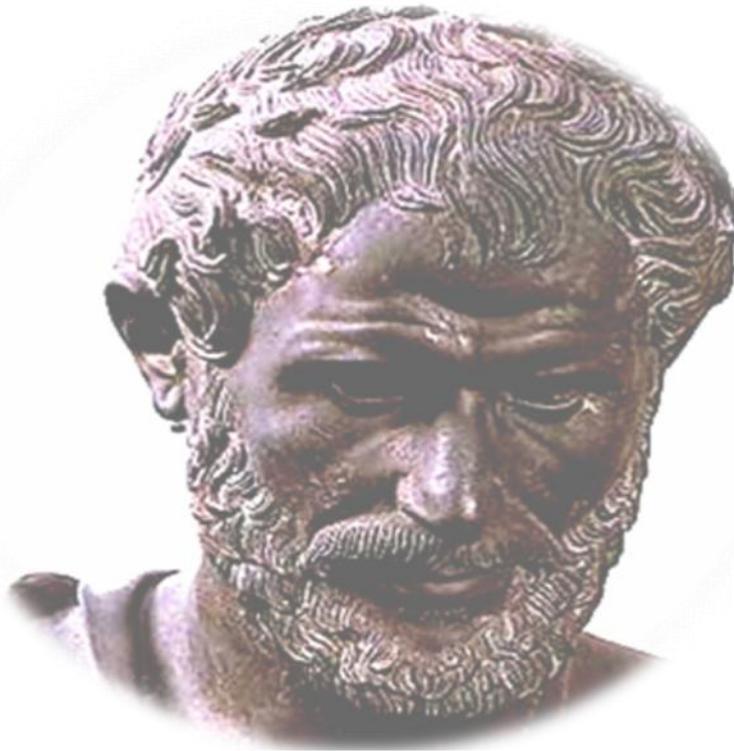
Yaser Morera Rivera

Firma del Tutor

Ing. Mario Aballe Rodríguez

Firma del Tutor

Dra.C. Rosa I Urquiza Salgado



La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.

Aristóteles

Gracias a mis padres, por darme la vida, por tanto amor y apoyo incondicional, por ser la mano amiga, el refugio perfecto de mis alegrías y tristezas.

Gracias a mi hermana por ser tan importante y especial para mi.

Gracias a mi novia por su amor y apoyo y por soportarme siempre.

Gracias a mi familia, porque en los momentos difíciles que enfrenté, estuvieron siempre ahí.

Gracias a los tutores Rosa Urquiza y Mario Aballe, por el apoyo y las horas dedicadas.

Gracias a esos compañeros irremplazables de la casa de software por su compañía ayuda y cooperación.

Quiero darles las gracias a todos mis profesores porque sin ellos no me sería posible estar aquí.

A mis compañeros de grupo que son excepcionales.

A mis compañeros de cuarto, por hacerme más llevadera esta difícil etapa de mi carrera.

Muchas gracias a todas esas personas de una forma u otra pusieron su granito en la realización de este, mi sueño. Muchas gracias.

Resumen

La empresa de Ingeniería y Proyecto del Níquel (CEPRONIQUEL) de Moa tiene entre sus procesos fundamentales la gestión de proyectos, entre los cuales se encuentra como subproceso el control de los proyectos, que diariamente maneja múltiples informaciones referentes a los planes de diseño generales de cada proyecto, que circulan por las áreas de manera insegura e informal, que se procesan de diferentes maneras. Al no tener implantado roles de seguridad sobre la información ni contar con una manera eficiente de obtención de reportes, no se logra un tiempo de respuesta breve en todo el proceso, lo que es de vital importancia para lograr eficiencia y calidad en sus servicios.

Esta investigación pretende elevar el nivel de informatización del proceso de control de proyectos, agilizar los procesos productivos de manera satisfactoria, mediante el diseño de un sistema informático (Módulo Web), el cual se caracteriza por la interconexión e integración al flujo de información de las áreas en la red, con alto grado de confiabilidad, uniformidad y consistencia.

Abstract

The Engineering and Project Nickel Company (CEPRONIQUEL) located in Moa, has among its fundamental processes, the project management and as by-product, the control of projects that daily carry out multiple informations related to the general design plans of each project that circulate for the areas in an insecure and informal way, being processed in different ways. They have neither implanted a security system over the information, nor an efficient way of obtaining reports which imply that achieving a brief time of answering, in the whole process, is the most importance for getting efficiency and quality in their services. This investigation seeks to rise the computerization process level of projects control, to speed up the productive processes satisfactorily, by means of designing a computer system (Module Web), which is characterized by the interconnection and integration to the information flow of the areas in the net, in high confidence, uniformity and consistency.

Índice

Introducción.....	1
Capítulo 1. Fundamentación Teórica	6
1.1 Caracterización de la Empresa de Ingeniería y Proyecto del Níquel.....	6
1.1.1 ¿Qué es un proyecto?	7
1.1.2 Descripción general de la administración de proyectos	7
1.1.3 El control en la administración del proyecto.....	9
1.1.5 Funciones de los Jefes de Proyecto	10
1.1.6 Ciclo de vida del proyecto	11
1.2 Procesos del negocio	12
1.2.1 Análisis de la ejecución de los procesos del negocio	13
1.2.2 Infraestructura tecnológica.....	14
1.3 Contexto teórico del desarrollo del módulo informático propuesto	14
1.3.1 ¿Qué es una aplicación Web?	14
1.3.2 ¿Qué es Internet?	16
1.3.3 ¿Qué es una Intranet?	18
1.4 Sistemas informáticos existentes vinculados al campo de acción.....	19
1.4.1 Arquitectura cliente – servidor	20
1.4.2 Características del modelo cliente/servidor	21
1.5 Tecnologías para el desarrollo de aplicaciones Web	23
1.5.1 Entorno de Desarrollo Integrado Eclipse	26
1.6 CMS Liferay.....	26
1.6.1 Frameworks	27
1.6.2 Patrón Modelo Vista Controlador	29
1.6.3 Portlet	30
1.7 Sistema Gestor de Base de Datos.....	31
1.8 Metodologías para el desarrollo del Software	32
1.8.1 Programación Extrema (Extreme Programming, XP)	32
1.8.2 ICONIX	33
1.8.3 Proceso Unificado de Rational (RUP).....	35
Conclusiones del Capítulo.....	36
Capítulo 2. Descripción de la solución propuesta.....	37

2.2 Modelo del Dominio.....	37
2.2.1 Definición de los principales objetos del modelo del dominio	38
2.2.2 Concepción del Modelo del Dominio.....	39
2.3 Modelación de los casos de uso	39
2.3.1 Actores del sistema.....	41
2.3.2 Descripción de los casos de uso.....	42
2.4 Estudio de Factibilidad	43
2.4.1 Factibilidad técnica	44
2.4.2 Factibilidad operacional	44
2.4.3 Factibilidad económica	44
2.5 Análisis de Robustez.....	46
2.6 Arquitectura Técnica	48
2.6.1 Requerimientos no Funcionales.....	49
2.6.2 Modelo de Despliegue	51
2.6.3 Arquitectura del sistema	51
2.7 Diagramas de Secuencia	53
2.8 Diagrama de clases persistentes	55
2.9 Modelo de Implementación	55
2.9.1 Descripción del sistema.....	56
2.10 Estándares de Código.....	58
2.11 Valoración de Sostenibilidad	58
2.12 Valoración de los resultados en la encuesta a los expertos y posibles usuarios. (VALORAR)	61
Conclusiones generales	65
Recomendaciones.....	66
Glosario de términos	67
Bibliografía	69

Índice de Tablas

TABLA 2.1 GLOSARIO DE TÉRMINOS	38
TABLA 2.2 DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA	42
TABLA 2.4 DESCRIPCIÓN DEL CUS MOSTRAR PLAN DE DISEÑO GENERAL.....	43
TABLA 2.5 DESCRIPCIÓN DEL CUS REDISTRIBUIR TAREAS.....	43
TABLA 2.6 DESCRIPCIÓN DEL CUS GENERAR REPORTE.....	43
TABLA 2.7 RESULTADOS COCOMO II.....	46

Índice de Figuras

FIGURA 1.1 ESTRUCTURA ORGANIZATIVA DE CEPRONIQUEL	7
FIGURA 1.2 TRIÁNGULO DEL PROYECTO.....	9
FIGURA 1.3 FLUJO DE INFORMACIÓN	10
FIGURA 1.4 FASES DE LA VIDA DEL PROYECTO.....	11
FIGURA 1.5 ARQUITECTURA CLIENTE - SERVIDOR INTERACTIVA PARA LA WEB.....	21
FIGURA 1.7 ESQUEMA DE LA METODOLOGÍA ICONIX.....	35
FIGURA 2.1 DIAGRAMA DEL DOMINIO	39
FIGURA 2.2 DIAGRAMA DE PAQUETES DE CASOS DE USOS DEL SISTEMA.....	40
FIGURA 2.2 DIAGRAMA DE CASOS DE USO DEL PAQUETE SEGURIDAD	40
FIGURA 2.3 DIAGRAMA DE CASOS DE USO DEL PAQUETE PROYECTO	41
FIGURA 2.4 DIAGRAMA DE ROBUSTEZ CUS MOSTRAR PLAN DE DISEÑO GENERAL	47
FIGURA 2.5 DIAGRAMA DE ROBUSTEZ CUS REDISTRIBUIR TAREAS.....	47
FIGURA 2.6 DIAGRAMA DE ROBUSTEZ CU GENERAR REPORTE	48
FIGURA 2.7 ARQUITECTURA DEL SISTEMA	52
FIGURA.2.8 DIAGRAMA DE SECUENCIA DEL CU MOSTRAR PLAN DE DISEÑO GENERAL	53
FIGURA 2.10 MODELO DE CLASES PERSISTENTES.....	55
FIGURA 2.12 INTERFAZ DE AUTENTICARSE	56
FIGURA 2.13 CALENDARIO DE CARGA DE TRABAJO (VISTA SEMANAL)	56
FIGURA 2.14 CALENDARIO DE CARGA DE TRABAJO (VISTA MENSUAL).	57
FIGURA 2.15 INTERFAZ CONTROLAR TAREAS POR PROYECTO.....	57

Introducción

La necesidad de organizar el desarrollo de todo tipo de actividades humanas es tan antigua como el hombre mismo: le ayudó a sobrevivir, evolucionar, expandirse por el planeta y asociarse en comunidades y sociedades. Organizar el conocimiento dio origen a la especialización de las ramas y disciplinas científicas, como las ciencias sociales, naturales y técnicas.

La política y la sociología son fruto de la organización de las interacciones sociales entre los seres humanos. Históricamente, la solución de muchas situaciones y problemas organizativos a los que se ha enfrentado el hombre ha sido gracias a su capacidad para planificar y controlar sus actividades, identificando procesos y subprocesos.

El crecimiento constante de la humanidad y de Internet ha posibilitado nuevas formas de comunicación entre las personas y las compañías, donde la información constituye una fortuna de gran valor, la misma representa para muchos la materia prima fundamental para el logro del producto final con los requerimientos de estos tiempos. Las experiencias que se van almacenando con el decursar del tiempo, los informes de los resultados obtenidos, entre otros, son utilizados por los directivos que se sirven de Sistemas Informáticos (SI) para apoyar la toma de decisiones.

Las distintas compañías prestadoras de servicios o manufactureras, por su parte, los utilizan para hacer pedidos de suministros y distribuir bienes con mayor rapidez. Debido a los diferentes puntos de vista con que se desarrollan los SI, se produce la incomunicación entre los mismos, por la falta de entendimiento mutuo. Por otro lado, existe la necesidad de reutilización de este conocimiento para que resulten eficientes.

La posibilidad que brindan los sitios Web y la gestión de información a través de ellos hace que la tendencia actual sea la de implementar en las instituciones empresariales sistemas para gestionar su información, simplificar sus procesos de negocios y elevar los objetivos económicos y sociales. Es el caso de la empresa de ingeniería y proyecto del níquel CEPRONIQUEL, líder en los servicios de Ingeniería, Consultoría y Dirección Integrada de Proyectos del Grupo Empresarial CubaNíquel y de la comunidad, ubicada en la zona nororiental de Cuba.

La Facultad de Informática y Matemática (FACINF) de la Universidad de Holguín desde al año 2010 viene consolidando fuertes lazos de colaboración con la mencionada empresa, con el objetivo de confeccionar un portal corporativo que integre los principales procesos modulares de dicha empresa, donde el núcleo fundamental es la Gestión de Proyectos, que tributa directamente a la razón de ser de la empresa: la producción de proyectos de ingeniería y diseño.

Durante el desarrollo de la presente investigación se detectaron distintas deficiencias. Desde que se inicia el flujo de procesos, el jefe de proyecto confecciona el plan de diseño general de cada proyecto en Microsoft Project[®], donde distribuye las tareas cronológicamente de acuerdo a las especialidades que van a estar implicadas, se asigna al responsable de cada una distintas informaciones como la tarea predecesora, la fecha de inicio y la duración de esta, entre otras. De acuerdo al tipo de contrato, ya sea en divisa o en moneda nacional, se determina el costo de la tarea por las tarifas establecidas.

No existen roles de seguridad implantados sobre la información, lo que provoca poca confiabilidad y pérdida de esta. Las vías actuales de circulación de datos, como el teléfono, correo electrónico, chats y memorias flash, entre otras, ocasionan una notable lentitud, pues la información no fluye con la rapidez y calidad requerida. A la hora de realizar el control y determinar dónde están las principales demoras en el proceso de producción se ocasionan innecesarios gastos de tiempo, el factor fundamentales de la vida de un proyecto.

A partir de la situación antes expuesta y del volumen de información existente entre las diversas especialidades implicadas en cada proyecto, se identificó el siguiente **problema científico**: ¿Cómo favorecer el control de los proyectos en la empresa CEPRONIQUEL de Moa?

El problema anteriormente definido se enmarca en el **objeto de estudio**: El proceso de control de los proyectos en la empresa CEPRONIQUEL de Moa.

Para la solución del problema planteado se persigue el siguiente **objetivo**: Desarrollar una aplicación Web para el control de los proyectos en la empresa CEPRONIQUEL de Moa.

El objetivo de la investigación delimita el **campo de acción**: Informatización del proceso de control de los proyectos en la empresa CEPRONIQUEL de Moa.

Para guiar la investigación, se trazaron las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos en cuanto al control de proyectos de la empresa CEPRONIQUEL de Moa?
2. ¿Cuál es el estado actual de la informatización del control de los proyectos de la empresa CEPRONIQUEL de Moa?
3. ¿Será factible y sostenible el sistema informático que se propone?
4. ¿Cómo diseñar e implementar un sistema informático para el control de los proyectos de la empresa CEPRONIQUEL de Moa?
5. ¿Cómo validar el sistema informático propuesto?

Para darles respuesta a las preguntas científicas y cumplir el objetivo trazado, se planificaron las siguientes **tareas**:

1. Elaborar los fundamentos teóricos para comprender el proceso de control de proyectos de la empresa CEPRONIQUEL de Moa.
2. Diagnosticar el estado actual de desarrollo obtenido en la informatización del control de los proyectos de la empresa CEPRONIQUEL.
3. Analizar y valorar la sostenibilidad del sistema informático.
4. Estudiar las tecnologías y tendencias actuales para la elaboración del módulo informático.
5. Diseñar e implementar un sistema informático para mejorar el control de proyectos en la empresa CEPRONIQUEL de Moa.
6. Validar parcialmente la solución propuesta mediante el criterio de expertos.

Para dar solución a las tareas planteadas se usó una combinación de métodos teóricos y empíricos.

Métodos Teóricos

Análisis y síntesis: Se utilizó con el fin de analizar la información manejada durante el proceso de gestión de proyectos y en particular el control de los proyectos en la empresa CEPRONIQUEL de Moa. Se elaboraron los fundamentos teóricos para descomponer las necesidades en requerimientos del sistema, así

como el propio sistema en subsistemas, para facilitar su comprensión; a través de este método se realizó la valoración de sostenibilidad del sistema.

Histórico y lógico: La aplicación de este método permitió analizar y estudiar las anteriores formas de controlar los proyectos para comprender mejor el funcionamiento y la lógica del negocio y expresar, en forma teórica, la esencia del proceso de gestión de proyectos.

Enfoque sistémico: Fue utilizado para identificar y descomponer el sistema en pequeños subsistemas para facilitar y organizar el trabajo y la lógica del negocio identificada.

Modelación: Permitted representar de manera simplificada el flujo de trabajo relacionado con el control de proyectos de la empresa CEPRONIQUEL de Moa, posibilitando una mejor comprensión de los procesos en las diferentes áreas de trabajo.

Métodos Empíricos

Entrevista: Posibilitó la obtención de información sobre cómo era el flujo de información entre los trabajadores y los Jefes de Proyecto a través de las diferentes fases del proceso de gestión de cada proyecto, cómo se procesaba y qué se necesitaba. Fueron entrevistados los Jefes de Proyectos, Controladores de la Calidad y Especialistas Principales, lo que facilitó la comunicación entre los usuarios y los desarrolladores.

Revisión de documentos: Se utilizó para la recopilación de la información y permitió conocer los detalles encontrados por los controladores de la calidad acerca de la estructura, funcionamiento y organización de la gestión de los proyectos, para así definir una idea única en flujo, así como también las normas por las cuales se rige la empresa. Se revisaron los siguientes documentos: Manual de identidad corporativa y Normas de Calidad ISO 9001.

Observación: Se realizó en diferentes períodos para permitir obtener un conocimiento superior sobre las labores rutinarias que se realizan en las diferentes fases de gestión de un proyecto. A partir de este conocimiento se estableció la problemática a investigar.

Encuestas: Se desarrollaron para elegir los expertos y obtener valoraciones conclusivas de éstos sobre el sistema. Para el tratamiento de las encuestas se usarán métodos para realizar el análisis de encuestas, como el método empírico Delphi.

El presente documento está estructurado en introducción, dos capítulos, conclusiones, recomendaciones, bibliografía y anexos.

En el Capítulo 1 (Fundamentación teórica) se realizó un estudio bibliográfico actualizado con el propósito de dar cumplimiento al objetivo trazado, se estudiaron las principales tendencias y tecnologías actuales, la metodología de Ingeniería de Software empleada, así como se ofrece una descripción del objeto de estudio de la investigación.

En el Capítulo 2 (Descripción de la solución propuesta) se expone el funcionamiento interno del proceso de control de proyectos, a través de la metodología empleada, y la determinación de las reglas del negocio, así como la captura de requerimientos y el modelo de casos de uso del sistema.

Capítulo 1. Fundamentación Teórica

En este capítulo se fundamenta la información que sustenta la presente investigación, se aplican definiciones necesarias en torno al objeto de estudio y la solución propuesta. Se exponen conceptos como administración de proyectos y flujo de información, así como las características del proceso medular de la empresa de Ingeniería y Proyecto del Níquel CEPRONIQUEL, donde el control de proyectos influye de manera directa en la producción. Se describen además las tendencias y tecnologías actuales sobre las que se desarrolla la propuesta, haciendo hincapié en sus características y ventajas que se proponen para la confección del presente trabajo. Finalmente, se fundamenta la metodología de desarrollo empleada.

1.1 Caracterización de la Empresa de Ingeniería y Proyecto del Níquel

CEPRONIQUEL es una organización dedicada al suministro de servicios de Ingeniería, Diseño y Consultoría, y Gestión de Proyectos capaces de cumplir el nivel esperado o deseado de satisfacción de sus clientes y expectativas de otras partes interesadas. Fue inaugurada en el año 1985, cuenta con un personal técnico y de dirección joven con adecuada preparación en la actividad de Ingeniería, con edad promedio de 44 años.

Entre sus principales prestaciones se encuentran: Servicios Técnicos de Ingeniería Conceptual, Básica y de Detalles; Dirección de la Construcción y Dirección Integrada de Proyectos. Su cliente primordial es la Industria Cubana del Níquel (CubaNíquel), brinda servicios a todas las empresas que pertenecen al grupo empresarial, y en un segundo plano a la comunidad.

A pesar de ser una institución joven, ha estado involucrada en proyectos de gran envergadura, como son:

- Modernización de la central termoeléctrica “Comandante Ernesto Che Guevara”.
- Proyecto de Urbanización Ciudad Moa.
- Transportador de mineral en Pinares de Mayarí.
- Montaje de calderas de 90 toneladas en la fábrica “René Ramos Latour”.

CEPRONIQUEL ha tenido asociaciones económicas internaciones con las firmas de ingeniería canadienses *Kvaerner Metals* y *Knight Piesold*, ambas reconocidas a nivel mundial, y ha realizado proyectos conjuntos con la participación de otras firmas extranjeras, tales como la firma alemana *FAM*, la firma de Ingeniería *Hatch Chile* entre otras. Ha logrado convenios de colaboración con firmas extranjeras tales como *Miessa* para temas de automática y eléctrica de tecnología *Siemens*. [1]

En la siguiente figura se muestra un esquema correspondiente a la estructura organizativa de la empresa CEPRONIQUEL de Moa.

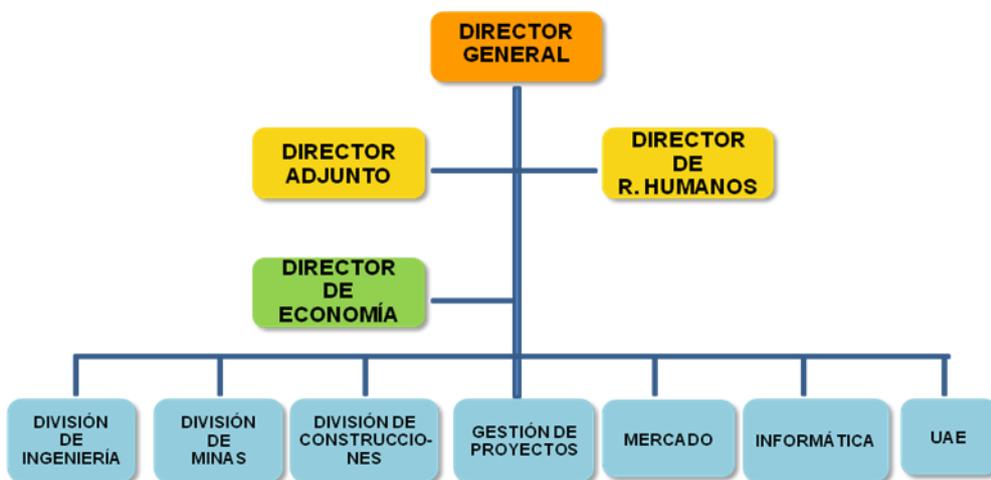


Figura 0.1 Estructura organizativa de CEPRONIQUEL

1.1.1 ¿Qué es un proyecto?

Según Maynard [2] "(un proyecto es) ... un proceso de trabajo grupal que se extiende por una determinada cantidad de tiempo y que debe llevar como resultado una facilidad tecnológica que puede ir desde una construcción civil o industrial, hasta un complejo productivo o de servicios pasando por una solución tecnológica de cualquier índole".

1.1.2 Descripción general de la administración de proyectos

La administración de proyectos comenzó a afianzarse a principios de los años sesenta del siglo pasado, las empresas y otras organizaciones estudiaron las ventajas de organizar el trabajo en forma de proyectos. Esta perspectiva de organización centrada en proyectos evolucionó aún más cuando las organizaciones empezaron a entender la necesidad fundamental de que sus empleados se

comuniquen y colaboren entre sí al tiempo que integran su trabajo en diferentes departamentos, profesiones y, en algunos casos, industrias completas.

La administración por proyectos es una modalidad de dirección que utiliza técnicas y herramientas estandarizadas, sistemas de información estructurada, organización con conocimiento y experiencia, para satisfacer las necesidades y expectativas que se tienen de un proyecto, logrando objetivos predefinidos tales como: Plazo, Costo y Calidad, entre otros.

Según Billows [3] la Administración por Proyectos (o Project Management) se refiere al *"..... proceso de organización de los recursos puestos a disposición del proyecto con el fin de obtener los objetivos deseados"*.

El Project Management Book of Knowledge (PMI) [4], lo define como *"el arte de dirigir y coordinar los recursos humanos y materiales, mediante el uso y técnicas de dirección para conseguir los objetivos prefijados de alcance, costos, plazo de entrega, calidad y satisfacción de los participantes o partes interesadas en el proyecto"*.

La adopción de esta forma de dirección respecto a otros enfoques puede constituir una vía exitosa hacia la integración de los diferentes elementos del sistema empresarial y contribuir al logro de los niveles de competitividad necesarios, entre otras razones por permitir una mayor flexibilidad y rápida respuesta ante la situación actual de incertidumbre y cambio de los mercados actuales.

Los preceptos básicos de la administración de proyectos están representados por el triángulo del proyecto (triángulo del proyecto: interrelación del trío formado por tiempo, dinero y ámbito en un proyecto. Si ajusta cualquiera de estos elementos, los otros dos se ven afectados. Por ejemplo, si se ajusta el plan del proyecto para acortar la programación, puede que aumenten los costos y disminuya el ámbito), un símbolo que popularizó Harold Kerzner en su obra de referencia, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*.

La siguiente figura muestra las tres aristas fundamentales que forman el triángulo del proyecto:



Figura 0.2 Triángulo del proyecto

En pocas palabras, la administración de proyectos es la disciplina de la planeación, organización y administración de los recursos necesarios para alcanzar un objetivo específico.

1.1.3 El control en la administración del proyecto

Desde el punto de vista de considerar a una empresa como el organismo de un ser humano implica que, para que un negocio pueda sobrevivir y prosperar, todas sus piezas funcionales deben trabajar en conjunto hacia objetivos o proyectos específicos.

En las décadas posteriores a los años sesenta del siglo pasado, este enfoque hacia la administración de proyectos comenzó a afianzarse en sus formas modernas. Aunque varios modelos empresariales evolucionaron durante este período, todos comparten una estructura subyacente común: un administrador del proyecto administra el proyecto, reúne un equipo y garantiza la integración y comunicación horizontal del flujo de trabajo en los diferentes departamentos.

El control es una etapa primordial en la administración pues, aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, el ejecutivo no podrá verificar cuál es la situación real de la organización si no existe un mecanismo que se cerciore e informe si los hechos van de acuerdo con los objetivos.

El concepto de control es muy general y puede ser utilizado en el contexto organizacional para evaluar el desempeño general frente a un plan estratégico.

A fin de incentivar que cada uno establezca una definición propia del concepto, se revisaron algunos planteamientos de varios autores estudiosos del tema:

- Chiavenato (2001): “El control es una función administrativa: es la fase del proceso administrativo que mide y evalúa el desempeño y toma la acción correctiva cuando se necesita. De este modo, el control es un proceso esencialmente regulador” [5].
- Robert Mockler (1924): El control administrativo es un esfuerzo sistemático para fijar niveles de desempeño con objetivos de planeación, para diseñar los sistemas de retroalimentación de la información, para comparar el desempeño real con esos niveles predeterminados, para establecer si hay desviaciones y medir su importancia y para tomar las medidas tendientes a garantizar que todos los recursos de la empresa se utilicen en la forma más eficaz y eficiente posible en la obtención de los objetivos [6].

1.1.5 Funciones de los Jefes de Proyecto

Una de las funciones primordiales de los administradores de proyectos es administrar los procesos internos del mismo, donde realmente se efectúa el trabajo más importante. Por pequeño que sea el proyecto, se requieren habilidades de administración del mismo para sortear las diferentes situaciones que se presenten, y además garantizar el cumplimiento de los objetivos dentro de los tiempos estipulados. Estas habilidades van desde la definición del proyecto, hasta la administración de las medidas de avance del mismo.

La siguiente figura el rol que juega el Jefe de Proyecto en el flujo de información entre el cliente y la empresa CEPRONIQUEL de Moa:

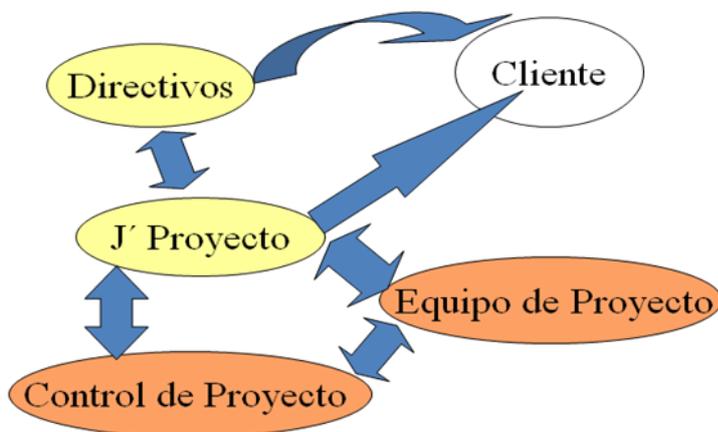


Figura 0.3 Flujo de información

1.1.6 Ciclo de vida del proyecto

El proyecto pasa por diferentes etapas, reconocidas generalmente como: Conceptualización, Diseño y Planificación, Control de la Ejecución y Cierre o Declinación (ver Figura 1.5) [2] .

La etapa de conceptualización comprende el desarrollo de la idea preliminar hasta la demostración de su viabilidad, se realiza un análisis de las posibles alternativas de solución del problema que se pretenda resolver. Generalmente, culmina con la aprobación de los resultados de estudios de viabilidad que comprenden análisis preliminares de comportamiento de mercado y demanda, suministros, condiciones del entorno, así como estimados de posibles resultados económicos y socio ambientales.

La etapa de diseño y planificación de un proyecto es la más larga e intensa en trabajo intelectual de equipo. En ella se analizan a detalle todas las posibles soluciones, se efectúan los diseños de procesos y productos, se planifica la ejecución de las diferentes fases de la etapa de desarrollo. A todo esto se le suele llamar Ingeniería de Detalle¹.

La puesta en marcha y control del proyecto es la etapa de trabajo más intensa en el colectivo, donde juegan un papel fundamental los jefes de proyectos en cada área, que tienen la función de determinar las fallas del proceso y corregirlas para eliminar errores y, consecuentemente, pérdidas de tiempo y de dinero.

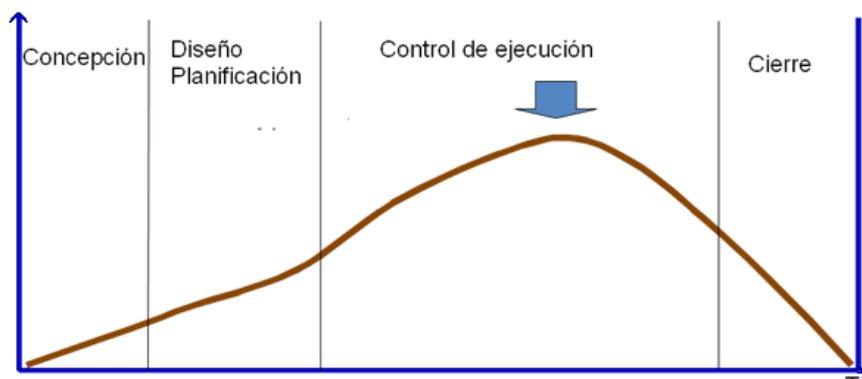


Figura 0.4 Fases de la vida del Proyecto

¹ Ingeniería de Detalle: Definición pormenorizada de los materiales, mano de obra, equipos necesarios, cálculo del presupuesto con una descripción detallada de las partidas etc., del proyecto, con una planificación de los hitos para la ejecución del mismo.

Control del Proyecto

Es la fase de la administración del proyecto más importante, que incluye una serie de mecanismos como son el uso del diagrama de flechas y la gráfica de tiempo para hacer reportes periódicos del progreso. Se deben representar las interdependencias y relaciones de precedencia entre las actividades del proyecto, utilizando distintos mecanismos para determinar el avance del proyecto y representar la terminación de unas actividades y el comienzo de otras.

1.2 Procesos del negocio

En visitas realizadas a la empresa se efectuaron búsquedas, a fin de encontrar el mayor volumen de información, referente al proceso de administración y control de la ejecución de tareas de cada plan de diseño general. El responsable de la organización y control de esta actividad es el Jefe de Proyecto, el cual tiene a su cargo la concepción y elaboración del Plan de Diseño General. La distribución de las tareas se realiza de acuerdo a la división que pertenezca, ya sea la de Ingeniería, Minería o la de Construcción.

Las distintas divisiones están formadas por grupos, cada uno está dirigido por el especialista principal y a su vez cada especialidad tiene un controlador de la calidad, aunque este puede atender varias especialidades.

Los jefes de proyecto son capaces de definir con un alto grado de seguridad las especialidades implicadas, además de dirigir y controlar las tareas correspondientes a los proyectos relacionados con sus divisiones; se conforma todo el engranaje en la reunión de alcance que, como su nombre lo indica, es para determinar las particularidades del proyecto y donde participan los jefes de divisiones, los especialistas principales y los controladores de la calidad.

Una vez terminada la reunión de alcance, los controladores de la calidad de cada una de las especialidades elaboran un conjunto de documentos. Estos son: un plan de diseño creado con la ayuda del software Microsoft Project®, un documento donde se plasme el alcance del proyecto y un listado de normas a usar, los cuales son revisados por los especialistas principales y luego por el jefe de división, quien envía al Jefe de Proyecto los documentos mencionados anteriormente por cada especialidad para conformar el plan de diseño general de cada proyecto.

Una vez realizada la reunión inicial de trabajo, los especialistas comienzan a trabajar, de acuerdo al Plan de Diseño General establecido. Cada semana, el Jefe de División asigna un porcentaje de cumplimiento a las tareas asignadas que tienen que hacer las especialidades. Las tareas realizadas son revisadas por los controladores de calidad, tanto el especialista principal como el Jefe de División y el Jefe de Proyecto controlan que el trabajo se realice con la calidad que identifica a la empresa según las normas de calidad internacional.

1.2.1 Análisis de la ejecución de los procesos del negocio

Actualmente, en CEPRONIQUEL el proceso de control de tareas correspondientes a los planes de diseño generales de cada proyecto no se realiza de la manera más eficiente. También existen inconvenientes en el flujo y manipulación de la información, la forma en que se realiza no explota al máximo las potencialidades de la infraestructura informática con que se cuenta en la entidad.

No existe la manera de saber en cada momento a los jefes de proyecto el estado de cumplimiento que encuentran las tareas correspondientes a los planes de diseño generales de cada proyecto, o simplemente los jefes de divisiones tener conocimiento del estado de las tareas implicadas cada una de las especialidades de su división sin necesidad de planificar una reunión de producción.

Todas las informaciones de relevante importancia en las distintas áreas referente al control de los proyectos fluyen entre otros a través del correo electrónico, memorias flash y archivos de Microsoft Project®, que circulan por la red sin un sistema Web **¡Error! No se encuentra el origen de la referencia.** que le sirva de soporte y le brinde confiabilidad e integridad a los datos.

"El objetivo esencial del seguimiento y control de los proyectos es incrementar los niveles de eficiencia y efectividad dentro de la organización. Incluye además la persistente búsqueda de nuevas formas de organizar y facilitar el trabajo para alcanzar propósitos con la menor cantidad de recursos y tiempo. Logrando la obtención de la información adecuada, para la persona adecuada, a su precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada" [7].

1.2.2 Infraestructura tecnológica.

La empresa CEPRONIQUEL cuenta con 184 estaciones de trabajo, conectadas a través de una red Lan y distribuidas todas en un edificio de 4 pisos de la siguiente manera:

- Cuarta planta: 30
- Tercera planta: 74
- Segunda planta: 57
- Primera planta: 23

Todas las estaciones están protegidas por sus Fuentes de Alimentación Ininterrumpible (UPS, por sus siglas en inglés). El nodo de la red está ubicado en la tercera planta. El diseño asistido por computadoras se realiza con 15 potentes servidores (10 PC dedicadas) de última generación, en general las estaciones de trabajo son en su mayoría Pentium IV - Core 2 Duo con sus Memorias de acceso aleatorio (RAM, por sus siglas en inglés) de 512 - 3 Gbyte, Pantallas de cristal líquido (LCD, por sus siglas en inglés) de 17"- 22". La administración de sistemas y conexiones son supervisados y mantenidos por el departamento de Informática con 4 especialistas. Su capacitado personal, su organización y su valiosa infraestructura técnica permiten que el proceso de informatización previsto por esta investigación pueda llevarse a cabo en corto plazo.

1.3 Contexto teórico del desarrollo del módulo informático propuesto

Para analizar las tendencias y tecnologías actuales disponibles para el desarrollo del módulo Web propuesto, se hace necesario enunciar conceptos y características fundamentales que sirven de fundamento teórico.

1.3.1 ¿Qué es una aplicación Web?

En la ingeniería de software se denomina aplicación Web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet o de Intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores Web (Lenguaje de Marcas Hipertextuales o HTML por sus siglas en inglés, JavaScript, Java, etc.), en la que se confía la ejecución al navegador. [8]

Una aplicación Web permite que siempre se esté accediendo a una información actualizada en cualquier parte de dicha red. Es un conjunto de páginas enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella, siendo una de las mejores herramientas para divulgar, gestionar y compartir la información. Trae consigo un aumento de la eficiencia en cuanto a la manipulación del volumen de la misma.

Ventajas de las aplicaciones Web

- Ahorra tiempo: se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- No hay problemas de compatibilidad: Basta tener un navegador mínimamente actualizado para poder utilizarlas.
- No ocupan espacio en el disco duro.
- Actualizaciones inmediatas: Como el software lo gestiona el propio desarrollador, cuando se conecta se usa siempre la última versión que haya lanzado.
- Consumo de recursos bajo: Dado que toda (o gran parte) de la aplicación no se encuentra en el ordenador local, muchas de las tareas que realiza el software no consumen recursos porque se realizan desde otro ordenador.
- Multiplataforma: Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: Es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil, un móvil...), porque se accede a través de una página Web (sólo es necesario disponer de acceso a Internet).
- La disponibilidad suele ser alta, porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Los virus no dañan los datos porque estos están guardados en el servidor de la aplicación.
- Colaboración: Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios.
- Los navegadores ofrecen cada vez más y mejores funcionalidades para crear aplicaciones enriquecidas de Internet.

En general, la gran ventaja de utilizar estas tecnologías es precisamente que las aplicaciones Web se desarrollan en un ambiente Cliente/Servidor. Esto trae como beneficio que la mayor parte del procesamiento se realice en el servidor, se dispone de una buena interfaz gráfica y una mejor interacción con el usuario, que garantiza la integridad de la información.

1.3.2 ¿Qué es Internet?

"Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET², entre tres universidades en California y una en Utah, Estados Unidos. Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW, o "la Web"), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Ésta fue un desarrollo posterior (1990) y utiliza Internet como medio de transmisión.

Existen, por tanto, muchos otros servicios y protocolos en que es Internet, aparte de la Web: el envío de correo electrónico (SMTP), la transmisión de archivos (FTP y P2P), las conversaciones en línea (IRC), la mensajería instantánea y presencia, la transmisión de contenido y comunicación multimedia -telefonía (VoIP), televisión (IPTV)-, los boletines electrónicos (NNTP), el acceso remoto a otros dispositivos (SSH y Telnet) o los juegos en línea. " [9]

Principales antecedentes en la historia de Internet

1969: la primera red interconectada nace el 21 de noviembre de 1969, cuando se crea el primer enlace entre las universidades de UCLA y Stanford por medio de la línea telefónica conmutada, y gracias a los trabajos y estudios anteriores de varios científicos y organizaciones desde 1959. El mito de que ARPANET, la primera red,

² Acrónimo en inglés de Advanced Research Projects Agency Network fue creada por el Departamento de Defensa de los Estados Unidos como medio de comunicación para sus diferentes organismos.

se construyó simplemente para sobrevivir a ataques nucleares sigue siendo muy popular.

1972: se realizó la primera demostración pública de ARPANET, una nueva red de comunicaciones financiada por la DARPA que funcionaba de forma distribuida sobre la red telefónica conmutada.

1973: se inició programa de investigación sobre posibles técnicas para interconectar redes (orientadas al tráfico de paquetes) de distintas clases. Para este fin, desarrollaron nuevos protocolos de comunicaciones que permitiesen este intercambio de información de forma "transparente" para las computadoras conectadas. De la filosofía del proyecto surgió el nombre de "Internet", que se aplicó al sistema de redes interconectadas mediante los protocolos TCP e IP.

1983: el 1 de enero, ARPANET cambió el protocolo NCP por TCP/IP. Ese mismo año, se creó el IAB con el fin de estandarizar el protocolo TCP/IP y de proporcionar recursos de investigación a Internet. Por otra parte, se centró la función de asignación de identificadores en la IANA que, más tarde, delegó parte de sus funciones en el Internet registry que, a su vez, proporciona servicios a los DNS.

1986: la agencia NSF comenzó el desarrollo de NSFNET, que se convirtió en la principal Red en árbol de Internet, complementada después con las redes NSINET y EUNET, todas ellas en Estados Unidos. Paralelamente, otras redes troncales en Europa, tanto públicas como comerciales, junto con las americanas formaban el esqueleto básico ("backbone") de que es Internet.

1989: Con la integración de los protocolos OSI en la arquitectura de Internet se inició la tendencia actual de permitir no sólo la interconexión de redes de estructuras dispares, sino también la de facilitar el uso de distintos protocolos de comunicaciones.

En el CERN de Ginebra, un grupo de físicos encabezado por Tim Berners-Lee creó el lenguaje HTML, basado en el SGML. En 1990, el mismo equipo construyó el primer cliente Web, llamado WorldWideWeb (WWW), y el primer servidor Web.

2006: El 3 de enero, Internet alcanzó los mil cien millones de usuarios. Se prevé que en diez años, la cantidad de navegantes de la Red aumentará a 2.000 millones.[9]

1.3.3 ¿Qué es una Intranet?

Intranet es la aplicación en una red de ordenadores de topología LAN³, estándares de Internet dentro de un ámbito corporativo para mejorar la productividad, reducir costes y mantener los sistemas de información existentes.

La implantación y explotación de una intranet trae múltiples ventajas:

- Publicar información fácilmente, se reducen los costos como por ejemplo los de imprimir, distribuir y archivar documentos.
- Acceder a información que de otro modo requeriría mucho tiempo para su publicación.
- Administrar la información dispersa por toda la organización y acceder a la base de datos de la empresa, con ahorro de costos e incremento de productividad.
- Compatibilizar distintas plataformas de sistemas utilizadas en una empresa y dar cabida a todo tipo de equipos, fabricantes, redes, tecnologías y medios físicos de transmisión.
- Establecer políticas de seguridad de la información, permite un control de alto nivel sobre el acceso a la misma.
- Utilizar una interfaz común (el navegador de Internet), con el consiguiente ahorro en capacitación y desarrollo de aplicaciones para usuarios finales.
- Permitir el acceso a información interna de la empresa desde cualquier lugar del mundo y en cualquier momento, incluso fuera de las horas habituales de trabajo, lo que le posibilita a un empleado, en su casa o de viaje, tener acceso a documentos necesarios con tan solo conectarse a Internet y usar una contraseña.
- Lograr que la información sea directamente administrada y mantenida por la gente que la genera.
- Hacer uso de las herramientas que habitualmente se utilizan en Internet, como el navegador, el correo electrónico, en este caso de uso interno en la organización, las listas de distribución, los boletines de noticias, la transferencia de ficheros, el acceso remoto, el chat y la videoconferencia, etc. [10]

³ LAN (Local Area Network). Topología de ordenadores conectados en una red de área local.

1.4 Sistemas informáticos existentes vinculados al campo de acción

La evolución de las Tecnologías de la Información y las Comunicaciones (TIC por sus siglas), en función de lograr adelantos en la sociedad actual, ha sido marcada fuertemente por el desarrollo acelerado de nuevas tecnologías de la información y las comunicaciones. En este aspecto, el desarrollo de sistemas informáticos se ha convertido en una herramienta crucial para las instituciones, ante la necesidad de lograr mayor eficiencia en los procesos y subprocesos para garantizar mejores resultados.

Mediante el estudio los antecedentes en sistemas informáticos que existen en este campo se encontraron una serie de SI pertenecientes a la compañía Oracle como son:

- El famoso *Phoenix Project Manager*, este software de programación basado en el Método del Camino Critico al igual que otros, ayuda a mantener el proyecto según lo planeado hasta su éxito.
- El *Project Plan* asegura ser el software de planificación que incluye todas las herramientas necesarias para llevar una planificación de proyecto de forma sencilla y en minutos.
- *Seavus Project Planner* es un software de gestión de proyectos para planificación y seguimiento de proyectos, gestión de sus recursos y control de costos.
- *Primavera P6 Enterprise Project Portfolio Management* surgido en diciembre del 2010 es la solución, según Oracle más potente, robusta y sencilla para la priorización, planificación y control de la ejecución de proyectos, incluye funcionalidades para satisfacer las necesidades de cada uno de los miembros del proyecto, según sus responsabilidades y tareas.

Herramientas como estas últimas presentan ciertas limitaciones a la hora de adquirirlas, ya que sus patentes y licencias pertenecen a compañías en su mayoría estadounidenses y privadas. Por ejemplo, Oracle confirmó el 31 de Diciembre de 2010 que abandonaba sus productos anteriormente mencionados, lo que obligó a muchas empresas que usaban estas potentes herramientas para el control de sus proyectos a migrar a *Primavera P6* o hacia otras alternativas.

En la actualidad, en la empresa CEPRONIQUEL el software utilizado es el Microsoft Project® que además de ser un software privativo de la compañía Microsoft no permite la explotación de la infraestructura tecnológica en toda su magnitud.

No obstante, a pesar de las ventajas que reporta por sus características en cuanto a la planificación, presenta ciertas limitaciones en sus funcionalidades a la hora de realizar el control de las tareas ya planificadas.

Software Libre

El software libre es una cuestión de libertad, no de precio. Para entender el concepto, debería pensar en libre como en libre expresión; permite ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las libertades esenciales [11].

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso [12].

1.4.1 Arquitectura cliente – servidor

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina

cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes (ver Figura 1.6). [13]

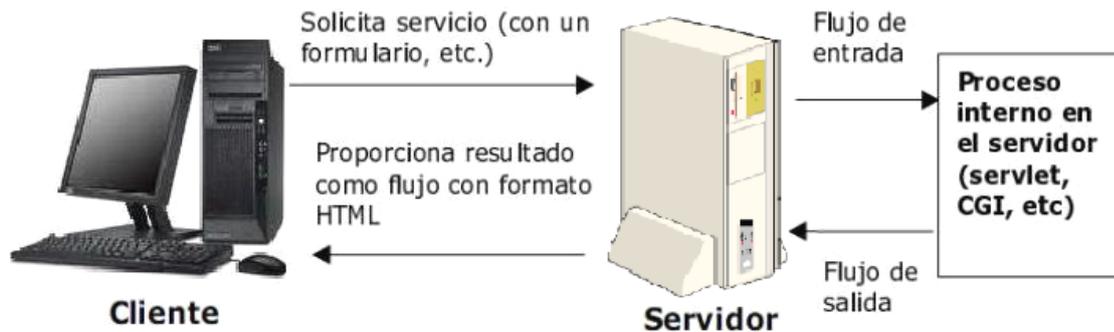


Figura 0.5 Arquitectura Cliente - Servidor interactiva para la Web

1.4.2 Características del modelo cliente/servidor

En el modelo cliente/servidor se puede encontrar las siguientes características:

- El cliente y el servidor pueden actuar como una sola entidad y también como entidades separadas, al realizar actividades o tareas independientes.
- Las funciones de cliente y servidor pueden estar en plataformas separadas, o en la misma plataforma.
- Un servidor da servicio a múltiples clientes en forma concurrente.
- Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los clientes o de los servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.
- La interrelación entre el hardware y el software está basada en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.
- Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, de esta manera el servicio se hace más efectivo para el usuario final.

- También es importante hacer notar que las funciones cliente/servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.
- Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.
- Además, se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.
- Designa un modelo de construcción de sistemas informáticos de carácter distribuido.
- Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización; al tiempo que puede acceder a los recursos de este host central y otros sistemas de la organización ponen a su servicio.

En conclusión, cliente/servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, donde tanto en arquitecturas propietarias como no propietarias funcionan al mismo tiempo. Por lo tanto, su implantación involucra diferentes tipos de estándares: APPC⁴, TCP/IP⁵, OSI⁶, DRDA⁷ entre otros corriendo sobre los sistemas operativos, en Ethernet⁸, FDDI⁹o algún medio coaxial, sólo por mencionar algunas de las posibilidades.[14]

⁴ APPC:(Advanced Program-to-Program Communications)Programa avanzado para comunicaciones entre programas.

⁵ TCP/IP (Transference Control Protocol / Internet Protocol) .Protocolo de control de transferencia / Protocolo de Internet.

⁶ OSI:(Open Systems Interconnection) Interconexión de sistemas abierto.

⁷ DRDA :(Distributed Relational Database Architecture) Arquitectura de base de datos distribuida.

⁸ ETHERNET:(Estándar de redes de computadoras de área local con acceso al medio)

⁹ FDDI:(Fiber Distributed Data Interface) Interfaz de datos distribuidos, de fibra.

1.5 Tecnologías para el desarrollo de aplicaciones Web

Las aplicaciones Web son las herramientas más efectivas para lograr un objetivo interoperacional, o de mucho flujo de información. Durante el estudio de diferentes tecnologías estas son las más populares, debido a lo práctico del navegador Web como cliente ligero, así como a la facilidad para actualizar y mantener las aplicaciones sin distribuir e instalar software a miles de usuarios potenciales.

Una aplicación Web permite que siempre se esté accediendo a una información actualizada en cualquier parte de dicha red, además de ser un conjunto de páginas enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella, siendo 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 del volumen de la misma. [15]

El software libre es ampliamente utilizado a partir de los beneficios que reporta, ya que es posible utilizar tecnologías avanzadas y eficientes para el desarrollo de sistemas informáticos de acuerdo a las necesidades actuales como los lenguajes de programación Web. En el caso de las aplicaciones Web, estas generan dinámicamente una serie de páginas en un formato estándar, como HTML o Lenguaje Extensible de Marcado de Hipertexto (XHTML, por sus siglas en inglés), soportadas por los navegadores Web comunes.

Se utilizan lenguajes interpretados en el lado y del cliente, para añadir elementos dinámicos a la interfaz de usuario. Durante la sesión, el navegador Web interpreta y muestra en pantalla las páginas, actuando como cliente para cualquier aplicación Web. Ejemplos: JavaScript, Visual Basic Script, Lenguaje Marcado de Hipertexto Dinámico (DHTML, por sus siglas en inglés). [16]

También se utilizan lenguajes de lado del servidor, que son aquellos lenguajes que se caracterizan por implementar la lógica del negocio dentro del servidor, además de encargarse de la tarea del acceso a las bases de datos y el tratamiento de la información. Ejemplos: Páginas Activas del Servidor (ASP, por sus siglas en inglés), Procesador de Hipertexto (PHP, por sus siglas en inglés) y Páginas de Servidor de Java (JSP, por sus siglas en inglés).

A continuación se relacionan algunas características de lenguajes utilizados para la creación de aplicaciones Web, se hace énfasis en el lenguaje de programación

Java y ZK framework, utilizado en desarrollo Web bajo plataforma Java, herramientas por las cuales se optó para el desarrollo de la aplicación. [17]

Ajax (Asynchronous JavaScript and XML)

En las últimas décadas las aplicaciones Web han evolucionado de páginas HTML estáticas a dinámicas, de Applets a Flash, y finalmente a tecnologías Ajax (Asynchronous JavaScript and XML, por sus siglas en inglés). Ilustrado por Google Maps y Suggest, Ajax se evidencia en las aplicaciones Web al ofrecer el mismo nivel de interactividad y respuesta de las aplicaciones de escritorio. A diferencia de los Applets o Flash, Ajax está basado en el navegador y JavaScript estándar sin requerir plugins propietarios.

Es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones siempre se ejecutan en el navegador del cliente, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano y a la vez la realización de cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en la aplicación. [18]

Lenguajes de programación para la Web

Son múltiples los lenguajes en el campo de la programación Web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas.

Los lenguajes se dividen en dos grupos fundamentales, los del lado del servidor y los del lado del cliente, entre los que se encuentran en la primera categoría están el ASP (Active Server Pages), PHP (Personal Home Page) y Java que contiene el Api JSP (Java Server Pages), este último seleccionado para el desarrollo de la aplicación. En el segundo grupo se encuentran JavaScript y Visual Basic Script, que se usan principalmente para aportar características a las aplicaciones.

Lenguaje de programación Java

En un principio, Java, fue pensado como un lenguaje para programar equipos electrodomésticos y artefactos electrónicos. Su popularidad fue aumentando con el paso del tiempo llegándose a convertir en un lenguaje de desarrollo de propósito general. Su sintaxis es muy similar al lenguaje C++, y posee características que lo hace un lenguaje de programación poderoso, sencillo y confiable, capaz de

ejecutarse en todo tipo de procesadores. Las características principales que ofrece Java respecto a otros lenguaje de programación, son: simplicidad, orientación a objetos, distribuido y confiabilidad.

Java es uno de los lenguajes de programación más difundidos en el mundo, y se usa con una gran variedad de fines, incluyendo el desarrollo de grandes aplicaciones empresariales.

Las características fundamentales que conllevaron a seleccionar como lenguaje de programación a Java se muestran 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.

Simple: Elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos. [19]

Robusto: Java verifica su código al mismo tiempo que lo escribe, y una vez más antes de ejecutarse, de manera que se consigue un alto margen de codificación sin errores.

Facilidad de obtención de IDEs: No sólo el código fuente de Java es de libre distribución, sino que existen numerosos entornos de programación para él totalmente gratuitos. Todos ofrecen garantías y características más que suficientes para desarrollar proyectos informáticos de elevada complejidad.

Java 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, aunque la gran cantidad y evolución de los frameworks existentes, ha traído como consecuencia no saber escoger el más adecuado para el desarrollo de una aplicación. [20]

1.5.1 Entorno de Desarrollo Integrado Eclipse

Eclipse es un entorno de desarrollo integrado, de código abierto y multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-Liviano" basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios Web, programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. [21]

Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de Software, Aplicaciones Web. El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en 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.

Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente, al permitirle a Eclipse extenderse al usar otros lenguajes de programación como C/C++ y Python, además de trabajar con procesadores de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos.

1.6 CMS Liferay

Liferay es el líder mundial en la creación de plataformas para portales corporativos de la comunidad de código abierto, provee la creación de entornos colaborativos. Dentro del portal Liferay una interfaz está compuesta por un número de portlets, ficheros war (Web Archive, por sus siglas en inglés), que pueden ser creados a través de IDEs como el NetBeans, Eclipse, JBuilder, los cuales están desarrollados y acoplados independientemente del portal por tanto son considerados como una

arquitectura orientada a servicios (SOA por sus siglas en ingles). Posee soporte para bibliotecas documentales, galería de imágenes, servicio de correo, blogs, chats, wikis, canales RSS, agendas compartidas y foros de discusión, gestionados por un completo sistema de roles y permisos de manera que cada usuario pueda gestionar sus propias páginas, con los portlets que desee.

Características del Liferay:

Modelo en hosting: Liferay ha sido diseñado desde su concepción para ser usado en formato proveedor de servicios de aplicación (ASP, por sus siglas en ingles) o hosting. Esto permite que se puedan alojar múltiples instancias del portal (simplemente diferenciadas por URL's distintas) sobre un único servidor de aplicaciones y una base de datos.

Independencia respecto de Bases de Datos. Liferay emplea Hibernate como herramienta ORM para la capa de persistencia, lo que facilita que soporte cualquier base de datos como DB2, Firebird, Hypersonic, InterBase, JDataStore, MySQL, Oracle, PostgreSQL, SAP o SQL Server.

Tomando como base las características antes expuestas, se determinó usar el CMS Liferay por las características particulares del entorno empresarial donde se desarrolla la investigación y la integración con las diversas tecnologías expuestas en el capítulo.

1.6.1 Frameworks

ZK es un framework para la construcción de aplicaciones Web basado en Ajax, completamente desarrollado en Java. ZK permite la construcción de interfaces ricas de usuario sin necesidad de usar ni conocer JavaScript. A diferencia de la mayoría de otros frameworks, Ajax, en lo que concierne a ZK, es una tecnología detrás de la escena, la sincronización del contenido de componentes y el oleoducto de eventos son hechos automáticamente por el motor de ZK. Además, incluye un motor manejado por eventos basado en Ajax, un conjunto rico de componentes XUL y XHTML y un lenguaje de marcado llamado lenguaje de marcado para el usuario de ZK (ZUML por sus siglas en inglés).

Este framework puede representar una aplicación rica y especial en componentes XUL y XHTML, y permite la manipulación sobre eventos lanzados por la actividad

del usuario, similar a lo que ocurre en las aplicaciones de escritorio. Implementa las funcionalidades del lado del servidor con la tecnología Java y las del cliente haciendo uso del lenguaje JavaScript.

Una de las principales ventajas que ha suministrado este framework es que soporta un lenguaje de marcación para la definición de una potente interfaz de usuario llamada ZUML, ya que está diseñado para que desarrolladores no expertos diseñen interfaces de usuario de forma eficiente, además de la programación basada en componentes, de manera similar a Swing¹⁰, que trae consigo una buena comunicación entre el usuario y el sistema. ZK es una de las plataformas que permite hacer más fácil la programación Web. Su fuerte paquete incluye interoperabilidad con navegadores, un paradigma para gestionar eventos y funcionalidades. [22]

Framework Spring

Spring es un contenedor ligero orientado a aspectos y con inversión de control. El núcleo del framework Spring está basado en el principio de Inversión de Control (IoC. por sus siglas en inglés), que es una técnica que externaliza la creación y administración de dependencias de componentes. Tiene soporte para la programación orientada a objetos, lo cual permite un desarrollo cohesivo separando la lógica del negocio de la aplicación de los servicios del sistema.

Una de las razones por la cual Spring provee los elementos necesarios para su implantación en el desarrollo de esta aplicación, es por el hecho de que puede ser utilizado para desarrollar cualquier tipo de aplicación en Java, así como no se limita de las aplicaciones Web, además de que tiene los mecanismos necesarios para el acceso a datos, teniendo soporte para JDBC, Hibernate, iBATIS, Java Data Objects (JDO, por sus siglas en inglés), proporciona un marco para la gestión de la seguridad del sistema informático, y administra el ciclo de vida y configuración de los objetos de la aplicación. [23]

Mapeo Objeto Relacional. Framework Hibernate

El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-

¹⁰ Swing es una biblioteca gráfica para Java que forma parte de las Java Foundation Classes (JFC por sus siglas en inglés).

Relational Mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. Hibernate es una herramienta de mapeo relacional de objeto para entornos Java. El termino 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 SQL¹¹.

Este framework de persistencia funciona como un capa intermedia entre la aplicación y la base de datos. A través de este se realizan consultas, se ejecuta código en lenguaje SQL y se manipulan los objetos según las necesidades de los usuarios.

Hibernate genera todo el SQL necesario, lo que trae como principal ventaja para el sistema que la velocidad de desarrollo aumenta al utilizar un framework de este tipo, pues los desarrolladores sólo deben preocuparse por escribir las clases que se mapearán, mientras que el framework se preocupa por hacer todas las operaciones de inserción, modificación, eliminación y consulta.

Algunas características de Hibernate:

- Es un ORM no intrusivo, es decir, las clases persistentes no necesitan implementar ninguna interfaz.
- Está escrito en Java, pero existe un plugin para ser utilizado en la plataforma .NET.
- Cada clase persistente necesita un fichero XML de mapeo. A partir de él, obtiene toda la información necesaria para realizar las operaciones CRUD.
- Hibernate es un producto open source, pero en 2003 se unió a jboss.org, que ofrece soporte comercial.

1.6.2 Patrón Modelo Vista Controlador

El uso de framework para desarrollar aplicaciones provee de buenas prácticas de diseño e implementación, que a su vez representan ventajas a la hora de elaborar

¹¹ Acrónimo en inglés de Structured Query Language, en español Lenguaje de Consulta Estructurado.

un software. El patrón Modelo-Vista-Controlador (MVC) es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados:

1. Modelo que representa los datos de la aplicación y sus reglas de negocio.
2. Conjunto de vistas que representa los formularios de entrada y salida de información.
3. Controladores que procesan las peticiones de los usuarios y controla el flujo de ejecución del sistema.

Al utilizar el patrón MVC existe una clara separación entre los componentes del programa; lo cual permite implementarlos por separado. La conexión entre el modelo y sus vistas es dinámica, se produce en tiempo de ejecución, no en tiempo de compilación. La principal ventaja de esta separación reside en la facilidad para realizar cambios en la aplicación puesto que cuando se hace un cambio de Bases de Datos, programación o interfaz de usuario, solo se afecta uno de los componentes. Además, se puede modificar uno de los componentes sin tener que conocer cómo funcionan los otros.

1.6.3 Portlet

Los portlets son componentes modulares de interfaz de usuario gestionadas y visualizadas en un portal web. Los portlets producen fragmentos de código de marcado que se agregan en una página de un portal. Un Fragmento no es más que un pedazo de código de lenguaje de etiquetas como por ejemplo HTML, XHTML, WML, etc. Estos fragmentos no pueden utilizar ciertas etiquetas de su respectivo lenguaje de etiquetas ya que pueden invalidar el fragmento entero, como por ejemplo en lenguaje HTML no se pueden utilizar las etiquetas `<html>`, `<head>`, etc. Siguiendo la metáfora de escritorio, una página de un portal se visualiza como una colección de ventanas de portlet que no se solapan, donde cada una de estas muestra un portlet. Por lo tanto un portlet (o colección de portlets) se asemeja a una aplicación web que está hospedada en un portal. Como por ejemplo, un portlet puede ser, un visor de correo electrónico, un foro, noticias, etc.

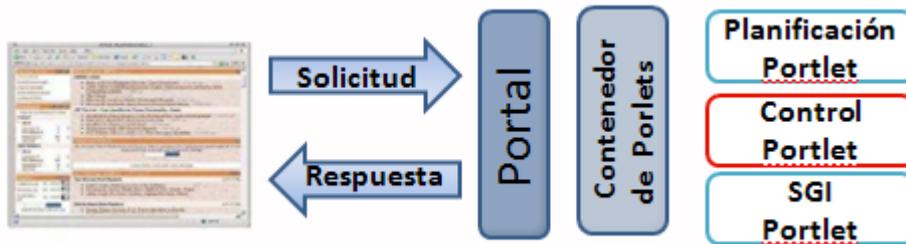


Figura 0.6 Relación entre entidades

1.7 Sistema Gestor de Base de Datos

Los sistemas de gestión de bases de datos o SGBD (en inglés Database Management System, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. El objetivo general de los sistemas de gestor de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. Entre los más utilizados en el mundo se encuentran: Microsoft SQL y PostgreSQL que poseen entre otras las siguientes ventajas:

- Simplifican la programación de equipos de consistencia.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL es una herramienta muy potente para los desarrolladores de sistemas de bases de datos, tiene todo de lo que carece MySQL; posee transacciones,

integridad referencial, vistas y multitud de funcionalidades, pero es un poco más lento y pesado que MySQL, aunque, en las últimas versiones del mismo, esto ha mejorado mucho. Con la aparición de nuevas versiones, los desarrolladores de PostgreSQL argumentaron que empezaba una nueva era: más rápido, más fiable. Han incorporado una funcionalidad llamada MVCC (Multiversion Concurrency Control), con lo que los bloqueos de escritura actúan sólo en la sesión del cliente que lo esté utilizando, no en las de los demás clientes. [24]

1.8 Metodologías para el desarrollo del Software

La metodología es el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevas aplicaciones informáticas. Normalmente, consiste en un conjunto de fases descompuestas en subfases (módulos, etapas, pasos, etc.) con el fin de hacerlo más predecible y eficiente. Esta descomposición del proceso de desarrollo guía a los desarrolladores en la elección de las técnicas que debe elegir para cada estado del proyecto, y facilita la planificación, gestión, control y evaluación de los proyectos.

Una metodología, por tanto, representa el camino para desarrollar aplicaciones informáticas de una manera sistemática. Los desarrolladores de software cuando comienzan un proyecto siempre entran en divergencias para escoger cuál metodología es la más apropiada para su desarrollo.

Por estar especialmente orientadas para proyectos pequeños, las metodologías ágiles constituyen una solución a medida para ese entorno, aporta una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Las mismas son sin duda uno de los temas recientes en ingeniería de software que están acaparando gran interés.

A continuación se puede encontrar algunas características de estas metodologías.

1.8.1 Programación Extrema (Extreme Programming, XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, a la vez promueve el trabajo en equipo, al preocuparse por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad

en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [25-26]

1.8.2 ICONIX

ICONIX es una metodología de desarrollo de software que predice ambos procesos del Proceso Unificado Racional (RUP), la Programación Extrema (XP) y el desarrollo del software Ágil. Como RUP, el proceso de ICONIX utiliza UML pero más ligero. Al contrario de los XP y los acercamientos ágiles, ICONIX proporciona requisito suficiente y documentación del plan, pero sin la parálisis del análisis, además es una metodología relativamente pequeña al igual que XP, pero sin descartar las etapas de análisis y diseño. Utiliza UML en sus etapas, de modo que se pueden seguir los requerimientos y adaptarse a nuevos cambios. El método ICONIX mantiene una atención sostenida en la rastreabilidad de los requerimientos. [27]

En el proceso de ICONIX se manejan casos de uso como en RUP, aunque le falta un poco para llegar al nivel de este. También es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño que hace este. En su proceso también hace uso aerodinámico del UML mientras guarda un enfoque afilado en el seguimiento de requisitos. El manejo de casos de uso, produce un resultado concreto, específico y fácilmente entendible. La Figura 1.7 muestra el cuadro del proceso. El diagrama retrata la esencia del enfoque aerodinámico al desarrollo del software, que incluye un juego mínimo de diagramas de UML y algunas valiosas técnicas que se toman de los casos del uso para codificar rápida y eficazmente. El enfoque es flexible y abierto; siempre se puede seleccionar de los otros aspectos del UML para complementar los materiales básicos.

Sus principales características son:

- Iterativo e incremental: varias iteraciones ocurren entre 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 caso de uso, diagrama de secuencia y de colaboración.
- Modelo de Dominio: con los requisitos se construye el diagrama de clases, que representa el modelo estático del sistema.
- Prototipación Rápida: se usa para simular el diseño del sistema. Se espera que los usuarios lo evalúen como si fuera el sistema final. Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El proceso se repite y finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias, o cuando es evidente que no se obtendrá mayor beneficio con una iteración adicional.

Las fases que establece ICONIX para el desarrollo de un software son:

1. Análisis de requisitos
2. Análisis y diseño preliminar
3. Diseño
4. Implementación

En la siguiente figura se muestra un esquema de la metodología ICONIX:

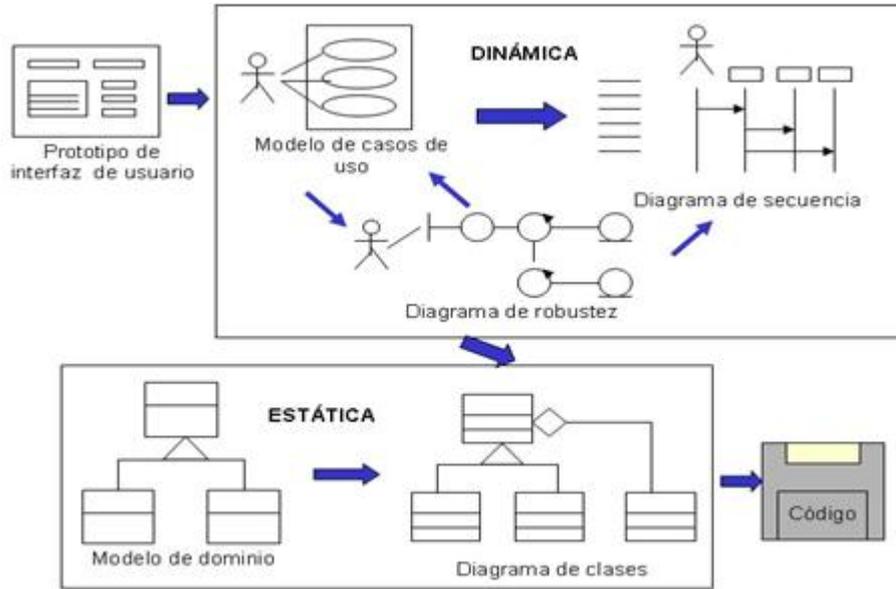


Figura 0.7 Esquema de la metodología ICONIX

1.8.3 Proceso Unificado de Rational (RUP)

Proceso Unificado de Desarrollo (RUP, Rational Unified Process por sus siglas en inglés). La misma es un proceso de desarrollo de software orientado a objeto que conjuntamente con el empleo del Lenguaje Unificado de Modelado (UML, Unified Model Language por sus siglas en inglés) para la descripción del sistema, conforman una potente herramienta para el ingeniero en sistemas. RUP aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y organizaciones. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje modelado común, el Lenguaje Unificado de Modelado.

Conclusiones del Capítulo

Al concluir el presente capítulo es posible constatar los fundamentos teóricos que sirven de base a la investigación. Mediante el estudio realizado se determinaron las principales deficiencias que presenta la empresa en cuanto al proceso de control de la ejecución de los proyectos, concluyendo que:

- El modelo Cliente/Servidor es el más adecuado a las necesidades de los proyectos de la empresa de Ingeniería y proyecto del níquel CEPRONIQUEL, por las características del flujo de información, su arquitectura y la necesidad inminente de un eficiente control de la ejecución de los proyectos.
- El uso de Java como lenguaje de programación y el PostgreSQL como sistema gestor de base de datos, constituyen una excelente opción para el desarrollo de sistemas Web con esta arquitectura.
- Para el desarrollo del sistema Web, se empleará el framework ZK, el cual permite el desarrollo de aplicaciones Web enriquecidas y la construcción de interfaces ricas para el usuario. Además, se usarán los frameworks Hibernate y Spring, los cuales proveen el marco para la implementación del sistema.
- La metodología ICONIX fue seleccionada teniendo en cuenta las características del sistema que se desarrolla así como las bondades que presenta la misma.

Capítulo 2. Descripción de la solución propuesta

En este capítulo se abordarán los temas relacionados con la ingeniería del software, se hace uso de los artefactos que propone la metodología ICONIX se modela y construye la aplicación propuesta. Se describe el funcionamiento del proceso de control de proyectos en la empresa CEPRONIQUEL de Moa a través del Modelo de Dominio, así como el Modelo de Casos de Uso con los Diagramas de Robustez y de Secuencia correspondientes.

2.2 Modelo del Dominio

El modelo del dominio captura los tipos de objetos y conceptos fundamentales relacionados con el problema que el sistema pretende resolver, se conocen de antemano los requerimientos del cliente. Los objetos representan cosas o eventos que ocurren en el entorno en el que trabaja el sistema. Muchos de estos objetos del dominio o clases son obtenidos a partir de la especificación de los requerimientos.

Listar los requerimientos funcionales:

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar, sin tomar en consideración ningún tipo de restricción física [28]

El sistema será capaz de favorecer el proceso de control de la información relacionada con los proyectos de la empresa CEPRONIQUEL de Moa.

- a) Permitirá al Jefe de Proyecto visualizar y modificar el estado estados de los proyectos (conciliados, contratados, ejecutados).
- b) Permitirá al Jefe de Proyecto mostrar las tareas de cada Plan de Diseño General del Proyecto.
- c) Permitirá al Jefe de Proyecto crear reportes sobre el cumplimiento de las tareas de cada proyecto.
- d) Permitirá al Jefe de Proyecto mostrar los responsables de cada tarea.
- e) Permitirá al Jefe de Proyecto redistribuir las tareas del Plan de Diseño General por responsable.
- f) Permitirá al Jefe de Proyecto eliminar tareas en Plan de Diseño General del proyecto.
- g) Permitirá al Jefe de Proyecto mostrar el nivel de cumplimiento de cada tarea.
- h) Permitirá al Jefe de Proyecto controlar y modificar la agenda de trabajo.

2.2.1 Definición de los principales objetos del modelo del dominio

A partir de la lista de requerimientos se realiza una inspección gramatical de los elementos señalados, se ordenan alfabéticamente y se eliminan términos duplicados e innecesarios que caen fuera del alcance del modelo del dominio. Se explica brevemente el significado de cada entidad en la Tabla 2.1

Objeto	Definición
Proyecto	Es el proceso de trabajo grupal que se extiende por todas las áreas de la empresa, en determinada cantidad de tiempo para lograr el producto final de una obra de ingeniería.
Jefe de Proyecto	Es el encargado de administrar los procesos internos de cada proyecto, donde realmente se efectúa el trabajo más importante organizando y controlando las tareas.
Controlador de la calidad	Está compuesto por un conjunto de especialistas ubicados en cada una de las áreas, que tienen la responsabilidad de controlar que el trabajo se realice según las normas de calidad internacional.
Plan de diseño general	Es la integración de las tareas a ejecutar de manera lógica y cronológica de cada proyecto.
Tareas	Actividades a desempeñar en un proyecto, las cuales serán desarrolladas por los trabajadores.
Estados	Son las fases en que se puede encontrar un proyecto o una tarea durante su ciclo de vida.
Usuarios	Los usuarios del sistema son: Jefe de Proyecto, Controlador de la Calidad y trabajadores de la empresa.
Responsables	Trabajadores encargados de realizar una tarea.

Tabla 2.1 Glosario de términos

2.2.2 Concepción del Modelo del Dominio

Luego de obtener las palabras claves de la lista de requerimientos, se ordenan para eliminar los términos duplicados e innecesarios que caen fuera del área que abarca el dominio. El diagrama correspondiente surge para ilustrar las relaciones entre estos objetos. (Ver Figura 2.1)

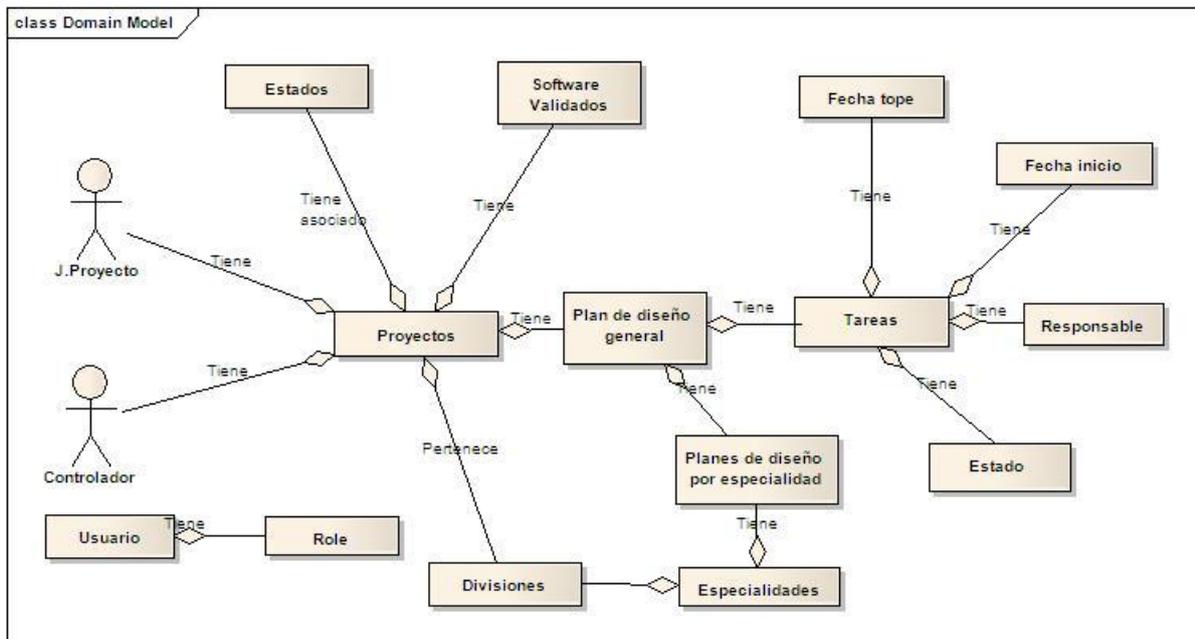


Figura 2.1 Diagrama del Dominio

2.3 Modelación de los casos de uso

Dentro del proceso de ICONIX, uno de los primeros pasos involucra la construcción del modelo de casos de usos (CUS). Se captura los requisitos de comportamiento de sistema de una manera estructurada. En el texto de los casos de usos se utilizan todos los términos utilizados en el dominio. Se realiza una representación de los diagramas de los casos de uso agrupados en paquetes (Figura 2.2).

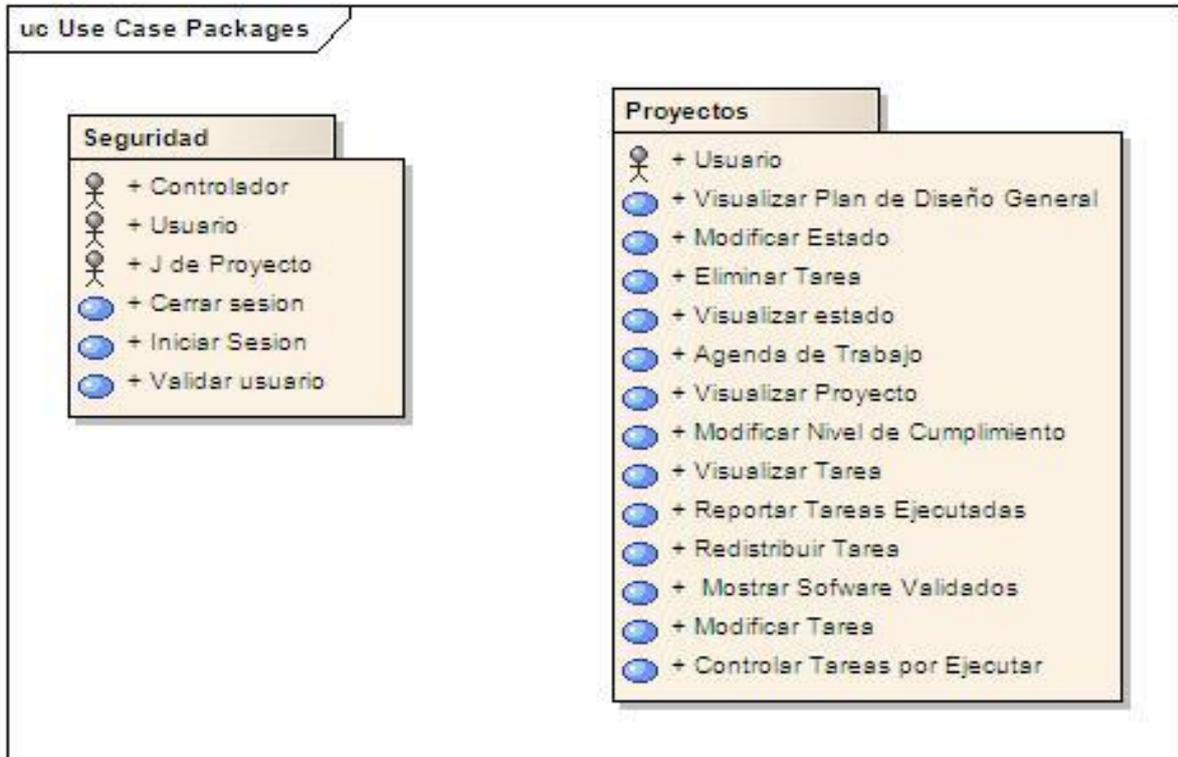


Figura 2.2 Diagrama de paquetes de casos de usos del sistema

A continuación se muestran los diagramas de casos de uso del sistema.

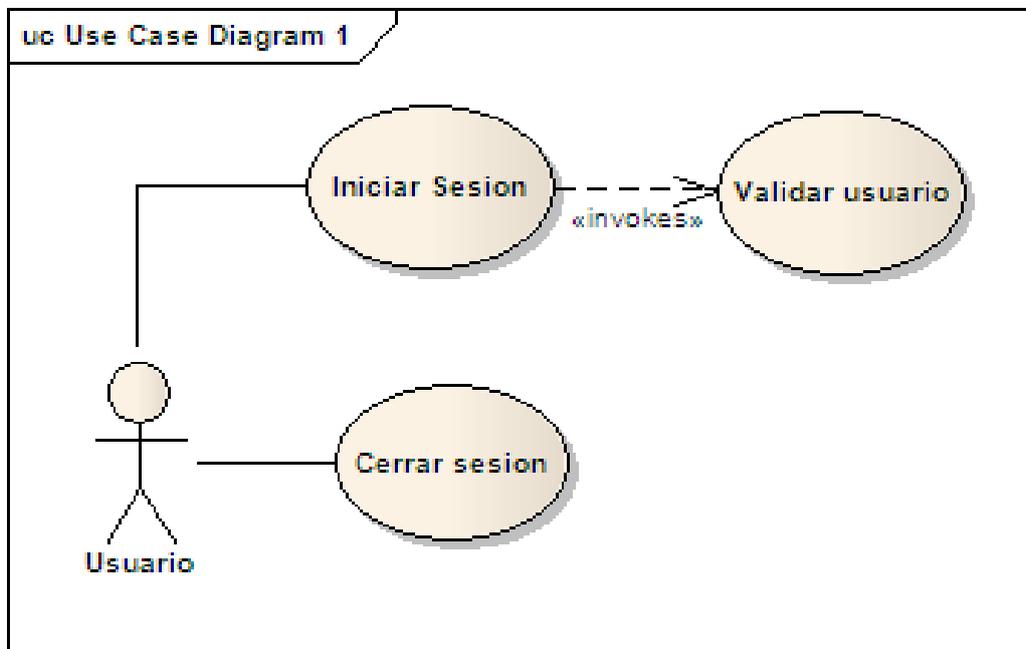


Figura 2.2 Diagrama de casos de uso del paquete Seguridad

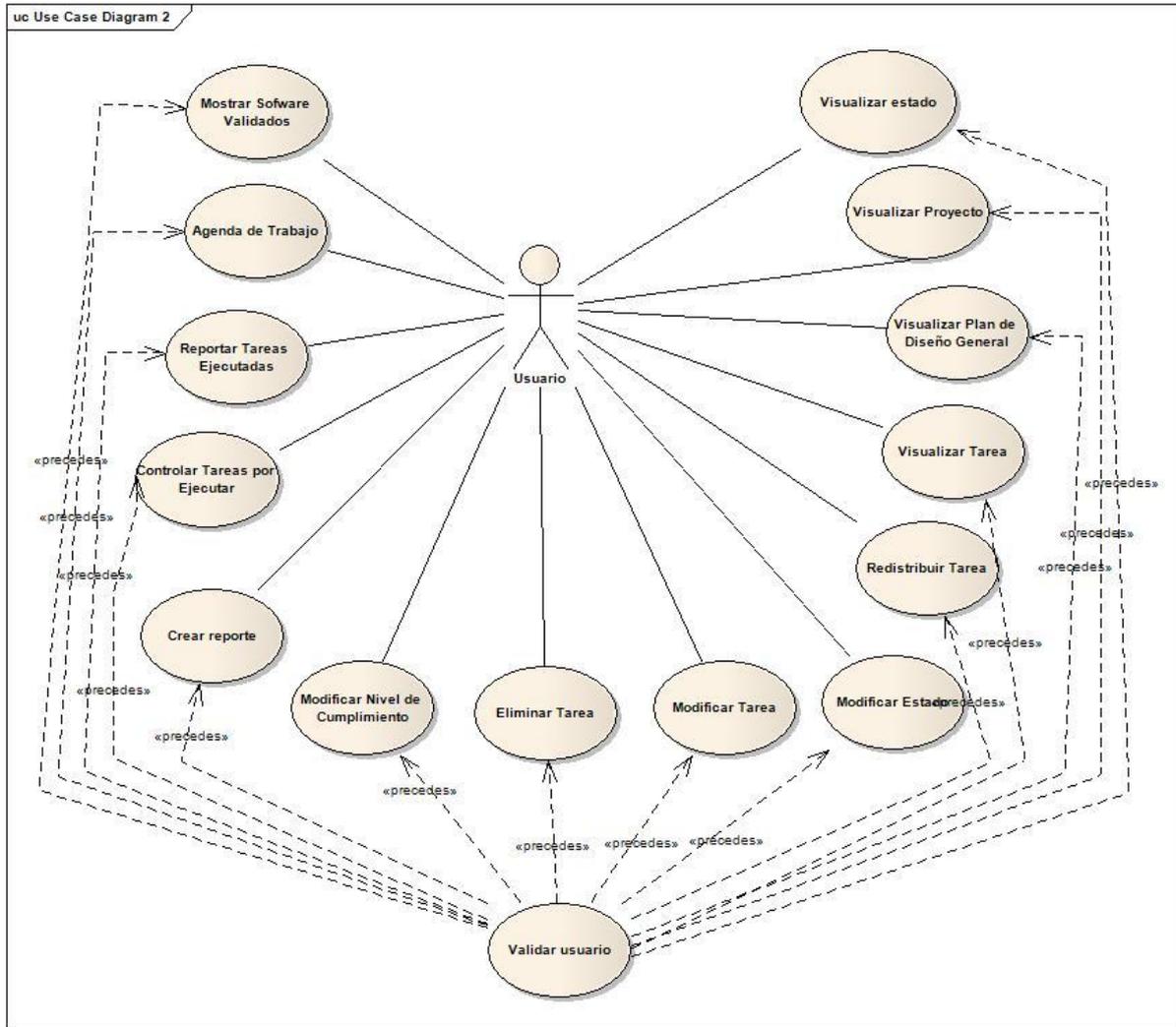


Figura 2.3 Diagrama de casos de uso del paquete proyecto

2.3.1 Actores del sistema

Cada diagrama de caso de uso representando anteriormente, es iniciado por un actor, los actores reflejan todas las entidades que deben intercambiar información con el sistema, el actor es llamado como “Usuario”, aunque le es dado el nombre del rol específico que desempeña en el sistema.

Se identificaron tres actores:

Objeto	Definición
Jefe de Proyecto	Es el encargado de administrar los procesos internos de cada proyecto, donde realmente se efectúa el trabajo más importante organizando y controlando las tareas.

Controlador de la calidad	Está conformado por un conjunto de especialistas ubicados en cada una de las áreas, que tienen la responsabilidad de controlar que el trabajo se realice con la calidad que identifica a la empresa según las normas de calidad internacional.
usuario	Es la persona que interactúa con el sistema

Tabla 2.2 Descripción de los actores del sistema

Se escogieron 3 casos de usos ilustrados anteriormente en el paquete proyecto para mostrar los pasos que se siguieron según la metodología ICONIX:

1. Mostrar Plan de Diseño General
2. Redistribuir tareas.
3. Generar reportes.

2.3.2 Descripción de los casos de uso

Las descripciones de los casos de usos están escritas en párrafos simples, en voz activa, siguiendo un flujo acción del actor/ respuesta del sistema siguiendo una estructura de oración sustantivo-verbo-sustantivo. Se reflejan dos cursos, un curso básico que muestra como se debe realizar el proceso y un curso alterno para describir los posibles errores o sucesos paralelos que pueden ocurrir durante las distintas operaciones.

A continuación se muestran las descripciones de los casos de usos antes mencionados.

Paquete Proyecto: CUS Mostrar Plan de Diseño General
<p>Curso básico</p> <p>El usuario (Jefe de Proyecto, Controlador de la Calidad, trabajador) hace clic en la opción plan de diseño general de la vista calendario, el sistema muestra la interfaz del calendario completo correspondiente a los proyectos de acuerdo al estado en que este se encuentre (conciliado, contratado, en espera de aprobación) seleccionando previamente el proyecto que dese visualizar se mostrarán las tareas organizadas cronológicamente con los detalles de cada una como la fecha de inicio, terminación y responsable.</p>

Curso alterno

No existen proyectos contratados ni conciliados: el sistema muestra el mensaje, “no hay proyectos contratados ni conciliados”.

Tabla 2.4 Descripción del CUS mostrar Plan de Diseño General

Paquete Proyecto: CUS Redistribuir tareas

Curso básico

El usuario (Jefe de Proyecto o Controlador de la Calidad) seleccionando previamente el proyecto en la vista calendario hace doble clic en la tarea que desea redistribuir y en las propiedades de esa tarea agrega el nuevo responsable de su ejecución y algún comentario de la nueva decisión, si desea cambiar la tarea de fecha solo tiene que arrastrar y soltar la tarea en la casilla del calendario donde desea ubicarla nuevamente.

Curso alterno

No existe ninguna decisión de modificar ni reubicar la tarea en el Plan de Diseño General.

Tabla 2.5 Descripción del CUS Redistribuir tareas

Paquete Proyecto: CUS Generar reporte

El usuario (Jefe de Proyecto o Controlador de la Calidad) da clic en la opción reporte de operaciones seleccionando el proyecto para emitir en un diagrama de Grantt del formato deseado el nivel de ejecución de las tareas correspondientes a dicho proyecto.

Curso alterno

No existen proyectos en ejecución: el sistema muestra el mensaje, "no hay proyectos ejecutándose en este momento".

Tabla 2.6 Descripción del CUS Generar Reporte

2.4 Estudio de Factibilidad

A raíz de la captura de requerimientos y el modelo del sistema obtenidos anteriormente, se hace necesario realizar un estudio de factibilidad que permita recopilar los datos fundamentales sobre el desarrollo del producto informático.

Estos datos contribuyen a tomar la importante decisión de si es posible el desarrollo y la implementación del sistema propuesto. La factibilidad es la disponibilidad de los recursos necesarios para llevar a cabo los objetivos o metas señalados. [29]

2.4.1 Factibilidad técnica

Al ser desarrollado el sistema por un estudiante en opción por el título Ingeniero Informático, la Facultad de Informática y Matemática facilita la mayoría de los recursos materiales y humanos necesarios para su correcta terminación o sea que el problema se puede resolver con los medios actualmente existentes en su mayoría en la facultad. Como resultado de este estudio se determina que en los momentos actuales, se cuenta con la infraestructura tecnológica para el desarrollo del sistema propuesto.

2.4.2 Factibilidad operacional

El sistema está concebido para que los usuarios finales, ante la implantación del sistema, acepten la forma en que el mismo concibe el flujo de información en el proceso de control de las tareas correspondientes a los Planes de Diseño Generales de cada proyecto en cualquiera de sus fases; así como la generación de reportes, que facilita los flujos de informaciones automáticamente de manera que se reducen las posibilidades de resistencia al cambio. El sistema propuesto no perjudicará el trabajo que ahí se realiza, excepto la informatización y automatización de algunas de las tareas, o sea, el sistema puede ser explotado sin ningún problema de tipo operacional.

2.4.3 Factibilidad económica

En todo proyecto de desarrollo de software el análisis de factibilidad es una fase de gran importancia pues si el sistema no es factible económicamente para la organización que lo desarrolla, entonces no debe ser elaborado, y mucho menos implantado [30]. En la determinación del costo asociado al desarrollo del proyecto, se emplea el Modelo Constructivo de Costos (COCOMO 2, por sus siglas en inglés), que es una herramienta utilizada para la estimación de costos en personas y tiempo. A continuación, se exponen los resultados de aplicar los pasos que propone COCOMO 2. (Ver más adelante) [31].

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:

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

$$E = 1,0668$$

PM ≈ 49 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 ≈ 10 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 ≈ 2 hombres.

Lo que muestra que el proyecto necesitaría 2 hombres para realizarlo en aproximadamente 10 meses.

Costo por mes (CHM):

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

$$SP = \$250 \text{ Salario Promedio}$$

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

$$CHM = \$ 500 \text{ Costo de hombres por mes para 2 trabajadores}$$

Costo total (CT):

$$CT = CHM * TDEV$$

$$CT \approx \$5, 000.00$$

El costo estimado del proyecto es de: \$10, 000.00

Variables	Valor	Aproximado
PM(Esfuerzo)	49, 2712	49
TDEV(meses)	10	10
Size(Tamaño estimado(KSLOC))	17,51	
E	1,133	
F	0,325	
Salario Medio	250	
Cantidad de hombres	3,95	2

Tabla 2.7 Resultados COCOMO II

2.5 Análisis de Robustez

Luego de haber realizado la descripción de los casos de uso y la representación grafica de sus diagramas, el siguiente paso es el análisis de robustez, siendo este imprescindible para comprender el diseño del sistema. El análisis de robustez está seriamente ligado a los casos de uso, el mismo presenta un vistazo preliminar sobre cómo diseñar un software que implemente un caso de uso dado. Uno de los propósitos principales de esta actividad es descubrir, cuando no se tienen, todos los objetos necesitados para la realización del producto informático.

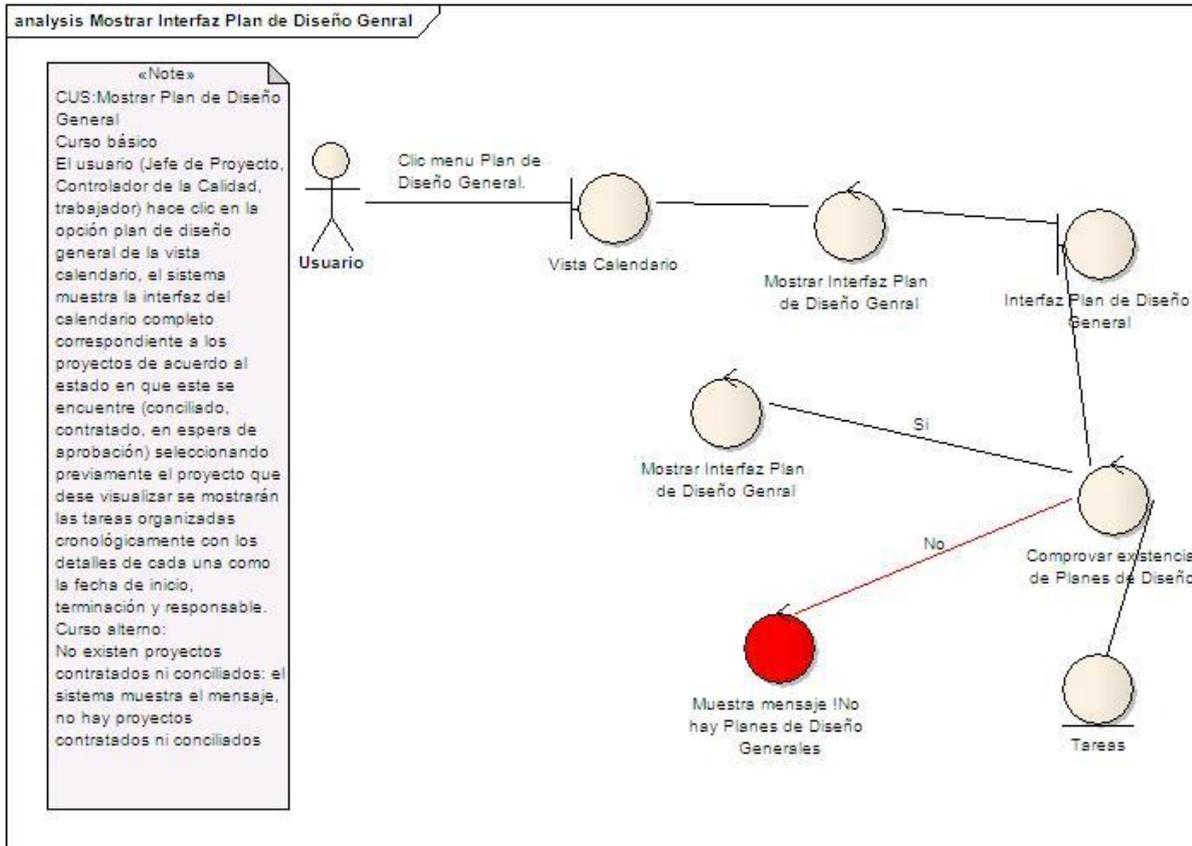


Figura 2.4 Diagrama de robustez CUS Mostrar Plan de Diseño General

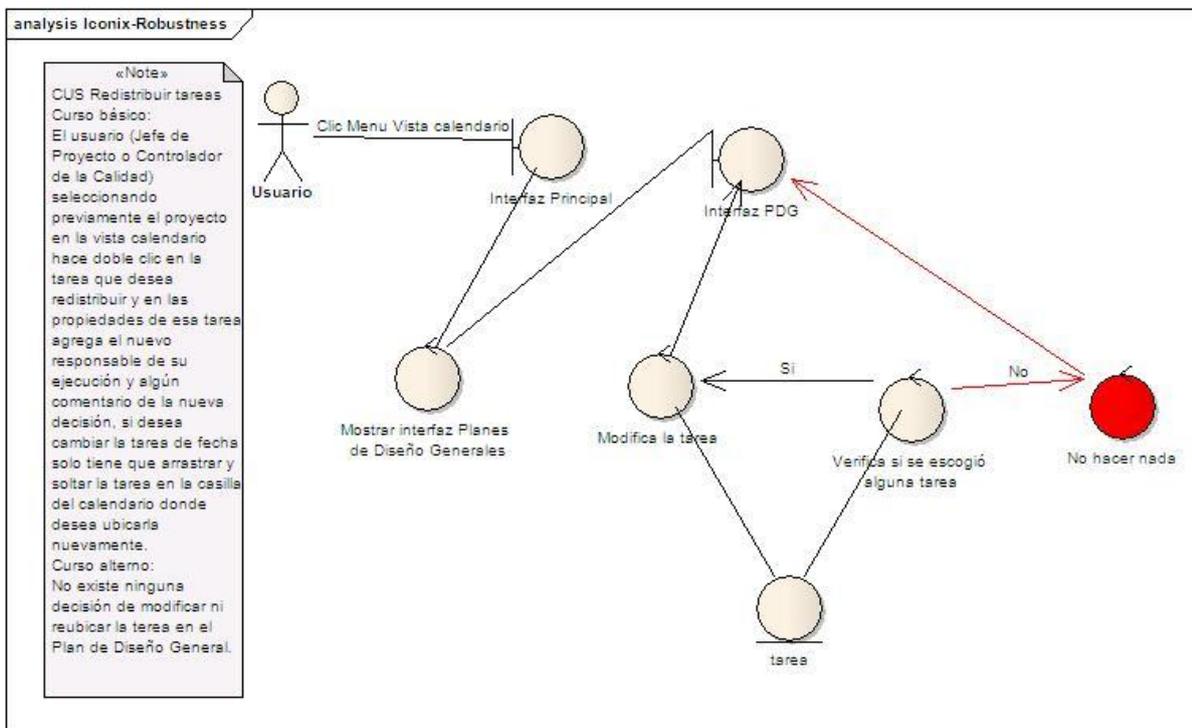


Figura 2.5 Diagrama de robustez CUS Redistribuir tareas

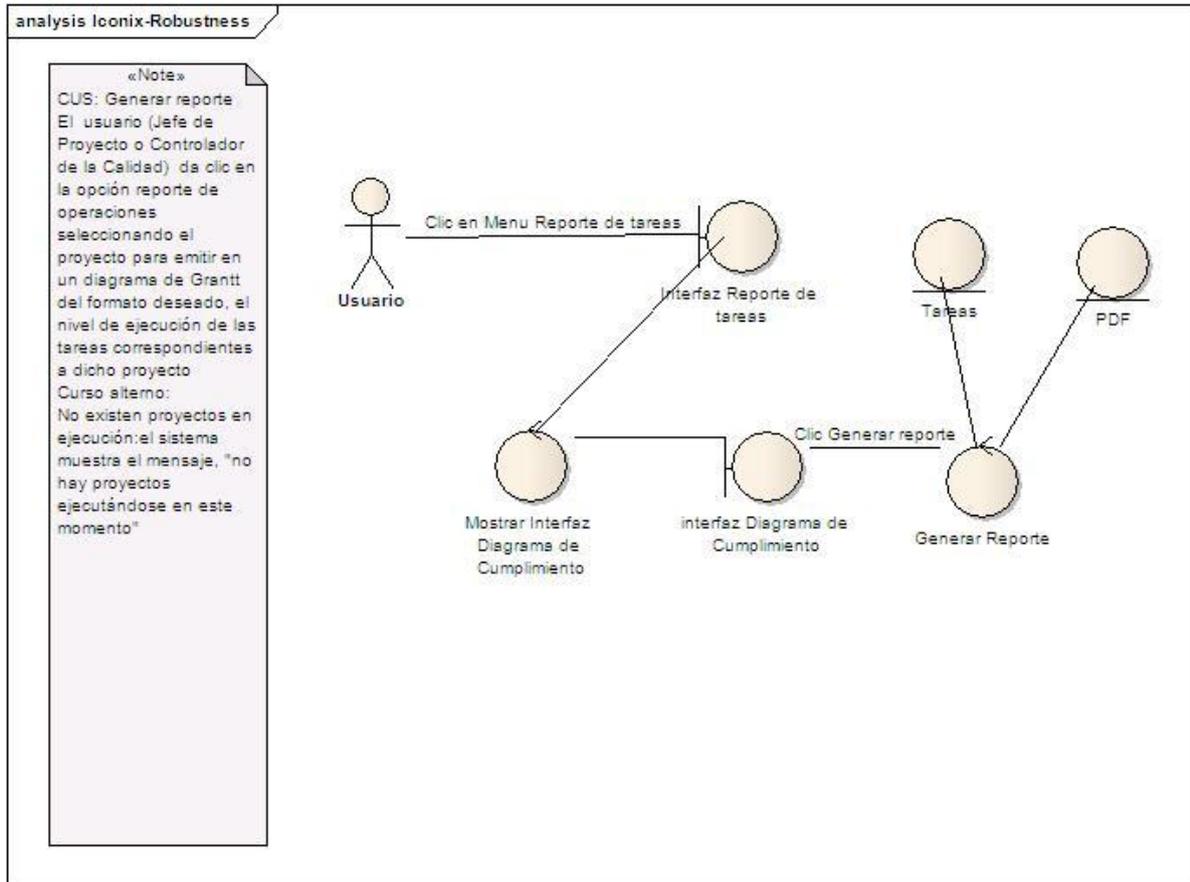


Figura 2.6 Diagrama de robustez CU Generar reporte

2.6 Arquitectura Técnica

La arquitectura técnica es la concepción de las tecnologías utilizadas en el desarrollo de la aplicación, ya que a partir de ello se tiene una consideración general del sistema que se está desarrollando. Además, es conocida como la arquitectura del sistema y de software, con el propósito de describir el sistema que se está intentando construir en términos de estructura, satisfacer los requerimientos del negocio y el nivel de servicio del sistema que se va a desarrollar. La arquitectura abarca dos amplias áreas las cuales son:

1. Requerimientos no funcionales
2. El modelo de despliegue (red y servidores de aplicación, y como se adaptan juntos; la topología del sistema; navegadores soportados)

2.6.1 Requerimientos no Funcionales

Los requerimientos no funcionales definidos limitan al sistema y se caracterizan por hacerlo más atractivo, usable, rápido, seguro y confiable. Se precisan con la intención de obtener el éxito, reflejada en la aceptación de los usuarios finales, así como el buen funcionamiento, la flexibilidad y escalabilidad que proporciona el mismo. Al mismo tiempo, se encuentran vinculados estos requerimientos con los funcionales a pesar de no alterar la funcionalidad del sistema [32]. ICONIX propone que estos deben adaptarse a la arquitectura técnica que va a adoptar la aplicación:

Apariencia o Interfaz externa

- El diseño debe ser agradable y atrayente a los usuarios generalizando la utilización de los colores que caracterizan a la empresa en el manual de identidad corporativa para lograr una mejor concentración.
- La interfaz no debe recargarse con imágenes para proporcionar una navegación cómoda.
- El flujo de trabajo en el sistema debe asemejarse al proceso rutinario llevado a cabo en la empresa.
- Para la interfaz de la aplicación se utilizará el framework ZK.

Usabilidad

- El sistema debe ser accesible desde cualquier lugar de la entidad.
- El sistema debe estar funcionando durante la jornada laboral.
- El diseño del sistema debe ser sencillo para agilizar el tiempo de conexión al mismo.

Portabilidad

- Las herramientas utilizadas para el desarrollo del sistema son tecnología de software libre y a su vez multiplataforma, lo cual le confiere al sistema esta última característica.

Seguridad

- Sólo los usuarios autorizados podrán acceder al sistema.

- Se garantiza que las funcionalidades del sistema se realicen de acuerdo a la actividad definida para cada uno, es decir el nivel de acceso debe ser restringido. La información debe ser actualizada según el personal autorizado.
- No existe información que se pueda obtener sin ser usuario del sistema.
- Sólo el administrador del sistema tendrá acceso a la BD, a los ficheros fuentes del sistema y es responsable de la autorización en general del mismo.

Confiabilidad

- El sistema debe posibilitar la recuperación de información en caso de fallos y/o errores.
- Debe dar facilidad de mantenimiento una vez implantado para posibilitar un perfeccionamiento continuo del sistema.
- Ayuda y documentación en línea.
- Debe contar con un Manual de Usuario y un sistema de ayuda de forma tal que le brinde orientación al usuario respecto a las opciones con que cuenta el sistema, utilizando textos explicativos que indiquen la acción de estas.

Software

- Las máquinas de los clientes deben disponer de un navegador Web, con soporte para JavaScript, con las características necesarias para el uso de la técnica AJAX, se recomienda el Internet Explorer 5.0 o superior, Mozilla FireFox 2.0 o superior u Opera 6 o superior.
- El sistema operativo de la máquina computadora cliente debe ser Windows XP o superior.
- La máquina computadora servidor debe tener Windows 2000 o superior, Servidor Web Tomcat versión 5.5, servidor de BD PostgreSQL 8.4, Java Development Kit, (JDK por sus siglas en inglés) versión 1.6.
- La resolución de pantalla debe ser 800x600.

Hardware

- Para ejecutar el software los requerimientos mínimos de hardware en el cliente son: microprocesador Intel Pentium II a 400 MHz de velocidad de

procesamiento u otro similar, con 128 MB de memoria RAM y un adaptador de red.

- La máquina computadora servidor debe tener 512 RAM o superior y debe ser un Pentium IV con un microprocesador cercano a los 2.4 GHz de velocidad.
- La máquina computadora servidor y las computadoras clientes deben estar conectadas a la red.

2.6.2 Modelo de Despliegue

Dentro de la arquitectura técnica, un paso muy importante es el diagrama de despliegue, el cual representa la correspondencia entre la arquitectura software y la arquitectura hardware. Este modelo se utilizó como entrada fundamental en las actividades de diseño e implementación, debido a que la distribución del sistema tiene una influencia principal en su diseño. Ver Figura 2.6.

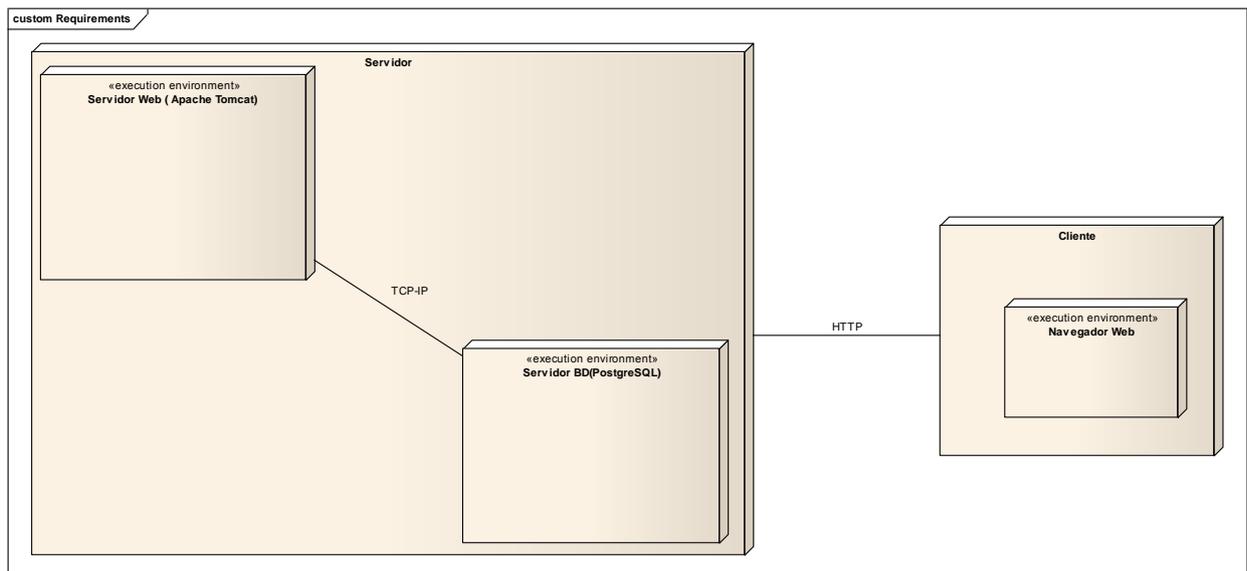


Figura 2.6 Modelo de despliegue

2.6.3 Arquitectura del sistema

En el desarrollo del módulo propuesto se estudiaron los principios de una arquitectura multicapa, la cual aumentaría la mantenibilidad y flexibilidad del sistema, pues a la hora de corregir, perfeccionar o adaptar el software resulta sencillo detectar en qué 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

entre los frameworks ZK y Spring. Todas las peticiones que ocurren entre el navegador y el servidor de aplicaciones son asíncronas, mediante el empleo de AJAX, se logra que las páginas web se ejecuten en el navegador del cliente sin recargar el servidor, lo que aumenta 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), el framework ZK es el encargado de gestionar los controladores correspondientes a las vistas de usuarios. Conjuntamente, ZK interactúa con el framework Spring, el cual provee la seguridad de la aplicación.

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. Ver Figura 2.7.

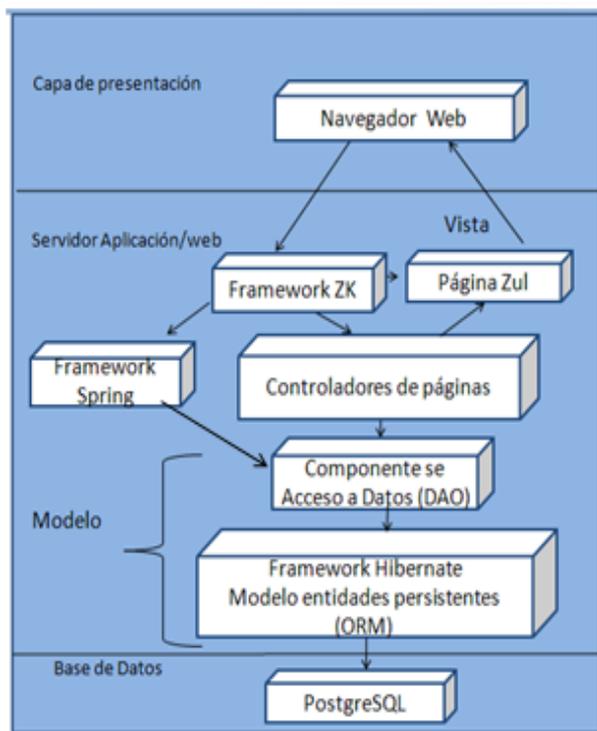


Figura 2.7 Arquitectura del sistema

2.7 Diagramas de Secuencia

A partir de los diagramas de casos de uso y de los diagramas de robustez se define gran parte de los atributos de las clases, aunque no es hasta el diagrama de secuencia donde se empiezan a ver qué métodos llevarán las clases del sistema que se desarrolla, así como las interacciones entre objetos durante el tiempo de vida del caso de uso.

El diagrama de secuencia tiene varios objetivos elementales, entre ellos los más importantes son:

- Asignar comportamiento a las clases.
- Mostrar en detalle como las clases interactúan entre sí durante el tiempo de vida del caso de uso.
- Terminar la distribución de las operaciones entre clases.

A continuación se mostrarán los principales diagramas de secuencia de los casos de uso antes mencionados.

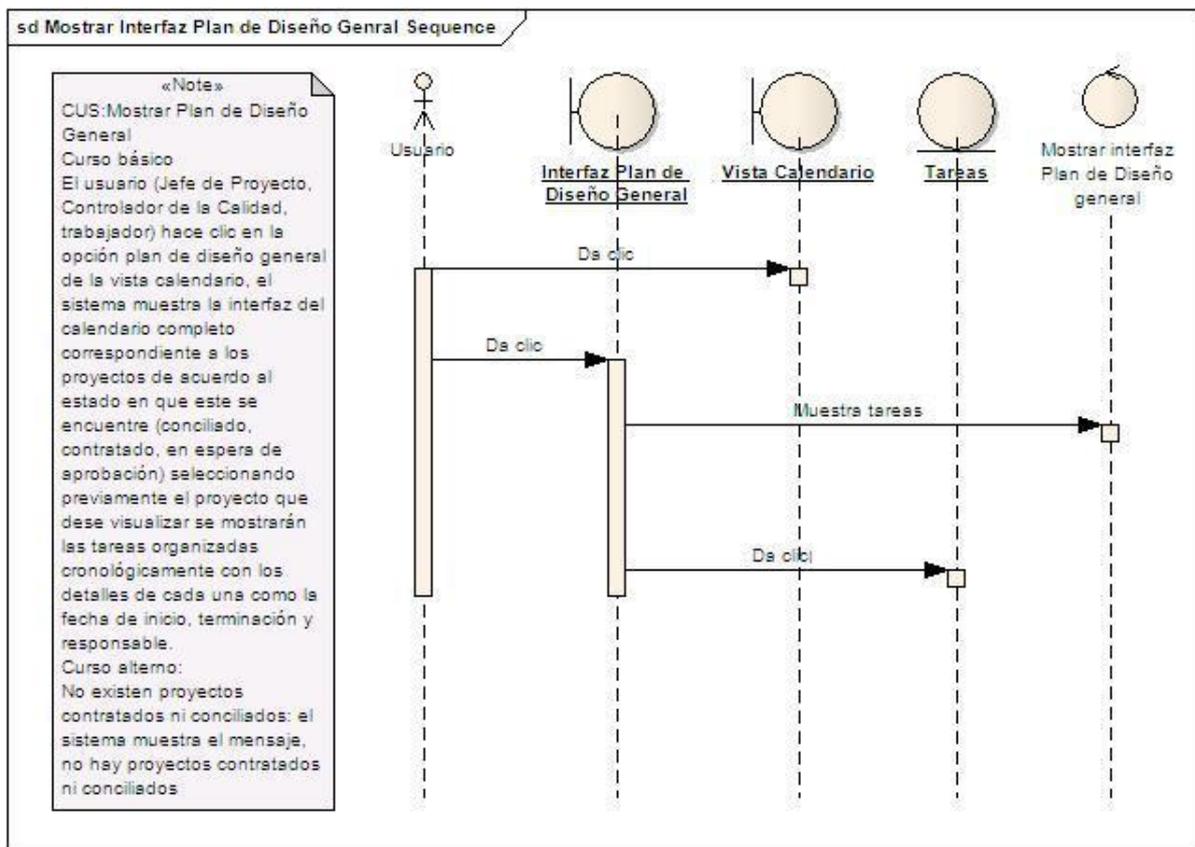


Figura.2.8 Diagrama de secuencia del CU Mostrar Plan de Diseño General

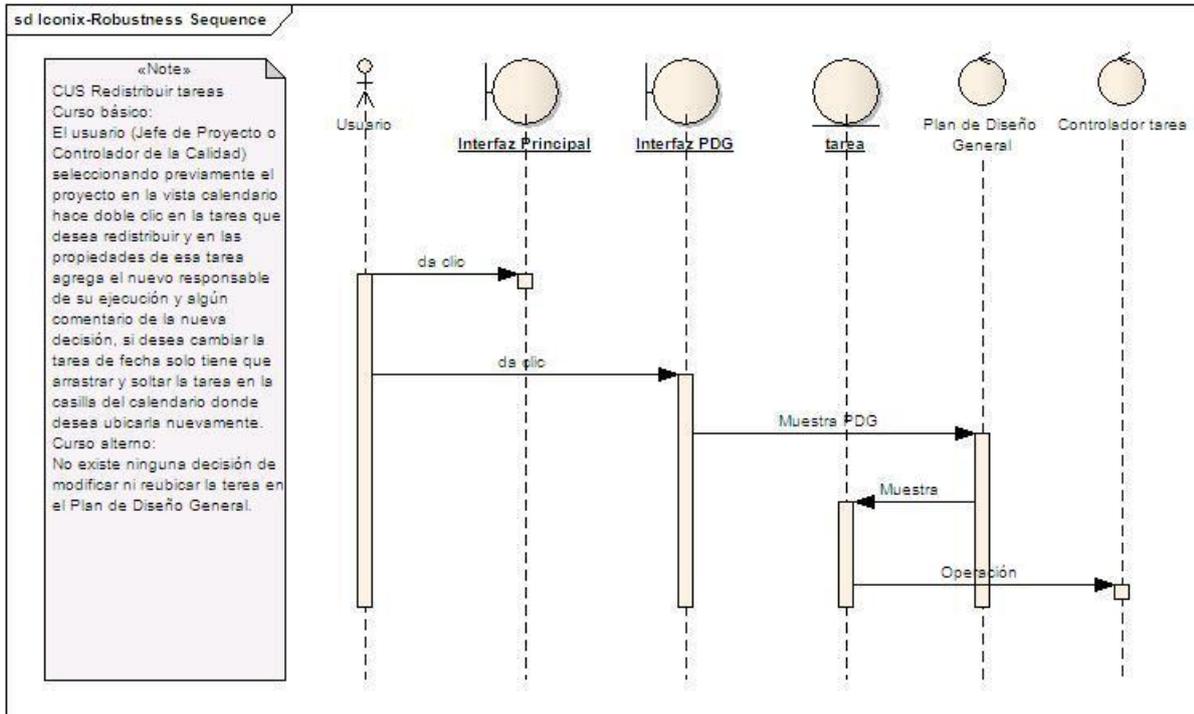


Figura.2.9 Diagrama de secuencia del CU Redistribuir tareas

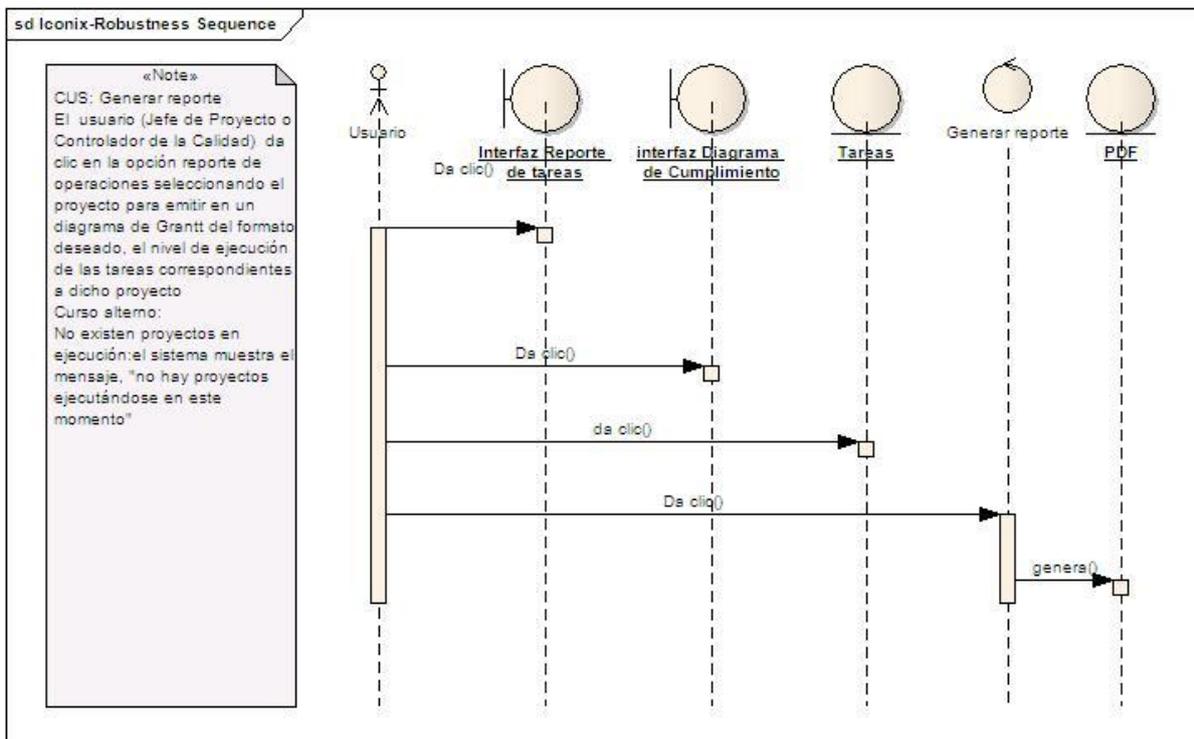


Figura.2.10 Diagrama de secuencia del CU Generar reporte

2.8 Diagrama de clases persistentes

El proyecto presenta gran cantidad de clases, después de estudiar sus posibles relaciones se determinó organizarlas por paquetes; después de un análisis para valorar su posible agrupación, se decide dividir en paquetes para su representación en el documento. A continuación se muestra el diagrama de clases persistente.

(Este es el último no se ha completado)

Figura 2.10 Modelo de clases persistentes.

2.9 Modelo de Implementación

La implementación se lleva 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. [33]

En este capítulo se describe el módulo propuesto desde el punto de vista de cómo lo verán los usuarios finales. Se exponen las funcionalidades de las principales pantallas y las interfaces escogidas.

2.9.1 Descripción del sistema

Un usuario (Jefe de Proyecto o Controlador de la calidad) comienza a interactuar con el sistema, cuando se autentica en la pantalla principal de la Intranet que a continuación se presenta (ver Figura 2.17):

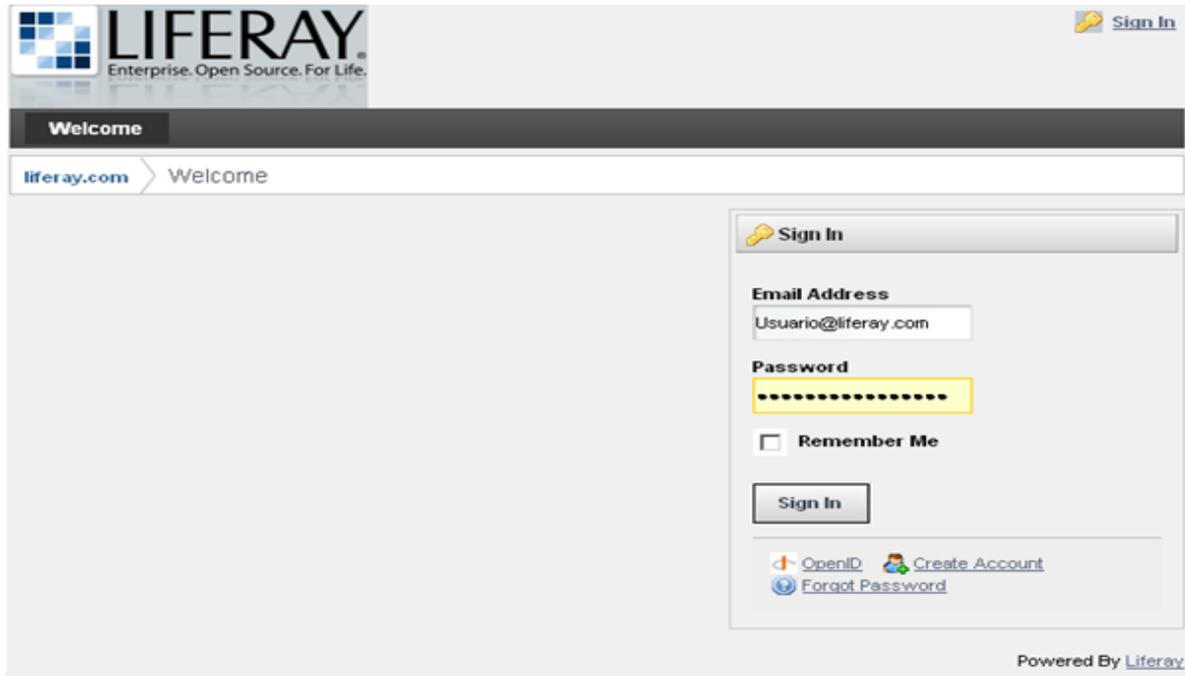


Figura 2.12 Interfaz de autenticarse

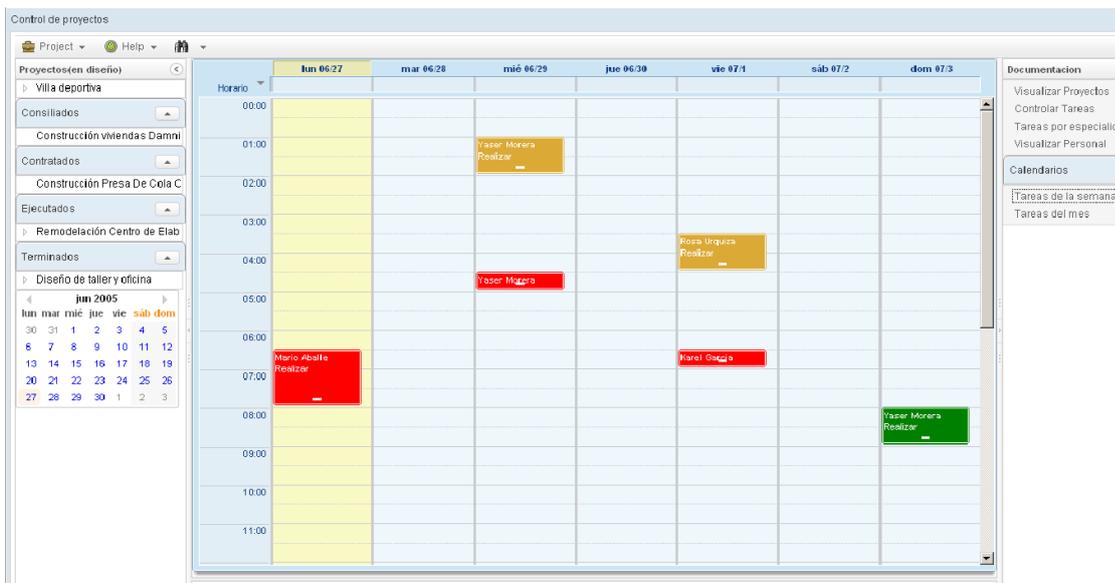


Figura 2.13 Calendario de carga de trabajo (vista semanal)

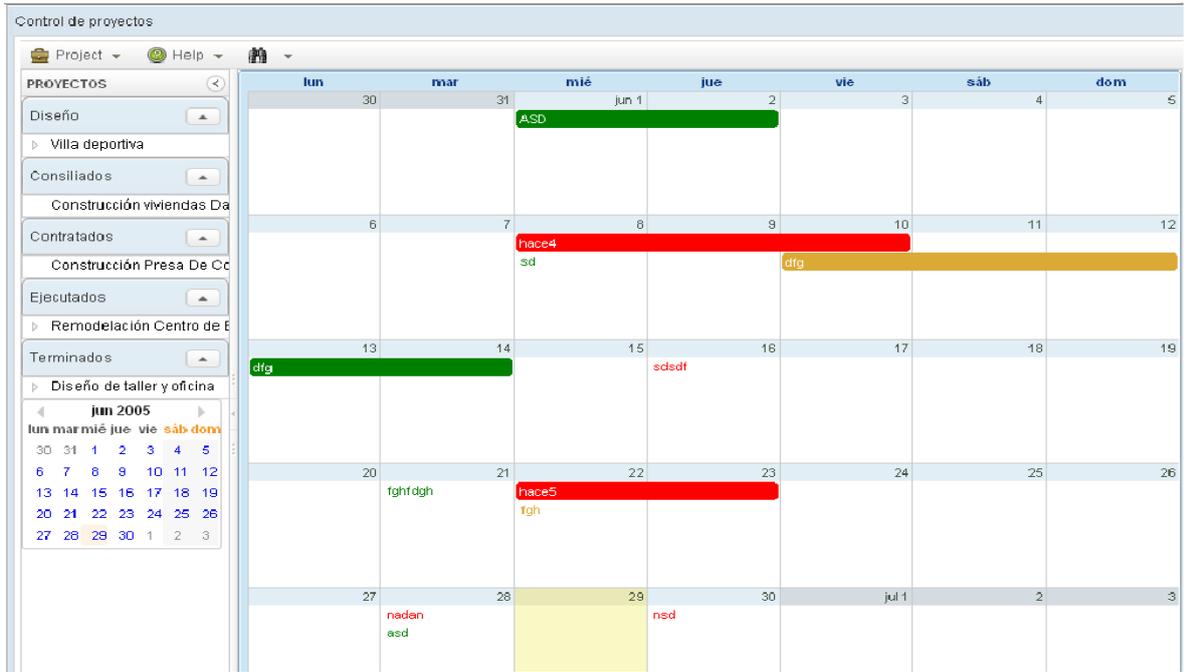


Figura 2.14 Calendario de carga de trabajo (vista mensual).

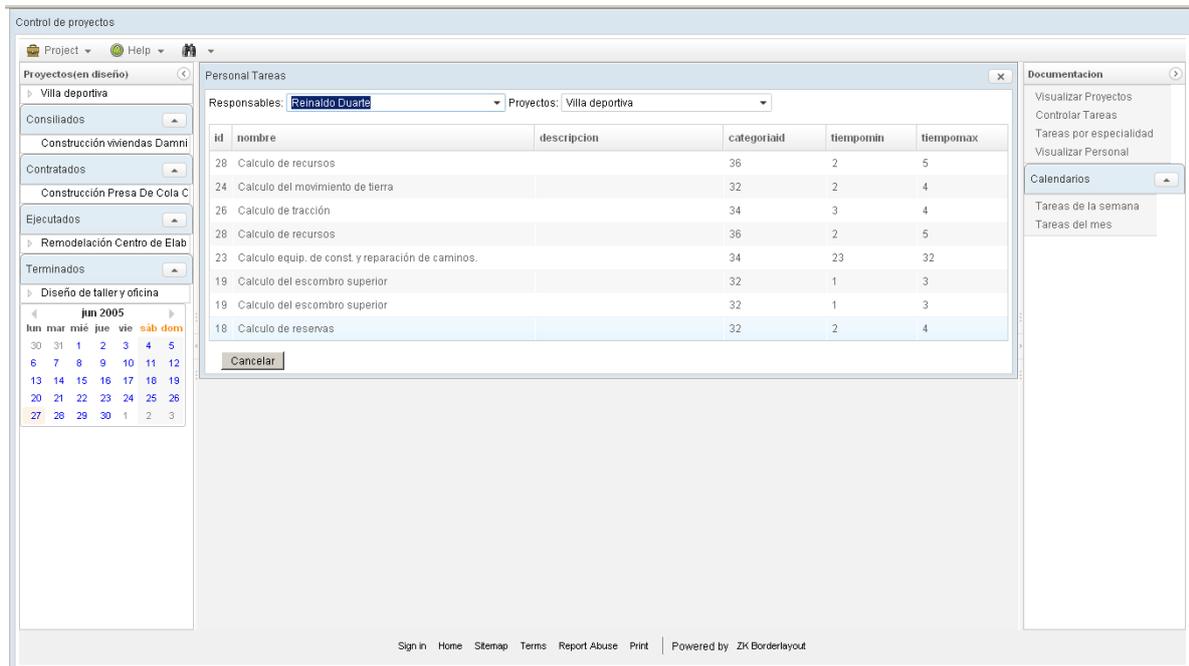


Figura 2.15 Interfaz controlar tareas por proyecto.

2.10 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 de un software. 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. (Ver Anexo 1)

2.11 Valoración de Sostenibilidad

Para desarrollar un sistema informático es importante tener presente el impacto social, económico, tecnológico y ambiental que este tendrá, para poder determinar si el producto informático es sostenible y así garantizar la calidad del mismo. En la presente investigación se adoptó el procedimiento que plantea que esta valoración debe realizarse a través de cuatro dimensiones de gestión de sostenibilidad: administrativa, socio-humanista, ambiental y tecnológica.

Se entiende por valoración de sostenibilidad de un producto informático al proceso de evaluación de impactos ambientales, socio humanistas, administrativos y tecnológicos de dicho producto, previsible desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente. [34]

Dimensión administrativa:

La dimensión administrativa incluye aspectos como: ahorro, gastos, calidad de la producción y los servicios, administración de recursos, toma de decisiones administrativas, entre otros. En el estudio de sostenibilidad se hace un estudio de factibilidad del proyecto. Para ello se utiliza el Modelo Constructivo de Costos II (COCOMO II, por sus siglas en inglés). COCOMO II, es una herramienta utilizada para la estimación de algunos parámetros (costes en personas, tiempo, etc.) en el diseño y construcción de programas, y de la documentación asociada requerida

para desarrollarlos, operarlos y mantenerlos, es decir, en la aplicación práctica de la Ingeniería del Software. [35]

En el caso de la confección del módulo informático propuesto para control de los proyectos de la empresa CEPRONIQUEL de Moa, no habrá gastos por mano de obra, ya que su realización será por un estudiante de la carrera de Ingeniería Informática como parte de un Trabajo de Diploma. Para su desarrollo se utilizará en su mayoría equipamiento existente en la universidad, por lo que no se incurrirá en gastos adicionales, que es ya utilizado en actividades afines, por lo que tampoco se incrementará el consumo de electricidad; aunque su costo de desarrollo está valorado en 12 000.00 pesos.

Una vez implantado, se ahorrarán recursos de gran importancia como el tiempo, lapiceros y hojas, pues al emplearse medios computarizados se reducirá considerablemente el consumo de estos.

La calidad del producto final se verá favorecida, debido a que el sistema incide en una de las aristas fundamentales del proyecto, como lo es el tiempo de terminación al eliminar el margen de error a la hora de realizar el control, que se hace con la mayor brevedad posible.

No genera ingresos directamente a la entidad donde se realiza, ya que su finalidad es facilitar el trabajo de los trabajadores de la empresa.

El desarrollo del proyecto no incurre en gastos económicos, pues se hace uso de software libre, evitando el pago de licencias. Por lo tanto se puede concluir que en la dimensión administrativa el SI es sostenible.

Dimensión socio-humanista:

En la dimensión socio-humanista se analiza el SI según aspectos como: modo de vida, desarrollo de un grupo social, satisfacción de las necesidades sociales, formación ético humanista de los gestores del SI, la ciencia y la tecnología como procesos sociales.

Con respecto al impacto socio-humanista del SI, se puede asegurar que una vez implantada la aplicación el Jefe de Proyectos, los Controladores de la Calidad y de forma general la empresa, contarán con acceso a información fiable y actualizada

del estado de ejecución de los proyectos; posibilidad que se dificulta en la actualidad.

Por otro lado, no trae consigo la disminución de empleo en la organización, por el contrario, agiliza el trabajo de las personas en la empresa, aunque no resulta preciso emplear más personal. El trabajo se realizará de una forma más cómoda e intuitiva, lo que trae consigo una mayor organización y control sobre la información.

El sistema puede ser extendido a las empresas de proyectos de este tipo en la provincia y en el país para su uso pues fue diseñado teniendo en cuenta los estándares para este tipo de empresas.

El SI se implantará de forma tal que el personal se pueda ir adaptando al mismo fácilmente, se confeccionará un manual de usuario detallado donde se expliquen las características, funcionalidades y ventajas reales del sistema y se impartirán cursos para adiestrar al personal que lo utilizará incluyendo en la entrega del SI suficientes materiales de ayuda y documentación.

Dimensión ambiental:

Su implantación favorece en gran medida al cuidado del medio ambiente si tenemos en cuenta el uso de recursos que generan daños al medio ambiente se ve evidenciado en el ahorro de materiales de oficina como el papel que se obtiene a partir de la tala de árboles.

No se hace uso de colores agresivos a la vista en la interfaz del sistema, sino de tonalidades claras que se encuentran en la gama de colores pertenecientes al manual corporativo de la empresa y que el personal está adaptado a su utilización, en busca de un efecto atractivo y amigable, con el objetivo de que los usuarios se identifiquen con el sistema; por lo que existirá una adecuada comunicación entre la aplicación y el usuario disminuyendo los riesgos de estrés psicológico.

Los efectos negativos en la salud de los usuarios son los mismos que los de cualquier sistema informático, con la peculiaridad de que el SI no es un software que necesite mucho tiempo frente a una estación de trabajo, por lo que la vista del usuario y la posición de sentarse no se ven afectadas más de lo normal.

Se puede plantear que este SI no tiene incidencia sobre el deterioro del medio ambiente por lo que es sostenible según esta dimensión.

Dimensión tecnológica:

El análisis y la valoración de la dimensión tecnológica del SI, comprende el uso de la tecnología adecuada y asimilable con el usuario.

La organización cuenta con la infraestructura electrónica necesaria para un correcto funcionamiento del PI, cuenta con una red interna que interconecta todas y cada una de las áreas de la empresa logrando un flujo continuo de las informaciones. Cuenta también con potentes servidores y estaciones de trabajo.

Desde el inicio del diseño e implementación del sistema existió una estrecha comunicación entre los desarrolladores y los usuarios finales, para lograr la mayor satisfacción de estos últimos. El SI permite adaptarse a cambios que no afecten su funcionalidad principal, debido a que está implementado con el empleo de técnicas de programación multicapa y la metodología ICONIX en la ingeniería de software, que permiten que el mantenimiento y los cambios ocurran de forma sencilla.

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

En sentido general, se puede llegar a las siguientes conclusiones que confirman la perdurabilidad del SI en el tiempo:

- El SI desarrollado responde a la necesidad fundamental que lo originó.
- El SI fue desarrollado con la misma filosofía de los demás sistemas que pertenecen a la Intranet de la empresa, para posibilitar su mantenimiento y actualización.
- En la elaboración e implantación del SI no se incurrió en gastos de infraestructura tecnológica, debido a que se utilizaron los mismos sistemas que ya contaba la empresa.

2.12 Valoración de los resultados en la encuesta a los expertos y posibles usuarios.

Una vez concluida la implementación del Sistema de Control de Proyectos propuesto así como las pruebas correspondientes, se aplicaron tres encuestas, la primera

(Ver Anexo 3), se realizó con la finalidad de seleccionar a los expertos y obtener a partir de su procesamiento, el coeficiente de competencia. Para la cual se seleccionaron 15 personas, de los cuales 12 resultaron expertos al tener un coeficiente de competencia alto o medio.

Con el objetivo de evaluar la selección y empleo de las herramientas y tecnologías de desarrollo del sistema propuesto y evaluación de sus funcionalidades, se aplica una segunda encuesta (Ver Anexo 4), a los expertos seleccionados. La tercera (Ver Anexo 5), se realizó para evaluar la satisfacción de los usuarios potenciales respecto al sistema. Las preguntas fueron enfocadas principalmente en los requerimientos no funcionales y otras generalidades para recoger posibles mejoras, y alguna que otra idea que el usuario desee aportar, para esto se escogen 10 personas, 8 posibles usuarios del sistema, y 4 especialistas de otras áreas vinculadas al sistema.

Las encuestas realizadas fueron procesadas mediante el método Delphi (Ver Anexo 6) con el objetivo de buscar el consenso de los encuestados.

A continuación se presentan los aspectos tenidos en cuenta para cada encuesta:

1. Encuesta a expertos:

- La selección de las herramientas y tecnologías de desarrollo.
- El empleo de la metodología de desarrollo.
- Diseño de las interfaces del sistema en cuanto a facilidad y comodidad de uso.
- Correcto cumplimiento de la implantación de los requerimientos del sistema.
- Uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema.

2. Encuesta a posibles usuarios:

- Nivel interacción con el sistema.
- Concepción organizativa del sistema.
- Diseño de las interfaces del sistema.
- Uso correcto de los colores en las interfaces del sistema.

➤ Facilidad y comodidad de uso.

Luego del procesamiento estadístico de los aspectos mencionados anteriormente, se determinó que de 7 aspectos los expertos encuestados coinciden en que 5 son “Muy Relevantes”, y un aspecto “Relevante” lo cual influye positivamente en la calidad del sistema. En la encuesta aplicada a los usuarios de 5 aspectos evaluados, los encuestados consideran que 4 aspectos son “Muy Relevantes” y un aspecto es “Relevante”. Los resultados arrojados demuestran la calidad del módulo informático propuesto en el presente trabajo.

Conclusiones del Capítulo

- El profundo estudio de las distintas tecnologías utilizadas como el cms Liferay, los frameworks Zk, Spring e Hibernate, una herramienta de mapeo objeto-relacional (ORM por sus siglas en ingles) constituyeron una solución factible para el desarrollo de la aplicación propuesta.
- El uso de la metodología ICONIX en la aplicación, ayudó a verificar y modelar en cada una de sus etapas el cumplimiento de los requisitos funcionales del sistema. El refinar y actualizar el modelo del dominio a lo largo del proyecto, permite, reflejar siempre la comprensión actual del problema.
- El análisis de robustez reduce la ambigüedad al describir los casos de uso.
- El análisis de sostenibilidad del producto según las dimensiones administrativa, socio – humanista, ambiental y la tecnológica arrojó que la herramienta propuesta cumple con los requerimientos antes señalados, responde a la necesidad social que la generó, por lo que se puede decir que constituye un producto informático sostenible y perdurable en el tiempo.

Conclusiones generales

Con el desarrollo del módulo informático propuesto, dedicado a favorecer el control de los proyectos de la empresa de Ingeniería y Proyecto del Níquel CEPRONIQUEL de Moa se ha dado cumplimiento al objetivo de esta investigación, pues como resultado se obtuvo un producto informático que facilitará las operaciones y que apoyará en cierta medida la toma de decisiones a los Jefes de Proyectos.

En el sector donde se desarrolla la investigación se encontraron distintos sistemas para el control de los proyectos, que se distribuyen bajo licencias privativas. Por tanto, la creación de un sistema informático para realizar estas operaciones es de gran prioridad. Se propuso como solución el desarrollo de un Módulo Web para el Control de los proyectos en la empresa. A continuación se relacionan las principales conclusiones a las que se arribaron:

- Las tecnologías seleccionadas para el desarrollo del sistema así como la integración de estas fueron de mucha eficacia para la elaboración del mismo.
- Teniendo en cuenta lo novedosas que son las tecnologías utilizadas en la investigación se requirió de un estudio muy profundo por parte del equipo de trabajo para lograr la perfecta integración de las mismas.
- La arquitectura de despliegue empleada resulta una propuesta efectiva para el problema que se aborda.
- Las etapas de la metodología ICONIX permitieron modelar y estructurar el sistema, facilitando representar la correspondencia entre la arquitectura software y la arquitectura hardware.
- Una vez realizada la valoración de sostenibilidad del sistema elaborado, dentro de la cual se analizó la utilización de herramientas y tecnologías de software libre, como una opción viable y superior al software comercial, se puede afirmar que el mismo según su impacto social, económico, tecnológico y ambiental como producto informático es sostenible.
- La combinación adecuada de las tecnologías utilizadas facilitó el desarrollo del Sistema para el Control de Proyectos en la empresa Ceproníquel potenciando la funcionalidad del mismo.

Recomendaciones

A partir de los resultados y las experiencias derivadas de esta investigación se proponen las siguientes recomendaciones:

- Utilizar como referencia este trabajo para la creación de futuros sistemas relacionados con el control de tareas de determinadas índoles.
- Implantar el sistema lo antes posible.

Glosario de términos

Sistema: Conjunto de elementos interrelacionados que trabajan juntos para obtener un resultado deseado.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework 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.

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.

SQL: Structured Query Language (Lenguaje de Consulta Estructurado). Es un lenguaje de acceso a bases de datos.

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado). Es un programa compuesto por un conjunto de herramientas para un programador.

Hardware: Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Plataforma: Una plataforma es una combinación de hardware y software usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado). Es un entorno de programación que ha sido empaquetado como un programa de aplicación.

HTML: Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet, es el conjunto de protocolos que rigen las comunicaciones entre las computadoras en Internet.

HTTP: Hypertext Transport Protocol. Es un conjunto de especificaciones para el intercambio de ficheros (texto, gráfico, imagen, sonido, vídeo) en la Web.

Bibliografía

1. González, E.H., *Manual de la calidad*. 2010.
2. Maynard, H., *Manual de Ingeniería y Organización Industrial*. 2009.
3. Billows, D., *Project Management*. 2002.
4. Web. *La gestión de proyectos*. 2010 [cited 2010 05-2010]; Available from: <http://www.ikaroo.es/ecommerce/1/la-gestion-de-proyectos-y-el-e-commerce-id-1893.htm>.
5. Chiavenato, I., *Fundamentos de Administración*. 2001.
6. Stoner, J., *Administración*. 1993.
7. Salas, K.R. *Gestión de la Información en las Organizaciones*. [cited 2011 02-2011]; Available from: <http://www.una.ac.cr/bibliotecologia/personal/KarlaR.pdf>.
8. Vegas, J. *Introducción a las Aplicaciones Web*. 21/ marzo/2010 [cited 21/enero/2010; Available from: <http://www.infor.uva.es/jvegas/cursos/buendia/pordocente/node11.html>.
9. *¿Que es Internet?* 2010 [cited 2011 05-2011]; Available from: <http://www.quesinternet.net/>.
10. Pujol, M.E.F.i. *Internet, Intranets, Extranets, ¿Son importantes en la empresa?* 1997 [cited 2011 05-2011]; Available from: <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/eulalia.htm>.
11. Libre, F.d.S. *La definición del software Libre*. 2010 [cited 2011 02-2011]; Available from: <http://www.gnu.org/philosophy/free-sw.es.html>.
12. Stallman, R.M., *Software libre para una sociedad libre*. 2004.
13. Álvarez, S., *Características principales de este tipo de arquitectura de cara a base de datos*.
14. Valle, J.G. *Definición arquitectura cliente servidor*. [documento en línea] 2005; Available from: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>
15. Vegas, J. *Introducción a las Aplicaciones Web*. 2011 [cited 2011 04-2011]; Available from: Disponible en: <http://www.infor.uva.es/jvegas/cursos/buendia/pordocente/node11.html>.
16. Torre, A.d.I. *Lenguajes del lado servidor o cliente*. [documento en línea] 2006; Available from: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
17. Almaguer, T.R., *Sistema para la gestión de información de la guardia obrera y estudiantil en la Universidad de Holguín "Oscar Lucero Moya"* 2010, Oscar Lucero Moya.
18. Cheng, R., *Developer's Guide*. 2010.
19. Flanagan, *Java in a Nutshell*. 1997.
20. Horton, I., *Beginning Java™ 2, JDK™ 5 Edition*. 2005, Wiley Publishing, Inc.
21. *Eclipse*. 2005 [cited 2011 10 de febrero]; Available from: http://www.ecured.cu/index.php/Eclipse_entorno_de_desarrollo_integrado.
22. Stäuble, M., *ZK Developer's Guide*. 2008.
23. Rod Johnson, J.H., *Spring Framework*. 2008.
24. Figueras Vigó, Y.E., *Sistema para el control de Unidades Docentes de la Universidad de Holguín*. 2007, Universidad de Holguín: Holguín. Cuba. p. 104.
25. Pupo, E.R., *Sistema de Gestión de Proyectos de Ciencia e Innovación Tecnológica de la Universidad de Holguín "Oscar Lucero Moya"*. 2010.
26. José H. Canós, P.L.y.C.P., *Métodologías Ágiles en el Desarrollo de Software: DSIC* -Universidad Politécnica de Valencia.
27. González, P.C., *Metodología de desarrollo ICONIX*. 2009.
28. Jacobson, I., *El Proceso Unificado de Desarrollo de Software*. 2000.

29. Jaramillo, Á. *Desarrollo de software -Estudio de factibilidad*. 2010 23 de marzo del 2010]; Available from: <http://aldrinjaramillo.googlepages.com/EstudiodeFactibilidad.doc>.
30. Andrade, C.A.C. *Estudio de Factibilidad y Mercadeo*. 2009 29 de marzo del 2010]; Available from: <http://www.cid.uc.edu.ve/fponte/factib.htm>.
31. González, F.R., *Cocoma II Modelo de Estimación de Costes*
32. Patricia, C.R., *Iconix Metodología de desarrollo de Software*. 2002.
33. Doug, M., *Use Case Driven Object Modeling with UML*. Jonathan Gennick ed, ed. 1. 2007, New York,.
34. Concepción, R., *Procedimiento para la valoración de sostenibilidad de un Producto Informático*. 2006.
35. Moya, A., *COCOMO v2 Modelo de estimación de costes para proyectos software* 1999.
36. Vegas, J. *Introducción a las Aplicaciones Web*. Available from: <http://www.infor.uva.es/jvegas/cursos/buendia/pordocente/node11.html>.
37. Valdes, D.A.P. *Los diferentes lenguajes de programación para la web*. 2007 10 de febrero del 2010]; Available from: <http://www.maestrosdelweb.com/>.
38. tec, I. *Nueva Solución de Gestión y Administración de Negocios Agropecuarios*. 2010 28 de junio del 2010]; Available from: <http://www.elsitioagricola.com/ads/software/agro.asp>
39. Sosa, L.A. *Contar con informática*. 2009 10 de febrero 2010]; Available from: http://www.forum.villaclara.cu/UserFiles/File/versat_sarasola.htm.
40. Ramírez, J. *Fundamentos de inventarios*. 2007 2 de enero del 2010]; Available from: <http://www.elprisma.com>.
41. PHYSIS. *PHYSIS GESTIÓN AGRO Nuestro Software Gestión Agropecuaria*. 2010 24 de Marzo del 2010]; Available from: <http://www.physis.com.ar/agro.asp>.
42. Massey, S. *Why MVC?* 2008 26 de marzo del 2010]; Available from: http://www.zkoss.org/smalltalks/zkstudio_vedemo/.
43. Machin, J. *Framework ZK para manejo de Ajax en Java*. 2009 23 de marzo del 2010]; Available from: <http://www.machin.com.mx/archives/1703>.
44. Kon, M.D. *Software Libre*. [documento en línea]; Available from: <http://www.Monografias.com/softwarelibre>.
45. Insoft. *Qué es AgroWin*. 2010 14 de marzo del 2010]; Available from: <http://www.insoftweb.com/agrowin/default.htm>.
46. Hernández, R.A. *Reflexiones sobre el uso del Software Libre en Cuba. Ventajas*. [documento en línea] 2006; Available from: <http://www.monografias.com/trabajos40/software-cuba/software-cuba.shtml?monosearch>.
47. González, C.D. *Curso Base de Datos PostgreSQL, SQL avanzado y PHP* [documento en línea] 2010; Available from: <http://www.usabilidadweb.com.ar/postgre.php>
48. Gonzales, M. *SQL SERVER*. [documento en línea]; Available from: <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>.
49. Garrett, J.J. *Ajax: Un nuevo acercamiento a las aplicaciones web*. 2010 11 de marzo del 2010]; Available from: <http://www.maestrosdelweb.com/editorial/ajax/>.
50. García, A. *Guía del autoestopista a Hibernate*. 2003 26 de marzo del 2010]; Available from: www.javahispano.org.
51. FullTrust. *FullTrust Edición Web o FullTrust-W*. 2010 29 de marzo del 2010]; Available from: http://www.fulltrust.net/ftw_es.php.
52. Fuenzalida, D. *Eclipse*. 2008 julio del 2008 26 de febrero del 2010]; Available from: <http://manzanamecanica.org/2008/07/eclipse.html>.
53. Falcón, V.V. *La contabilidad y su evolución en Cuba*. 2009 5 de marzo del 2010]; Available from: <http://www.monografias.com/trabajos14/contacuba/contacuba.shtml>.

54. Engel Bustamante. *Inventario*. 2010 10 de mayo del 2010]; Available from: <http://www.monografias.com/trabajos11/conin/conin2.shtml>.
55. DISAIC. *El VERSAT-Sarasola: Sistema cubano de Gestión Contable-Financiero* 2010 21 de febrero del 2010]; Available from: <http://www.disaic.cu/modules.php?name=Content&pa=showpage&pid=818>.
56. Deluca, S.M. *¿Qué es ASSET NS?* 2010 15 de marzo del 2010]; Available from: <http://assets.co.cu/assets.asp>.
57. Delgado, E. *Metodologías de desarrollo de software. ¿Cuál es el camino?* . [documento en línea]; Available from: <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software.shtml>.
58. CITMATEL. *Rodas XXI: un producto cubano para la empresa cubana*. 2010 15 de febrero del 2010]; Available from: <http://www.rodasxxi.cu/>.
59. Chen, H. *Hibernate + ZK*. 2006 21 de febrero del 2010].
60. Anaya, A. *Desarrollo de software bajo metodologías ágiles (Agile Methods XP) en la práctica*. [documento en línea]; Available from: <http://www.monografias.com/trabajos48/desarrollo-software-agil/desarrollo-software-agil.shtml>.
61. Amartino, M. *Ajax: Un nuevo acercamiento a las aplicaciones Web*. [documento en línea] 2005; Available from: <http://www.maestrosdelweb.com/editorial/ajax/>.
62. Álvarez, S. *Sistemas gestores de bases de datos*. 2007; Available from: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
63. Alvarez, M.A. *Qué es JSP*. [documento en línea] 2002; Available from: <http://www.desarrolloweb.com/articulos/831.php>.
64. Álvarez, G. *Características del lenguaje Java*. 1999 19 Marzo 2010]; Available from: <http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html#top>.
65. *Características de PostgreSQL* 2006 23 Marzo 2010]; Available from: <http://www.manualesdeayuda.com/manuales/bases-de-datos/postgresql/caracteristicas-de-postgresql-01844.html>.

Anexos

Anexo 1. Estándar de código generado para el proyecto SistContPro

Historial de versiones del documento:

Fecha	Versión	Descripción	Autor
30/10/2010	1.0	Estándar Yaser Morera Rivera	Yaser

Organización del código

Aspectos generales

1. Idioma:

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

2. Identación:

La indentación va ser fija para todos elementos a 6 espacios.

3. Anidamiento:

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

4. Tamaño máximo de líneas:

Las líneas de código no deben exceder los 80 caracteres. Por su parte, las de ruptura (continuación de una línea de código que excedió los 80 caracteres) no deben exceder a 35 caracteres.

5. Módulos:

Los módulos no deben contener más de 150 líneas de código.

6. Apertura y cierre de ámbito ([9], etc):

Los ámbitos van a ser abiertos y cerrados en una línea aparte a la sentencia que los precede.

Líneas y espacios en blanco

7. 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.

8. 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 SistContPro. 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 contener letras, y la longitud de este no debe exceder los 20 caracteres.

Nombre para Unidades: Indicar propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Tipos: 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 10 caracteres.

Nombre para Interfaces : Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Clases: Los nombres de las clases deben indicar propósito, que contener letras, y la longitud de este no debe exceder los 10 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 Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Métodos: Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Constructores: Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 10 caracteres.

Nombre para Parámetros: 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 10 caracteres.

Nombre para Excepciones y sus objetos: Notación Camell. 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 10 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 10 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 10 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.

En 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:

Definir un constructor copia.

No redeclarar parámetros para que contenga un valor por defecto.

Definir constructores virtuales para súper clases.

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.

Evitar paso de parámetros por valor.

Usar parámetros por defecto en lugar de sobrecarga de funciones.

Aspectos de complejidad: Tiene que ser menor que 10. Mientras que la complejidad ciclomática extendida que 15. El número máximo de líneas que puede contener el cuerpo de una función es de 60, 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 de los demás módulos. 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 10 caracteres.

Nombre para Procedimientos: llevarán el prefijo pa, 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 10 caracteres.

Nombre para Triggers: No requieren prefijo. 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 10 caracteres.

Anexo 2. Cálculo de esfuerzo de desarrollo. COCOMO 2

A partir de los requerimientos funcionales, se identificaron las características (entradas externas, salidas externas, etc.) con el objetivo de calcular los puntos de función desajustados (UFP, por sus siglas en inglés) se clasificaron y se realizó la ponderación; A continuación se presentan las características del sistema representadas mediante diferentes tablas.

Entradas Externas (EI): son todas aquellas entradas que le son proporcionadas al sistema.

Nombre	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Insertar	3	5	Alto
modificar	3	5	Alto

Salidas externas (EO). Salidas asociadas al sistema que tiene elementos de filtraje de información.

Nombre	Cantidad de Ficheros	Cantidad de elementos De datos	Clasificación
Listar	3	10	Medio
Listar	3	16	Alto
Listar	1	4	Bajo
Listar	1	4	Bajo
Listar	2	10	Medio
Listar	1	10	Bajo
Listar	1	10	Bajo

Peticiones (EQ)

Nombre de las peticiones	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Listar	1	3	Bajo

Ficheros lógicos internos (ILF): son los grupos de datos relacionados lógicamente que permiten el almacenamiento de la información perteneciente al sistema.

Nombre	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
tb_asigancion_personal_tarea	1	5	Bajo
tb_asignacion_tarea	1	4	Bajo
tb_categoria	1	4	Bajo
tb_especialidad_tareas	1	2	Bajo
tb_formato_hojas	1	3	Bajo
tb_plan_disenno	1	2	Bajo
tb_proyecto	1	8	Bajo
tb_proyecto_estado	1	6	Bajo
tb_proyecto_servicios	1	2	Bajo
tb_servicios	1	2	Bajo
tb_servicios_especialidades	1	3	Bajo
tb_tarea_estado	1	3	Bajo
tb_tareas	1	6	Bajo
tb_tareas_planos	1	3	Bajo
organization	1	5	Bajo
role	1	4	Bajo
user	1	3	Bajo

Puntos de función desajustados.

Elementos	Bajos		Medios			Altos	Subtotal de puntos de función
	No.	Peso	No.	Peso	No.	Peso	
Entradas externas (EI)	12	3	6	4	6	6	96
Salidas externas (EO)	7	4	2	5	6	7	80
Ficheros lógicos internos(ILF)	17	7	0	10	0	15	119
Peticiones EQ	11	20		0	0	0	20
	1						
Total							315

El cálculo de las instrucciones fuentes (SLOC), COCOMO II se basa en la cantidad de instrucciones por punto de función que genera el lenguaje de programación empleado. Utilizando la siguiente fórmula se obtiene:

$$\text{SLOC} = \text{UFP} * \text{ratio}$$

Características	
Características	Valor
Puntos de función desajustados	327
Lenguaje	Java 80 %
	SGLM 15 %
	SQL 5 %
Ratio del lenguaje	Java 55

	SGML 58
	SQL 31
Instrucciones * lenguaje	Java 13 860
	SGML 2740
	SQL 488
Instrucciones fuentes (SLOC, en líneas de código)	17,08
Reducción de un 30 % (Reutilización de código)	11,96

Cálculo de esfuerzo, tiempo, cantidad de hombres y costo

A continuación se obtendrán valores necesarios para el cálculo del esfuerzo:

Multiplicadores de esfuerzo

Los multiplicadores de esfuerzo (ME), que se muestran a continuación representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software. Para determinar el esfuerzo asociado al desarrollo del sistema, denominado PM, se utilizan los multiplicadores de esfuerzo, los factores de escala, así como los valores constantes A, B, C, D.

Multiplicadores	Descripción	Valor
RELY	Garantía de funcionamiento requerida al software.	2

DATA	Tamaño de la base de datos.	3
CPLX	Complejidad del producto.	5
DOCU	Cantidad de artefactos que deben ser documentados.	5
RUSE	Desarrollo para ser reutilizado.	4
TIME	Exigencias sobre capacidad de ejecución.	5
STOR	Almacenamiento.	3
PVOL	Volatilidad de la plataforma.	5
ACAP	Capacidad de los analistas.	9
PCAP	Capacidad de los programadores.	3
PCON	Continuidad del personal.	2
APEX	Experiencia previa de los analistas.	2
PLEX	Experiencia con la plataforma.	3
LTEX	Experiencia previa con el lenguaje y herramientas de desarrollo.	3
ITOO	Uso de herramientas de software.	3
SITE	Desarrollo en localidades distribuidas.	5
SCED	Exigencias sobre el calendario.	4

Factores de Escala

Factores de escala	Descripción	Valor
PREC	Precedencia.	3,72
FLEX	Flexibilidad	4,05
RESL	Riesgos	4,24
TEAM	Cohesión del equipo	0

PMAT	Madurez de las capacidades	4,68
------	----------------------------	------

Valores calibrados

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

Anexo 3. Encuesta para la determinación del coeficiente de competencias de expertos

Nombre y apellidos: _____.

Cargo que desempeña: _____.

Usted ha sido seleccionado como posible experto para ser consultado respecto al grado de utilidad del Sistema de Control de Proyectos de la empresa CEPRONIQUEL de Moa.

Necesitamos, antes de realizarle la consulta correspondiente, como parte del método empírico de investigación "consulta a expertos", determinar su coeficiente de competencia en este tema, a los efectos de reforzar la validez del resultado de la consulta que realizaremos. Se necesita 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 le presentamos 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 y ha tenido en su conocimiento y criterio sobre los indicadores puestos 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 (alto)	M (medio)	B (bajo)
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.			

Muchas gracias.

Anexo 4. Encuesta aplicada a expertos

Estimado compañero(a), con motivo de evaluar el Sistema de Control de Proyectos de la empresa CEPRONIQUEL de Moa, desarrollado recientemente, se aplica esta encuesta. Por lo que la información que brinde será crucial para estos objetivos; rogamos que al responder estas preguntas lo haga de la manera más explícita posible. De antemano gracias.

1. A continuación sometemos a su valoración una serie de elementos que se consideran importantes en la evaluación del sistema. Para esta evaluación se le propone las siguientes categorías:

Muy Relevante (MR) Bastante Relevante (BR) Relevante (R)

Poco Relevante (PR) No Relevante (NR)

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

Pregunta 1

Criterios	M R	BR	R	PR	NR
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?					
¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?					
¿Cómo aprecia el diseño de las interfaces del sistema?					
¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del sistema?					
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?					
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?					

Si quisiera opinar alguna u otra idea que no se haya abordado en esta encuesta, por favor, exprésela a continuación.

Gracias por su amable opinión.

Anexo 5. Encuesta aplicada a los usuarios

Estimado compañero(a), con motivo de evaluar el Sistema de Control de Proyectos de la empresa CEPRONÍQUEL de Moa, desarrollado recientemente, se aplica esta encuesta. Por lo que la información que brinde será crucial para estos objetivos; rogamos que al responder estas preguntas lo haga de la manera más explícita posible. De antemano gracias.

1. A continuación sometemos a su valoración una serie de elementos que se consideran importantes en la evaluación del sistema.

Para esta evaluación se le propone las siguientes categorías:

Muy Relevante (MR) Bastante Relevante (BR) Relevante (R)

Poco Relevante (PR) No Relevante (NR)

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

Pregunta 1

Criterios	MR	BR	R	PR	NR
¿Cómo evalúa la concepción organizativa del sistema?					
¿Cómo evalúa diseño de las interfaces del sistema?					
¿Cuánto contribuirá el sistema a la toma de decisiones?					
¿Cómo aprecia Nivel interacción con el sistema?					
¿Cree correcto el uso de los colores en las interfaces?					

2. ¿Qué otros elementos novedosos encuentra en la aplicación?

3. Si quisiera opinar alguna u otra idea que no se haya abordado en esta encuesta, por favor, exprésela a continuación.

Gracias por su amable opinión.

Anexo 6. Procesamiento de la encuesta de opinión de los expertos aplicando el método Delphi.

TABLA DE FRECUENCIA ABSOLUTA						
Fases y Momentos	MR	BR	R	PR	NR	TOTAL
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?	12	3	4	1	0	20

¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?	13	4	2	1	0	20
¿Cómo aprecia el diseño de las interfaces del sistema?	15	2	3	0	0	20
¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del sistema?	12	5	3	0	0	20
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?	15	5	0	0	0	20
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?	14	3	3	0	0	20
¿Posee el módulo un nivel apropiado de interacción con el usuario?	16	2	2	0	0	20
¿Cómo evalúa la concepción organizativa del sistema?	8	6	6	0	0	20
¿Cómo evalúa diseño de las interfaces del sistema?	10	7	3	0	0	20
¿Cuánto contribuirá el sistema a la toma de decisiones?	12	8	0	0	0	20
¿Cómo aprecia Nivel interacción con el sistema?	12	3	4	1	0	20
¿Cree correcto el uso de los colores en las interfaces?	15	3	2	0	0	20

TABLA DE FRECUENCIA ABSOLUTA ACUMULADA					
Fases y Momentos	MR	BR	R	PR	NR
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?	12	15	19	20	20
¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?	13	17	19	20	20
¿Cómo aprecia el diseño de las interfaces del sistema?	15	17	20	20	20
¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del	12	17	20	20	20

sistema?					
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?	15	20	20	20	20
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?	14	17	20	20	20
¿Posee el módulo un nivel apropiado de interacción con el usuario?	16	18	20	20	20
¿Cómo evalúa la concepción organizativa del sistema?	8	14	20	20	20
¿Cómo evalúa diseño de las interfaces del sistema?	10	17	20	20	20
¿Cuánto contribuirá el sistema a la toma de decisiones?	12	20	20	20	20
¿Cómo aprecia Nivel interacción con el sistema?	12	15	19	20	20
¿Cree correcto el uso de los colores en las interfaces?	15	18	20	20	20

TABLA DEL INVERSO DE LA FRECUENCIA ABSOLUTA ACUNULADA				
Fases y Momentos	MR	BR	R	PR
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?	0,6	0,75	0,95	1
¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?	0,65	0,85	0,95	1
¿Cómo aprecia el diseño de las interfaces del sistema?	0,75	0,85	1	1

¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del sistema?	0,6	0,85	1	1
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?	0,75	1	1	1
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?	0,7	0,85	1	1
¿Posee el módulo un nivel apropiado de interacción con el usuario?	0,8	0,9	1	1
¿Cómo evalúa la concepción organizativa del sistema?	0,4	0,7	1	1
¿Cómo evalúa diseño de las interfaces del sistema?	0,5	0,85	1	1
¿Cuánto contribuirá el sistema a la toma de decisiones?	0,6	1	1	1
¿Cómo aprecia Nivel interacción con el sistema?	0,6	0,75	0,95	1
¿Cree correcto el uso de los colores en las interfaces?	0,75	0,9	1	1

TABLA DE DETERMINACIÓN DE LOS PUNTOS DE CORTES							
Fases y Momentos	MR	BR	R	PR	Suma	Prom.	N-Prom.
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?	0,25	0,67	1,64	3,49	6,05	1,51	0,56
¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?	0,39	1,04	1,64	3,49	6,56	1,64	0,43
¿Cómo aprecia el diseño de las interfaces del sistema?	0,67	1,04	3,49	3,49	8,69	2,17	-0,1
¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del SI?	0,25	1,04	3,49	3,49	8,27	2,07	0
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?	0,67	3,49	3,49	3,49	11,14	2,79	-0,72
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?	0,52	1,04	3,49	3,49	8,54	2,14	-0,07
¿Posee el módulo un nivel apropiado de interacción con el usuario?	0,84	1,28	3,49	3,49	9,1	2,28	-0,21
¿Cómo evalúa la concepción organizativa del sistema?	- 0,25	0,52	3,49	3,49	7,25	1,81	0,26
¿Cómo evalúa diseño de las interfaces del sistema?	0	1,04	3,49	3,49	8,02	2,01	0,06
¿Cuánto contribuirá el sistema a la toma de decisiones?	0,25	3,49	3,49	3,49	10,72	2,68	-0,61
¿Cómo aprecia Nivel interacción con el sistema?	0,25	0,67	1,64	3,49	6,05	1,51	0,56
¿Cree correcto el uso de los colores en las interfaces?	0,67	1,28	3,49	3,49	8,93	2,23	-0,16

CONCLUSIONES GENERALES					
Fases y Momentos	MR	BR	R	PR	NR
¿Cómo evalúa la combinación de las tecnologías y herramientas de desarrollo?	-	SI	-	-	-
¿Cómo evalúa el empleo de la metodología de desarrollo ICONIX?	-	SI	-	-	-
¿Cómo aprecia el diseño de las interfaces del sistema?	Si	-	-	-	-
¿Cómo evalúa el cumplimiento de la implantación de los requerimientos del sistema?	Si	-	-	-	-
¿Cómo evalúa el uso correcto de la gama de colores que identifican a la empresa en las interfaces del sistema?	Si	-	-	-	-
¿Cómo evalúa en cuanto a facilidad y comodidad el módulo?	Si	-	-	-	-
¿Posee el módulo un nivel apropiado de interacción con el usuario?	Si	-	-	-	-
¿Cómo evalúa la concepción organizativa del sistema?	Si	-	-	-	-
¿Cómo evalúa diseño de las interfaces del sistema?	Si	-	-	-	-
¿Cuánto contribuirá el sistema a la toma de decisiones?	Si	-	-	-	-
¿Cómo aprecia Nivel interacción con el sistema?	-	SI	-	-	-
¿Cree correcto el uso de los colores en las interfaces?					