

Universidad de Holguín
“Oscar Lucero Moya”
Facultad de Informática y Matemática

**Mantenimiento de software del
módulo de seguimiento y control
a proyectos del portal
Ceproníquel**

Trabajo de diploma en opción al título de
Ingeniero Informático

Autor: Arnaldo Amaro Quiñones

Tutor: Ing. Mario Aballe Rodríguez

Holguín 2014

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Universidad de Holguín "Oscar Lucero Moya" para que hagan el uso que estimen pertinente con este trabajo.

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

Nombre completo del autor

Nombre completo del tutor

Nombre completo del tutor

Si buscas resultados distintos, no hagas siempre lo mismo.

Albert Einstein

Dedicatoria

A mis dos queridos hijos que siempre están en mi corazón: Arnaldito y Elizabeth.

A mi familia por todo su amor y educación.

A mi gran amigo Raúl por todo el apoyo que me brindó durante la carrera en los momentos más difíciles.

A mí, por la constancia y el esfuerzo durante el transcurso de la carrera.

Agradecimientos

A mis padres por su apoyo incondicional y por quererme tanto, por darme la vida y sobre todo porque lo son todo para mí.

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

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

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

A mis amistades por su preocupación y su ayuda.

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

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

A todos muchas gracias.

Resumen

En Cuba son varias las empresas que usan la informática en beneficio de sus procesos, entre las que se encuentra Ceproníquel, una organización dedicada al suministro de servicios de ingeniería, diseño y consultoría, y gestión de proyectos. Dentro de los procesos que esta empresa incorpora está el proceso de seguimiento y control a proyectos, que ha sido, desde sus inicios, una tarea compleja y de importancia en la búsqueda de eficiencia. En el año 2013 se realizó un estudio sobre el flujo de la información concerniente al proceso de seguimiento y control a proyectos. Como resultado se obtuvo una aplicación web para dar solución a las dificultades encontradas en este proceso, la cual tiene un alto grado de integración con el portal empresarial Ceproníquel, sin embargo carece de funcionalidades de importancia para la dirección de la empresa. Lo antes mencionado y el surgimiento de nuevos requisitos, conllevan a la necesidad de realizar el proceso de mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel. Para implementar este proceso, se utilizan las siguientes herramientas y tecnologías: Liferay Portal, *framework* ZK, IDE Eclipse y PostgreSQL como sistema gestor de bases de datos.

Abstract

In Cuba there are several companies using informatics for the benefit of its processes, among which is Ceproníquel, an organization that provides services of engineering, design and consulting, and project management. Among the processes within the company, is the control and monitoring process, which always has been of importance to achieve efficiency. In 2013, a study of this process workflow in the company was made, resulting a web application to resolve the difficulties found, however, it doesn't contain all functionalities of importance the company. The mentioned above and the capture of new requirements, make necessary the maintenance process to this software product. In order to do so, the following tools will be used: ZK framework, Liferay Portal, IDE Eclipse, and PostgreSQL as Databases Management System.

Índice

Resumen	6
Índice	8
Índices de Figuras	11
Índice de Tablas	12
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS	7
1.1.Seguimiento y control a proyectos	7
1.1.1.Seguimiento y control a proyectos en Ceproníquel	8
Figura 2 Estructura organizativa de la empresa Ceproníquel	8
1.2.Módulo de seguimiento y control a proyectos del portal Ceproníquel.....	10
1.3.Mantenimiento de software	12
1.3.1.Tipos de Mantenimiento	13
1.3.2.Actividades y problemas de mantenimiento de software.....	14
1.4.Herramientas y tecnologías	15
1.4.1.Aplicación web	15
1.4.2.Liferay Portal	16
1.4.3.Framework ZK.....	20
1.4.4.Entorno de Desarrollo Integrado	21
1.4.5.Sistemas Gestores de Bases de Datos.....	22
1.4.6.Metodologías de desarrollo de software.....	23
1.4.6.1.ICONIX.....	24
Conclusiones del capítulo	27
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	28
2.1.Mantenimiento correctivo.....	28
2.2.Mantenimiento preventivo	28

2.3.Mantenimiento perfecto.....	29
2.3.1Análisis de requisitos.....	30
Tabla 2 Definición de los principales conceptos del modelo del dominio	31
Figura 14 Diagrama de Caso de Uso del sistema	34
Tabla 3 Actores del sistema.....	34
2.3.2Análisis y diseño preliminar	35
Tabla 4 Caso de uso: Registrar solicitud.....	35
Tabla 5 Caso de uso: Realizar reporte de plan de diseño	36
Figura 15 Diagrama de robustez. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.....	37
2.3.3Figura 17 Diagrama del Modelo de Despliegue Diseño detallado.....	40
Figura 18 Diagrama Secuencia. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.....	42
2.3.4Implementación	43
2.3.Valoración de sostenibilidad	45
2.3.1.Dimensión administrativa	45
Tabla 6 Factor de Peso de los Actores sin ajustar	46
Tabla 7 Factor de Peso de los Casos de Uso sin ajustar	47
Tabla 8 Factores de complejidad técnica	47
Tabla 9 Factores de ambiente	48
2.3.2.Dimensión Socio-Humanista	49
2.3.3.Dimensión ambiental.....	51
2.3.4.Dimensión tecnológica	51
2.4.Valoración de los resultados en la encuesta a los expertos y posibles usuarios	52
2.5.Conclusiones del capítulo	53
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES	55
REFERENCIAS BIBLIOGRÁFICAS	56

ANEXOS 57

Índices	de	Figuras
Figura 1	Fases del ciclo de vida de un proyecto	7
Figura 2	Estructura organizativa de la empresa Ceproníquel.....	8
Figura 3	Flujo de información	8
Figura 4	Diagrama de paquetes del módulo.....	11
Figura 5	Modelo de despliegue del módulo.....	12
Figura 6	Origen de los defectos del software	13
Figura 7	Cuadrante Mágico de Gartner para portales horizontales.....	17
Figura 8	Arquitectura de Liferay.	19
Figura 9	Arquitectura de Liferay Service Builder	20
Figura 10	Arquitectura de ZK	21
Figura 11	Modelo de entidades del portal Ceproníquel.....	29
Figura 12	Modelo del dominio	32
Figura 13	Diagrama de paquetes	33
Figura 14	Diagrama de Caso de Uso del sistema.....	34
Figura 15	Diagrama de robustez. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.....	37
Figura 16	Diagrama de robustez. Paquete: Seguimiento y control. Caso de uso: Importar proyecto en mpp.....	37
Figura 17	Diagrama del Modelo de Despliegue Diseño detallado.....	40
Figura 18	Diagrama Secuencia. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.....	42
Figura 19	Diagrama Secuencia. Paquete: Seguimiento y control. Caso de uso: Importar proyecto en mpp.....	42

Índice	de	Tablas
Tabla 1	Importancia de las actividades de mantenimiento	14
Tabla 2	Definición de los principales conceptos del modelo del dominio	31
Tabla 3	Actores del sistema	34
Tabla 4	Caso de uso: Registrar solicitud	35
Tabla 5	Caso de uso: Realizar reporte de plan de diseño	36
Tabla 6	Factor de Peso de los Actores sin ajustar	46
Tabla 7	Factor de Peso de los Casos de Uso sin ajustar	47
Tabla 8	Factores de complejidad técnica	47
Tabla 9	Factores de ambiente	48

INTRODUCCIÓN

En Cuba, con el desarrollo de las nuevas tecnologías, son varias las empresas que se suman al grupo de las que emplean la Informática para beneficio de sus procesos, sin embargo, son escasas las que poseen una fuerte infraestructura para explotar al máximo la gama de posibilidades que ofrece.

Una de las que está llevando a cabo la informatización de sus procesos es la Empresa de Ingeniería y Proyectos del Níquel (Ceproníquel), ubicada en carretera Moa-Sagua Km 1 ½ Moa, Holguín, Cuba. Esta entidad fue creada en 1985 y juega un importante papel en la industria cubana del níquel, pues es encargada de prestar servicios de Ingeniería, Consultoría y Dirección Integrada de Proyectos al Grupo Empresarial Cuba níquel, así como a la comunidad.

Su misión es ayudar a los clientes a encontrar las mejores soluciones en cuanto a las necesidades de Ingeniería, Diseño y Desarrollo y Gestión de Proyectos en la Industria Minero-Metalúrgica, proporcionando un desarrollo sustentable, garantizando la salud y seguridad del personal y otras partes interesadas. Mientras que su visión se basa en ser Empresa Líder en los Servicios de Ingeniería, Diseño y Desarrollo y Gestión de Proyectos en la Industria Minero-Metalúrgica y otros, proporcionando un desarrollo sustentable, garantizando la salud y seguridad del personal y otras partes interesadas .

Dentro de sus principales prestaciones se encuentran: Servicios Técnicos de Ingeniería Conceptual, Básica y de Detalles, Dirección de la Construcción y Dirección Integrada de Proyectos. Cuenta con una producción anual alta y por ende sus ingresos al país son de gran importancia, en la obtención de estos resultados interviene un personal técnico-directivo joven, con elevada preparación en la actividad de ingeniería.

Esta joven institución ha estado involucrada en proyectos de gran envergadura, como son la Modernización de la Empresa Ernesto Che Guevara, Proyecto de Urbanización Ciudad Moa, Transportador de Mineral Pinares de Mayarí y Montaje de Calderas de 90 toneladas en la planta René Ramos Latour, entre otros.

Ha tenido asociaciones económicas internacionales con las firmas de ingeniería canadienses *Kvaerner Metals* y *Knight Piesold*, ambas reconocidas a nivel Mundial, 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 utilizando tecnología *Siemens*.

Dentro de los procesos que esta empresa incorpora está el proceso de seguimiento y control a proyectos, que ha sido, desde sus inicios, una tarea compleja y de importancia en la búsqueda de eficiencia¹.

Las actividades de control y seguimiento de la ejecución de los proyectos están encaminadas a conocer, en períodos regulares preestablecidos, el comportamiento de importantes parámetros como el tiempo y la utilización de recursos, lo que posteriormente definirá las bases para la toma de decisiones .

Al estudiar y analizar la realización del proceso de control y seguimiento a proyectos en la empresa se detectaron las siguientes deficiencias:

- La información referente a los proyectos está almacenada en ficheros con formato de Microsoft Project, lo que dificulta su extracción, consulta y análisis. Esto incluye la falta de trazabilidad en control de cambios de las tareas planificadas, por lo que pueden ocurrir errores durante este proceso.
- Toda la documentación referente a los proyectos está almacenada en una carpeta contenedora ubicada en un servidor al cual tienen acceso todos los usuarios involucrados en los proyectos. Esto permite al personal no autorizado la realización de cambios en los documentos, comprometiendo la integridad de la información.

1

- La comunicación del personal solo se realiza por vía telefónica o correo electrónico lo que alarga el tiempo de realización de las tareas de control y seguimiento, retrasando así la entrega de la propuesta al cliente.
- No se tiene una forma eficiente de brindar información a la dirección sobre el proceso de control y seguimiento a proyectos.

En el año 2013 se realizó un trabajo de diploma para optar por el título de Ingeniero Informático, en el que se hizo un estudio sobre el flujo de la información concerniente al proceso de seguimiento y control a proyectos. Como resultado se obtuvo una aplicación web para dar solución a las dificultades antes expuestas, la cual tiene un alto grado de integración con el portal empresarial Ceproníquel, sin embargo carece de las siguientes funcionalidades:

- Mostrar a los especialistas principales la carga de trabajo del personal involucrado en los proyectos, funcionalidad que garantizaría una planificación realista y un mejor control y seguimiento en las tareas programadas.
- Importar a la plataforma proyectos con formato de Microsoft Project, para facilitar el trabajo y disminuir la resistencia al cambio por parte del cliente.
- Conocer el estado general actual del proyecto, brindando a los jefes de proyecto una herramienta para tomar decisiones.

Y contiene las siguientes deficiencias:

- La realización de reportes es defectuosa.
- No se manipulan las conexiones a la base de datos (BD) de forma correcta, lo que provoca que el servidor de BD colapse luego de un tiempo de uso de la aplicación.
- El diseño de algunos elementos visuales dificulta el acceso de forma intuitiva a la información por parte de los usuarios.
- No se brindan recursos gráficos como diagramas de Gantt, para visualizar las tareas en cada proyecto. Esto dificulta en cierta medida la interacción de los usuarios con el sistema.

Los elementos mencionados, y el surgimiento de nuevos requisitos, conllevan al siguiente **problema científico**: ¿Cómo favorecer la gestión de información del módulo de seguimiento y control a proyectos del portal Ceproníquel?

El anterior problema científico tiene como **objeto de estudio** el módulo de seguimiento y control a proyectos del portal Ceproníquel.

Para solucionar el problema científico se persigue el siguiente **objetivo**: realizar el proceso de mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel sobre la base de la corrección de errores y la extensión de funcionalidades.

Donde el **campo de acción** es el mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel.

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

- ¿Cuáles son los fundamentos teóricos que sustentan el proceso de seguimiento y control a proyectos de la empresa Ceproníquel?
- ¿Cuál es la situación actual de la gestión del módulo de seguimiento y control a proyectos del portal Ceproníquel?
- ¿Cómo desarrollar el mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel?
- ¿Será sostenible el producto informático resultante del proceso de mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel?
- ¿Cómo valorar el grado de aceptación del producto informático propuesto?

Para lograr el objetivo de la investigación se plantean las siguientes **tareas**:

1. Estudiar la fundamentación teórica que sustenta el proceso de seguimiento y control a proyectos de la empresa Ceproníquel.
2. Determinar el estado actual de la gestión del módulo de seguimiento y control a proyectos del portal Ceproníquel.
3. Realizar el mantenimiento de software al módulo de seguimiento y control a proyectos del portal Ceproníquel.

4. Valorar la sostenibilidad del producto informático.
5. Valorar la solución propuesta mediante el criterio de expertos.

Métodos teóricos:

Análisis y síntesis: Se utilizó en la elaboración de los fundamentos teóricos, en el procesamiento de la información y en la descomposición de cada uno de los requerimientos del sistema, así como del mismo sistema en pequeños subsistemas para facilitar la comprensión del problema con más detalle. Otra aplicación de este método de investigación ocurrió en la valoración de sostenibilidad del producto.

Histórico y lógico: La aplicación de este método teórico se puso en práctica durante el estudio de las herramientas utilizadas, así como en el análisis de los sistemas utilizados en la empresa anteriormente con este fin.

Método de modelación: Este método fue utilizado durante casi toda la etapa de elaboración del sistema, desde su análisis, hasta su implementación; pues se utilizó la metodología de ingeniería de software ICONIX, la cual utiliza el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés), lográndose con esto que el conjunto de modelos de la metodología describiera todas las perspectivas posibles del proceso de desarrollo en sentido general, y que el sistema se pudiese modelar de una forma menos abstracta.

Enfoque Sistémico: Fue utilizado para identificar y descomponer el sistema en subsistemas, así como las relaciones entre ellos, facilitando además organizar el trabajo y la lógica del negocio identificada.

Métodos empíricos:

Observación científica: Se empleó para llevar a cabo la implantación y diagnóstico del resultado obtenido de esta investigación. Se acompaña de procedimientos y técnicas propias para las etapas de desarrollo, lo que permitió que se tuviera una mayor precisión y seguridad en las decisiones tomadas y los resultados obtenidos acerca del comportamiento del objeto de investigación, tal y como éste se comporta en la realidad, además de obtener la información directa e inmediata sobre el proceso de desarrollo y objeto que está siendo investigado.

Entrevista: Debido a que en todo proceso de desarrollo de un sistema informático el usuario final es un actor más, se tuvieron en cuenta las sugerencias, criterios y necesidades que pudieran influir en la correcta concepción del sistema. De ahí que se aplicara este método para obtener información, búsqueda de opiniones y conocimientos de expertos; además de recopilar elementos para el análisis del sistema.

Revisión de documentos: Este método se utilizó fundamentalmente para entender y recopilar los requerimientos funcionales del sistema.

Criterio de expertos: Se utilizó para llegar a un consenso entre los expertos valorando así el sistema informático resultante, para procesar estas encuestas se usó el método Delphi.

El presente documento está estructurado en dos capítulos:

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

En el Capítulo 2 (Descripción de la solución propuesta) se muestran los flujos de trabajo principales de la ingeniería de software realizada al sistema propuesto, se incluye un estudio de factibilidad y una valoración de sostenibilidad del sistema como producto informático.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

En este capítulo se exponen los fundamentos teóricos que sustentan la presente investigación; se hace un análisis de las principales categorías relacionadas con el objeto de estudio, para entender con mayor claridad el proceso de seguimiento y control a proyectos en Ceproníquel. Se presenta un estudio de las tecnologías y herramientas utilizadas en la solución propuesta y finalmente se hace un análisis de la metodología de desarrollo de software utilizada.

1.1. Seguimiento y control a proyectos

Un proyecto es un proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme con requisitos específicos, incluyendo las limitaciones de tiempo, costo y recursos . Pasa por diferentes etapas, reconocidas generalmente como: conceptualización, diseño y planificación, control de la ejecución y cierre o declinación.

Como puede apreciarse en la Figura 1, la etapa de control es la que más se extiende en el ciclo, debido a que aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, la alta dirección 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 planificados.

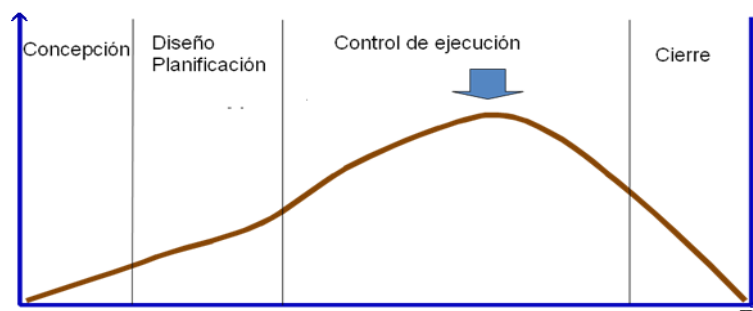


Figura 1 Fases del ciclo de vida de un proyecto

En esta etapa se 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.

1.1.1. Seguimiento y control a proyectos en Ceproníquel

Para una mejor comprensión del proceso de seguimiento y control a proyectos que se realiza en la empresa Ceproníquel es necesario conocer su estructura como se muestra en la Figura 2:



Figura 2 Estructura organizativa de la empresa Ceproníquel

Según la figura anterior, la empresa está conformada por un Consejo Directivo y diferentes Divisiones, las cuales intervienen en la realización de los proyectos. Los jefes de proyecto son los encargados de realizar las actividades de control y seguimiento para garantizar el cumplimiento de las tareas planificadas dentro de los proyectos. La muestra el rol que juega el jefe de proyecto en el flujo de información entre el cliente y la empresa Ceproníquel.

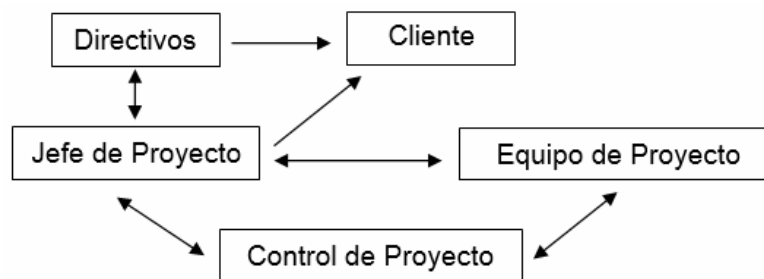


Figura 3 Flujo de información

El proceso de control y seguimiento a proyectos en la Ceproníquel es el que da seguimiento al proceso de planificación, constituyendo uno de los más importantes en la entidad, puesto que pertenece al área de producción. Esta área resalta por encima de otras, debido fundamentalmente, a que en ella es donde tiene razón de ser el objeto social de la organización.

Una vez aprobado los proyectos por ambas partes, tanto la empresa como el cliente, el jefe de proyecto es el encargado de asignarle una fecha de inicio. Durante el proceso de planeación, el planificador asigna a las tareas de los planes de diseño, el rol que llevara a cabo la misma, pero en el siguiente proceso, el jefe de división es el responsable de asignar el recurso humano (personal disponible).

Con el objetivo de que la organización tenga total conocimiento acerca de los proyectos a ejecutar, se lleva cabo un proceso de notificación, brindando la información necesaria a los usuarios. Durante este proceso es muy importante que se le notifiquen a todas las áreas de la empresa, ya que estas son las encargas de darle cumplimiento a los planes de diseño generales los cuales están compuestos por planes de diseño, estos son confeccionados por especialidad y éstos a su vez por tareas. Cuando el proyecto comienza a ejecutarse intervienen en él los especialistas y divisiones de diferentes especialidades. La realización del proyecto depende en su mayoría de la elaboración correcta y en los plazos establecidos de las diferentes tareas que realizan los especialistas encargados.

La documentación referente a los proyectos se encuentra almacenada en un servidor (carpeta contenedora), al cual se puede acceder mediante la red. El área de informática es la responsable de crear y compartir esta información, así como permitir el acceso para aquellos usuarios que estén involucrados en los proyectos.

Los especialistas acceden a la documentación referente a los proyectos, que se encuentra en extensión .mpp debido a la utilización del Microsoft Project (MP), buscando las tareas asignadas de forma individual y confeccionando un nuevo fichero en el mismo formato, por lo que el proceso de filtrado de la información se hace tedioso.

Para conocer el estado del proyecto en tiempo real la dirección de la empresa tiene que realizar múltiples llamadas telefónicas a cada una de las divisiones involucradas en un proyecto en particular, enviar correos electrónicos para solicitarles a los jefes de divisiones información en tiempo real de los mismos o

esperar períodos regulares de tiempo (semanalmente) para la obtención de información importante.

Actualmente el proceso de control y seguimiento a proyectos en la Ceproníquel es llevado a cabo por todo el personal vinculado a la dirección de la institución, utilizando herramientas como el MP, el cual es un *software* de administración de proyectos diseñado, desarrollado y comercializado por *Microsoft* para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo. Sin embargo, la empresa no utiliza el MP como cliente, es decir el personal no se conecta a un servidor de proyectos como el *Microsoft Project Server*, esto trae como consecuencia que la empresa no posea un flujo que permita la corrección de la ejecución de las tareas a realizar que no sea por vía de acceso red a una carpeta contenedora de los planes de diseño con extensión .mpp.

Es necesario aclarar que la solución que se propone en el presente trabajo no pretende remplazar esta herramienta, sino, ofrecer una alternativa que posibilite el uso del MP y las funcionalidades que brinda el portal de la empresa, integrando la información para satisfacer las necesidades mencionadas.

1.2. Módulo de seguimiento y control a proyectos del portal Ceproníquel

La base de este estudio estuvo fundamentada en la necesidad de la utilización de un portal empresarial que pueda integrar varios módulos, garantizando la retroalimentación de la información en la empresa. El módulo de seguimiento y control a proyectos desarrollado en 2013 sigue casi en su totalidad el flujo de trabajo de este proceso en Ceproníquel, y mantiene una comunicación estrecha con el módulo de planificación elaborado.

Tecnologías y herramientas utilizadas

Para su implementación se utilizaron herramientas y tecnologías con un alto nivel de integración con Liferay, seleccionado como la plataforma web para el desarrollo del portal empresarial Ceproníquel; y que sus ventajas respondan específicamente a los requerimientos del cliente.

Se hizo uso de Java como lenguaje de programación, y Eclipse como entorno de desarrollo integrado.

Para el manejo de las transacciones y el mapeo objeto relacional se utilizaron los *frameworks* Spring e Hibernate respectivamente. El trabajo con estos *frameworks* es complejo, lo que extiende el tiempo de desarrollo; además, no se puede reutilizar el código de un *portlet* en otro, lo que lo vuelve ineficiente en este contexto.

Para el desarrollo de las vistas se utilizó el *framework* ZK, que constituye una buena opción debido a la facilidad con la que se integra al portal Liferay.

Estructura

Para el desarrollo del módulo se utilizó la metodología ICONIX, que en las primeras etapas sugiere la captura de requisitos, y los agrupa en el diagrama de paquetes según la lógica del negocio. En la Figura 4 se muestra este diagrama.

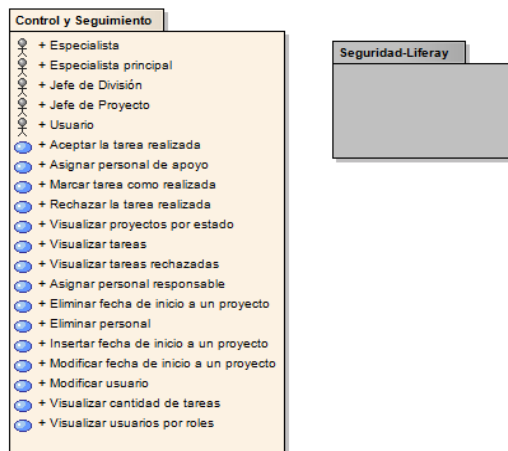


Figura 4 Diagrama de paquetes del módulo

Tomado de :

También sugiere la metodología la especificación de la arquitectura técnica, que abarca el modelo de despliegue. El del módulo de seguimiento y control a proyectos se puede visualizar en la Figura 5.

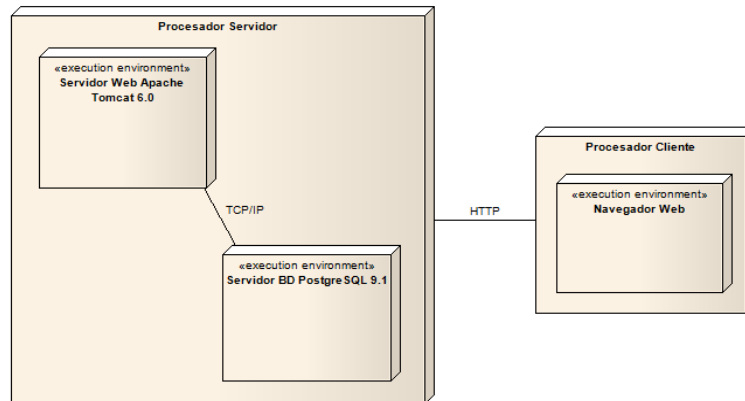


Figura 5 Modelo de despliegue del módulo

Tomado de:

Otros elementos que se tuvieron en cuenta durante la implementación que facilitan en gran parte la etapa de mantenimiento fueron el uso de los estándares de código, y la utilización del patrón de diseño Modelo Vista Controlador, para la separación en capas del software.

1.3. Mantenimiento de *software*

El proceso de desarrollo del *software* se intenta estructurar en diversas etapas, a esta descomposición se le conoce como Ciclo de vida del *software*. Las tareas de mantenimiento son las últimas en realizarse en dicho ciclo. Al ser la actividad de mantenimiento la última en el ciclo, no quiere decir que es la menos importante dentro de este. Este puede definirse como la modificación de un producto *software* después de su entrega al cliente o usuario para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno.

El mantenimiento de *software* (MS) se define identificando cuatro actividades diferentes: mantenimiento correctivo, mantenimiento adaptativo, mejora o mantenimiento de perfeccionamiento y mantenimiento preventivo. Sólo el 20% del trabajo de mantenimiento se emplea en “corregir errores”, el restante 80% se dedica a adaptar los sistemas existentes a los cambios en su entorno externo, realizar las mejoras que solicitan sus usuarios y rediseñar una aplicación para usarla en lo futuro.

1.3.1. Tipos de Mantenimiento

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del *software* y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Por lo tanto luego de haberse realizado la fase de mantenimiento se puede obtener una nueva versión del *software*. **Mantenimiento correctivo**

Tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de provocar un fallo. Un fallo se produce cuando el comportamiento de un sistema difiere con respecto al comportamiento definido en la especificación. En la Figura 6 se muestra una gráfica que resume un estudio realizado sobre los orígenes de los defectos del *software*.

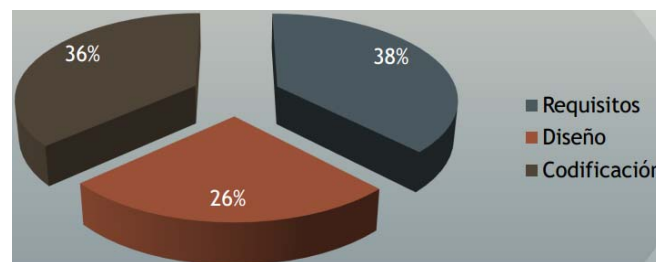


Figura 6 Origen de los defectos del *software*

Mantenimiento adaptativo

Consiste en la modificación de un programa debido a cambios en el entorno (*hardware* o *software*) en el que se ejecuta. Desde cambios en el sistema operativo, pasando por cambios en la arquitectura física del sistema informático, hasta en el entorno de desarrollo del *software*. Este tipo de mantenimiento puede ser desde un pequeño retoque hasta una reescritura de todo el código.

Mantenimiento de perfeccionamiento

Conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Mantenimiento preventivo

Modificación del *software* para mejorar las propiedades de dicho *software* (calidad y mantenimiento) sin alterar sus especificaciones funcionales. Incluir sentencias que comprueben la validez de los datos de entrada, reestructuración de los programas para aumentar su legibilidad o incluir nuevos comentarios. Este tipo de mantenimiento utiliza las técnicas de ingeniería inversa y reingeniería. El mantenimiento para la reutilización especializado en mejorar la reusabilidad del *software* se incluye en este tipo.

1.3.2. Actividades y problemas de mantenimiento de *software*

Las actividades de mantenimiento se agrupan en tres categorías funcionales:

1. **Comprensión del *software* y de los cambios a realizar (Comprender):** es necesario el conocimiento a fondo de la funcionalidad, objetivos, estructura interna y requisitos del *software*. Alrededor del 50% de tiempo de mantenimiento se dedica a esta actividad, a consecuencia de lo cual, las herramientas CASE² incorporan utilidades que automatizan este tipo de tareas aumentando de manera notable la productividad.
2. **Modificación del *software* (Corregir):** crear y modificar las estructuras de datos, la lógica de procesos, las interfaces y la documentación. Los programadores deben evitar los efectos laterales provocados por sus cambios. Esta actividad representa $\frac{1}{4}$ del tiempo total de mantenimiento.
3. **Realización de pruebas (Comprobar):** realizar pruebas selectivas que nos aseguren la corrección del *software*.

Tabla 1 Importancia de las actividades de mantenimiento

Fuente

<i>Categoría</i>	<i>Actividad</i>	<i>Tiempo (%)</i>
<i>Comprensión del software y de los cambios a realizar</i>	Realización de pruebas	18

²

Acrónimo del inglés *Computer Assisted Software Engineering*: Ingeniería de software asistida por computadora.

	Estudiar la documentación	6
	Estudiar el código	23
<i>Modificación del software</i>	Modificar el código	19
	Actualizar la documentación	6
<i>Realización de pruebas</i>	Diseñar y realizar pruebas	28

Problemas del mantenimiento

Entre los problemas de mantenimiento más habituales se encuentran:

1. Es habitual realizar el mantenimiento de forma *ad hoc* en un estilo libre del programador. Esto es debido a que no existen o son poco conocidos los métodos, técnicas y herramientas que proporcionan soluciones globales al problema del mantenimiento.
2. Después de cada cambio los programas tienden a ser menos estructurados. Como consecuencia se produce una documentación desfasada, código que no cumple los estándares, incremento en el tiempo de comprensión de los programas o incremento de los efectos secundarios de los cambios.
3. Los sistemas que son mantenidos son cada vez más difíciles de cambiar.
4. Los usuarios participan poco en el desarrollo del *software*, con el riesgo de que no satisfaga sus necesidades y aumenten los esfuerzos en el mantenimiento.

1.4. Herramientas y tecnologías

Hasta este punto, se conoce cómo funciona el proceso de planificación de proyectos en Ceproníquel y los problemas detectados; en consecuencia, en el siguiente subepígrafe se plantean las herramientas y tecnologías existentes que intervendrán en el desarrollo de la solución.

1.4.1. 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 una [intranet](#) mediante un [navegador](#). En otras palabras,

es un *software* que se codifica en un lenguaje soportado por los navegadores *web* el cual se encarga de su ejecución [6].

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, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener las aplicaciones sin distribuir e instalar *software* a miles de usuarios potenciales [6].

1.4.2. Liferay Portal

Un portal es una infraestructura de *software web* que provee interacción con medios relevantes de información (por ejemplo: información/contenido, aplicaciones y procesos del negocio), recursos de conocimiento, y recursos humanos, expuestos de una forma altamente personalizada.

Liferay Portal, o conocido simplemente como Liferay, es una plataforma *web* empresarial para construir aplicaciones de negocios. Fue diseñado para facilitar (a los miembros de una empresa o usuarios externos autorizados) el acceso a las distintas aplicaciones o contenido de información institucional que se desee administrar bajo un entorno de trabajo unificado, con el fin de desarrollar soluciones empresariales con resultados inmediatos y valor a largo plazo. El mismo constituye un gestor de contenidos de código abierto basado en Java, que permite un desarrollo fácil y rápido de portales *web*. Portales como el Liferay están compuestos por un número de *portlets*³ con diferentes funcionalidades, que pueden ir desde el manejo de simples *blogs*⁴, foros de discusión, *wikis*⁵, calendarios compartidos etc., a grandes intranets corporativas, usadas para la comunicación interna a nivel empresarial.

³ En Liferay, un portlet es una pequeña aplicación web escrita en Java que corre en una porción de una página web.

⁴ Un blog es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

⁵ Un wiki o una wiki (del hawaiano wiki, 'rápido') es un sitio web cuyas páginas pueden ser editadas por múltiples voluntarios a través del navegador web.

Según la consultora Gartner⁶, en 2013, Liferay destaca entre las empresas suministradoras de plataformas *web* líderes, junto a Microsoft con *SharePoint*, IBM con *WebSphere* y Oracle con *WebCenter*, presentando productos muy elaborados, capacitados, y con una amplia gama de funcionalidades.



Figura 7 Cuadrante Mágico de Gartner para portales horizontales.

Fuente: <https://www.liferay.com/about-us/awards/gartnermq-portals>

Se decide la utilización de Liferay debido a que, aún perteneciendo a ese subconjunto de líderes mundiales, constituye una compañía de código abierto y su producto Liferay está disponible como una *Community Edition* gratis, lo cual fue un punto fundamental debido a la política de migración hacia el *software* libre que tiene el país, brindando cierta flexibilidad a sus usuarios.

Liferay posee un conjunto de características que los han colocado entre los mejores productos de su tipo en el mercado:

Basado en estándares: Está basado en tecnologías estándares populares, donde se incluyen:

- Construido en Java: Java es un lenguaje de programación popular, por lo que no es muy difícil encontrar desarrolladores que puedan adaptar Liferay a las necesidades de una empresa.
- Basado en componentes probados: Liferay utiliza un conjunto de

⁶ Organización consultora, líder en investigación y consultoría sobre tecnologías de la información.

tecnologías y componentes conocidos y probados para minimizar la ocurrencia de errores. Entre estos están: Apache ServiceMix, Hibernate, Java J2EE/JEE, jBPM, Lucene, PHP, Ruby, Spring y Alloy UI.

- Uso de formas estándares para comunicación con otros *softwares*: Existen varios estándares para compartir datos entre fragmentos de *software*. Entre los estándares implementados por Liferay están: AJAX, iCalendar, Microformat, JavaJSR-168, JSR-127, JSR-170, JSR-286 (Portlet 2.0), JSR-314 (JSF 2.0), OpenSearch, JSON.
- Alloy UI: Soporta HTML 5, CSS 3, y YUI 3 (*Yahoo! User Interface Library*).

Publicación de contenidos basada en roles: Los portales permiten que diferentes tipos de usuarios accedan a un sitio web único y presentarles una página adaptada según sus roles, grupos, organizaciones o preferencias personales. Liferay permite a los administradores establecer políticas corporativas de acceso a contenidos y funcionalidades, pudiendo especificar quien puede editar y publicar contenidos, ficheros, comunidades, y aplicaciones, desde una interfaz centralizada.

Fácil de usar: La facilidad de uso de Liferay ha sido uno de los aspectos claves para conseguir su popularidad. Con sólo unos clics de ratón, se puede cambiar tanto el aspecto como la presentación, las configuraciones de los *portlets*, los temas y el formato de las diferentes páginas. Mediante sencillas secuencias de arrastre se pueden añadir y reubicar aplicaciones, herramientas y otros elementos en el portal.

Arquitectura: Liferay se ha desarrollado conforme a una Arquitectura Orientada a Servicios (SOA, por sus siglas en inglés), lo que la convierte en una herramienta para integración de aplicaciones corporativas. La arquitectura de Liferay (ver **Figura 8**) soporta una alta disponibilidad para aplicaciones críticas usando *clustering*⁷, una completa distribución de la cache y soporte para replicación en múltiples servidores.

⁷ Consiste en la división de los datos en grupos de objetos similares, simplifica la información.

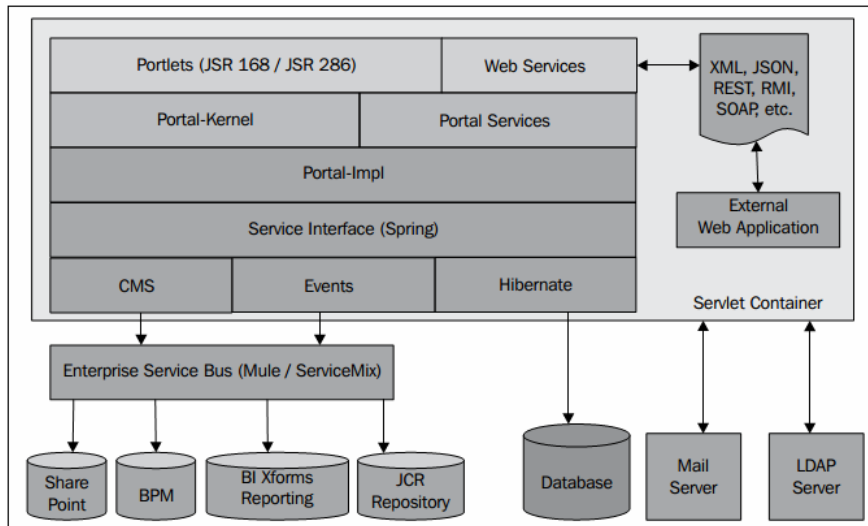


Figura 8 Arquitectura de Liferay.

Fuente:

Constructor de Servicios: El Constructor de Servicios (*Service Builder*) es el generador de código de Liferay para persistencia en bases de datos. Esta utilidad (que está incluida en Liferay) crea código Java y SQL para el acceso a la base de datos desde los *portlets*. Utiliza los *frameworks* Spring e Hibernate para implementar esta funcionalidad, y realiza automáticamente gran parte del trabajo que los desarrolladores hacían manualmente, por lo que estos pueden dedicar más tiempo a implementar la lógica del negocio. También genera buscadores que permiten a los desarrolladores el acceso a datos como objetos Java. Mediante el uso de patrones de diseño conocidos ampliamente como DAO (*Data Access Objects*) y DTO (*Data Transfer Objects*), implementa un diseño consistente y facilita el trabajo de los desarrolladores. Es importante destacar que Liferay utiliza *Service Builder*, por lo que es una herramienta probada que produce código para emplear en desarrollo empresarial.

Service Builder es una herramienta que utiliza Hibernate y Spring como base, generando las configuraciones correspondientes a ambos *frameworks*, así como el código necesario en Java para hacer las entidades definidas persistentes en la base de datos. Utiliza las capas DAO y DTO para persistencia en base de datos, generándolas automáticamente. La capa DTO se comunica con la capa de Servicios, que también es generada, y que permite trabajar con objetos que necesitan persistir o fueron extraídos de la BD. Se llama a métodos de la capa DAO, que invocan métodos en la capa de

Persistencia para realizar la persistencia. La lógica del negocio necesita llamar a métodos genéricos que no tienen nada que ver con la BD, es entonces donde DTO entra en acción, actuando como mediador entre la lógica de negocio y el código de BD subyacente.

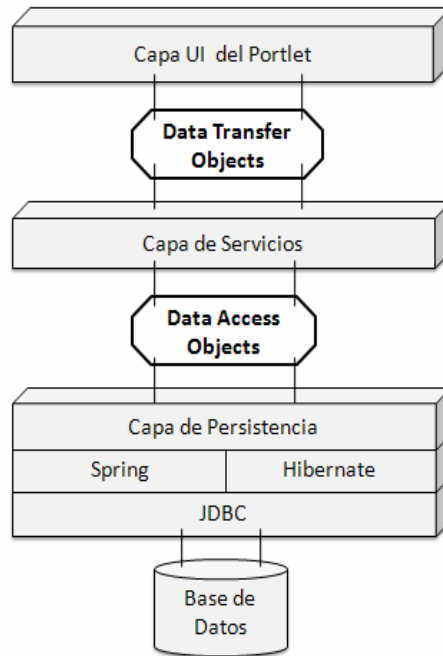


Figura 9 Arquitectura de Liferay Service Builder

Fuente:

1.4.3. Framework ZK

El núcleo de ZK es un mecanismo conducido por eventos basado en Ajax⁸, completamente en Java, sustentado sobre 70 componentes XUL y 80 componentes XHTML, y un lenguaje de marcado para el diseño de interfaces de usuario. Los programadores diseñan las páginas de su aplicación en componentes XUL/XHTML ricos en características, y los manipulan con eventos disparados por la actividad del usuario.

ZK utiliza el procesamiento centrado-en-el-servidor para que la sincronización de componentes y el envío de múltiples peticiones del cliente al servidor se haga automáticamente y los códigos de Ajax sean completamente transparentes para los desarrolladores de aplicaciones web. A continuación,

⁸ Ajax, acrónimo de *Asynchronous JavaScript And XML* ([JavaScript](#) asíncrono y [XML](#)), es una técnica de [desarrollo web](#) para crear aplicaciones interactivas o [RIA](#) (*Rich Internet Applications*).

en la Figura 10 se muestra la arquitectura de ZK la cual explica su comportamiento.

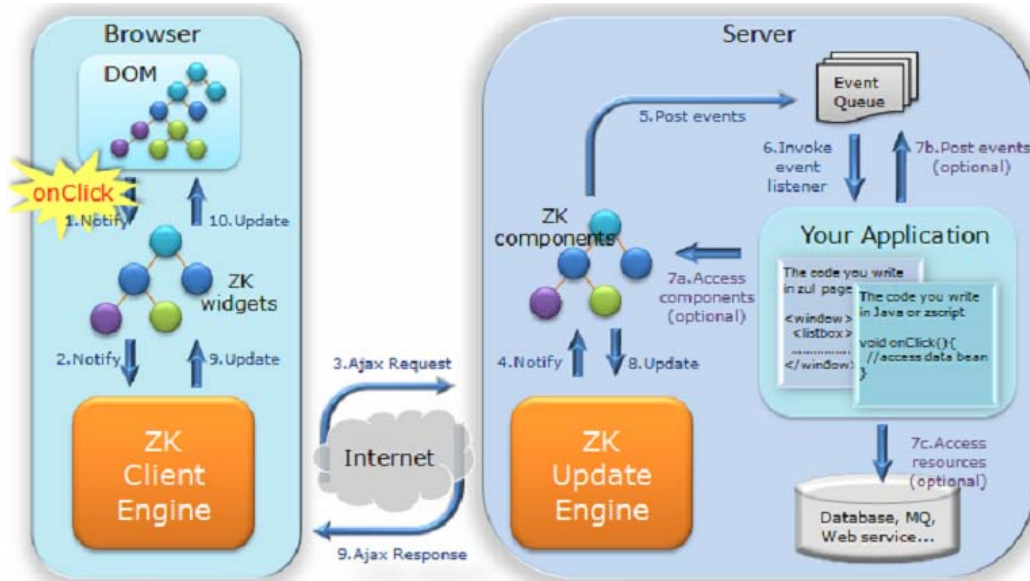


Figura 10 Arquitectura de ZK

Además de la programación basada en componentes y manejada por eventos, ZK soporta un lenguaje de marcación para la definición de una potente interfaz de usuario llamada ZUML.

Es necesario destacar que una aplicación desarrollada en ZK se acopla perfectamente con Liferay formando parte de él como un *portlet*, haciendo que el uso de este *framework* sea ideal para llevar a cabo la aplicación que se desea construir. Por lo antes expuesto además de que existe una previa experiencia de trabajo con ZK se decide el uso del mismo.

1.4.4. Entorno de Desarrollo Integrado

Existe un conjunto de herramientas para el desarrollo de aplicaciones en Java, tanto web como de escritorio, también conocidas como IDEs⁹ (del inglés *integrated development environment*). Dentro de este conjunto resaltan dos en particular, Eclipse y Netbean. Para estos IDEs existen *plugins* que permiten el trabajo con ZK y Liferay, mencionados en este capítulo. Por lo antes expuesto la selección de una de estas herramientas se hace indiferente debido a que

⁹ Un entorno de desarrollo integrado (*IDE* por sus siglas en inglés), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

ambas brindan las prestaciones necesarias. Para el desarrollo de la solución propuesta en este trabajo se decidió por Eclipse.

Eclipse es un IDE 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++¹⁰ o aplicaciones Java [17].

Eclipse emplea módulos o *plugins* 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 [17].

Como se ha mencionado, la filosofía que tiene Eclipse de extender sus funcionalidades a base de *plugins*, hace que el mismo se integre fácilmente con una gran cantidad de herramientas haciendo que las que han sido mencionadas con anterioridad no sean la excepción.

1.4.5. Sistemas Gestores de Bases de Datos

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una BD de manera que sea tanto práctica como eficiente. Estos están diseñados para gestionar grandes cantidades de información, lo que implica tanto la definición de estructuras para almacenarla como la provisión de mecanismos para su manipulación. Además, deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. **PostgreSQL**

¹⁰ Es un lenguaje imperativo orientado a objetos derivado de C.

PostgreSQL es un SGBD objeto-relacional de libre y de código. Corre en los sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Tiene interfaces nativas de programación para C/C++, Java, Python, entre otros y extensa documentación. A continuación se muestran las características más notables de esta herramienta:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de clientes y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de *rollback*¹¹, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel.

1.4.6. Metodologías de desarrollo de software

Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías imponen un proceso disciplinado sobre el desarrollo de *software* con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de *software* tiene como principal objetivo aumentar la calidad en todas y cada una de sus fases de desarrollo. No existe una metodología de *software* universal, pues toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable.

Las metodologías de desarrollo de *software* se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

Metodologías ágiles:

Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar

¹¹ Es una operación que devuelve a la base de datos a algún estado previo.

cambios de último momento. Se puede hacer mención dentro de las metodologías ágiles a: Programación Extrema (*XP* por sus siglas en inglés), Scrum y Crystal.

XP

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

Metodologías robustas

Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: *Microsoft Solution Framework* (*MSF* por sus siglas en inglés), MÉTRICA 3 y *Rational Unified Process* (*RUP* por sus siglas en inglés).

RUP

RUP es un proceso para el desarrollo de un proyecto de *software* que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Dicho proceso tiene tres características fundamentales. Es dirigido por casos de uso, es decir, el proyecto se orienta a la importancia que tiene para el usuario lo que debe hacer el producto. Es centrado en la arquitectura pues relaciona la toma de decisiones que indica cómo tiene que ser constituido el sistema y en qué orden se debe hacer. Es iterativo e incremental, divide el proyecto en mini proyectos donde los casos de usos y la arquitectura cumplen sus objetivos de manera más depurada.

1.4.6.1. ICONIX

ICONIX es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Presenta claramente las actividades de cada etapa y exhibe una secuencia de datos que deben ser seguidos. Un aspecto que contribuyó a la selección de esta metodología fue que está entre la complejidad de RUP y la simplicidad de XP.

Etapas de ICONIX:

ICONIX consta de cuatro etapas y en cada una de estas se realizan una serie de tareas como se muestra a continuación:

1. Análisis de requisitos.

- Modelo del dominio.
- Prototipación rápida.
- Modelo de casos de uso.

2. Análisis y diseño preliminar.

- Descripción de los casos de uso.
- Análisis de robustez.

3. Diseño detallado.

- Diagrama de secuencia.

4. Implementación.

- Escribir /Generar el código.

Características

- **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 de software producidos.

- **Dinámica del UML:** La metodología ofrece un uso dinámico del Lenguaje Unificado de Modelado (*UML* por sus siglas en inglés) por que utiliza algunos diagramas del UML, sin exigir la utilización de todos, como en el caso de RUP.

Conclusiones del capítulo

En este capítulo se presentó la base teórica que sustenta a la presente investigación. Se realizó un análisis del módulo de seguimiento y control a proyectos del portal de la empresa, y a partir de un análisis se tuvieron en cuenta un conjunto de herramientas y tecnologías que en conjunto contribuyeran al desarrollo de la solución propuesta.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

En este capítulo se tendrán en cuenta los temas relacionados con la metodología de desarrollo de software utilizada para llevar a cabo la solución propuesta, la cual fue abordada en el capítulo anterior. Se especifican los requisitos nuevos que debe cumplir el sistema, se describe el funcionamiento interno de la propuesta a través del modelo del dominio así como el modelo de casos de uso con los diagramas de robustez y de secuencia correspondientes. Se realiza una valoración de sostenibilidad del sistema propuesto de acuerdo con las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

2.1. Mantenimiento correctivo

El mantenimiento correctivo se propone cambiar el software para corregir los defectos. Aunque los programas hayan sido sometidos a pruebas rigurosas es muy probable que puedan seguir teniendo errores.

A continuación se describen algunos de los errores que se detectaron y correcciones que se realizaron en el módulo de seguimiento y control a proyectos:

Manipulación incorrecta de las conexiones a la Base de Datos (BD):

No se manipulan las conexiones a la BD de forma correcta, lo que provoca que el servidor de BD colapse luego de un tiempo de uso de la aplicación. Para corregir este error se decidió utilizar las funcionalidades que brinda Liferay para el manejo de conexiones a la base de datos.

Diseño poco intuitivo de las interfaces visuales:

El diseño de algunos elementos visuales, como los componentes *Tabs* pequeños, dificulta el acceso de forma intuitiva a la información por parte de los usuarios. Por ello se decidió rediseñar las interfaces gráficas de la aplicación, haciendo mayor uso de las listas y agrandando las fuentes de algunos encabezados para facilitar la lectura por parte de los usuarios.

2.2. Mantenimiento preventivo

Como se explicó en el capítulo anterior, el módulo de seguimiento y control a proyectos trabaja con los *frameworks* Spring e Hibernate, haciendo bastante

complejas la persistencia y el acceso a datos en la BD. Esto extendía el tiempo de desarrollo y dificultaba la reutilización de código de un *portlet* en otro.

Para optimizar esta parte del software y hacer uso en profundidad de los beneficios que ofrece Liferay, se decidió utilizar el constructor de servicios que este provee. *Service builder* proporciona una interfaz visual que permite el modelado de la entidades del negocio, y a partir del modelo creado se generan los servicios de la aplicación (ver Figura 11).

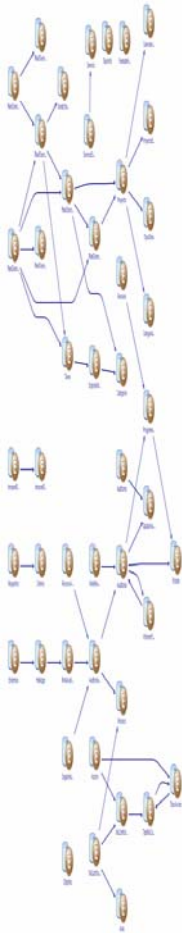


Figura 11 Modelo de entidades del portal Ceproníquel

2.3. Mantenimiento perfectivo

Durante el mantenimiento perfectivo se incorporaron nuevas funcionalidades de importancia sugeridas por el cliente para hacer más usable la aplicación, como la informatización de la solicitud de servicios técnicos que se realiza en la empresa, ya que los proyectos se llevan a cabo en el área de producción a partir surgen de este proceso, descrito en el Anexo 1.

2.3.1 Análisis de requisitos

En esta fase se definen todos los requisitos que en principio deberían ser parte del sistema. Se debe capturar información sobre lo que les gusta y lo que les desagrada a los usuarios. Se crea el modelo del dominio, se identifican los casos de uso, se gráfica y se organizan lógicamente los mismos en grupos y por último se captura la información en un diagrama de paquetes [23].

2.3.1.1 Modelo del Dominio

En la práctica el modelo del dominio es un simple diagrama con líneas entre las diferentes clases (objetos del dominio) que muestran la relación entre estas [23]. El modelo del dominio está compuesto por varios pasos los cuales son:

- Listar los requerimientos del sistema.
- Extraer los sustantivos o frases sustantivas de los requerimientos.
- Realizar el diagrama del modelo de dominio.

Requerimientos Funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A continuación se muestra la lista de los requerimientos funcionales.

1. El sistema debe permitir al **especialista de mercado** registrar una **solicitud de servicios técnicos**, que genera un nuevo proyecto, y especificar:
 - α) Datos generales del cliente.
 - β) El **tipo de obra**.
 - χ) **Especialidades y Servicios** involucrados.
2. El sistema debe permitir al **planificador** importar a la plataforma un **plan de diseño** de un **proyecto** con formato .mpp.
3. El sistema debe permitir al director de proyecto visualizar **tareas** del proyecto.

Capítulo 2: Descripción de la solución propuesta

4. El sistema debe permitir al director de proyecto visualizar el estado general de un proyecto.
5. El sistema debe permitir al jefe de división visualizar **tareas** del proyecto según su división.
6. El sistema debe permitir al jefe de división visualizar el estado general de un proyecto en el cual su división está involucrada.
7. El sistema debe permitir al especialista principal visualizar tareas del proyecto según su especialidad.
8. El sistema debe permitir al especialista principal visualizar el estado general de un proyecto en el cual su especialidad está involucrada.
9. El sistema debe permitir notificar en el portal la creación de un nuevo proyecto a los roles: jefe de proyecto, especialistas principales de cada especialidad involucrada y jefes de división.

Tabla 2 Definición de los principales conceptos del modelo del dominio

Objeto	Definición
Proyecto	Proceso único que consiste en un conjunto de actividades coordinadas y controladas, con fecha de inicio y fin, tiempo, costo y recursos.
División	Área de la empresa conformada por un conjunto de especialidades.
Especialidad	Constituye un departamento especializado en una actividad específica.
Plan de diseño	Es donde se plasman todas las tareas a realizar para la realización de un proyecto, donde se sigue un orden.
Tarea	Actividades a desempeñar en un proyecto, las cuales serán desarrolladas por los trabajadores, siendo específicas de la empresa.

Director o 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.
Servicio	Los servicios son las diferentes prestaciones que brinda cada especialidad y están presentes en los proyectos.
Planificador	Es la persona que elabora los Planes de Diseños Generales para cada proyecto.

Diagrama del modelo del dominio

Luego de haberse obtenido, organizado y analizado los principales conceptos de la lista de requerimientos para la construcción del modelo del dominio, a continuación en la Figura 12 se muestra el diagrama donde se ilustran las relaciones entre estos objetos.

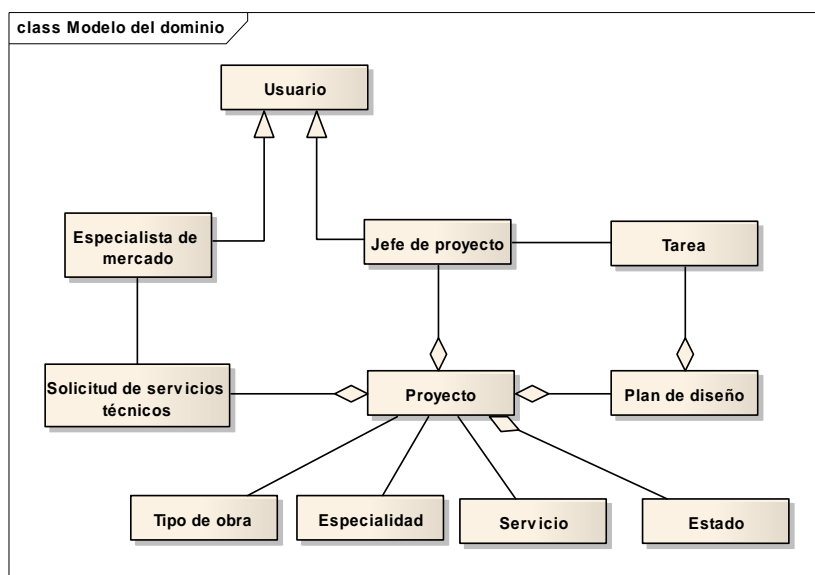


Figura 12 Modelo del dominio

2.3.1.2 Modelo de casos de uso

Los casos de uso brindan una manera estructurada de capturar los requerimientos de comportamiento de un sistema, de esta forma se puede crear un diseño razonable de estos. El modelo de casos de uso es un catálogo de las funcionalidades del sistema. Cada caso de uso representa una interacción que el usuario o actor experimenta cuando usa el sistema .

Un sistema puede llegar a tener un número elevado de casos de uso, de esta forma es necesario organizarlos para tener el control sobre ellos. El UML brinda la solución con un mecanismo de paquetes que es una forma de agrupar elementos como clases, diagramas o casos de uso. Además permite dividir el trabajo en diferentes áreas. A continuación en la figura se muestra el diagrama de paquetes del sistema.

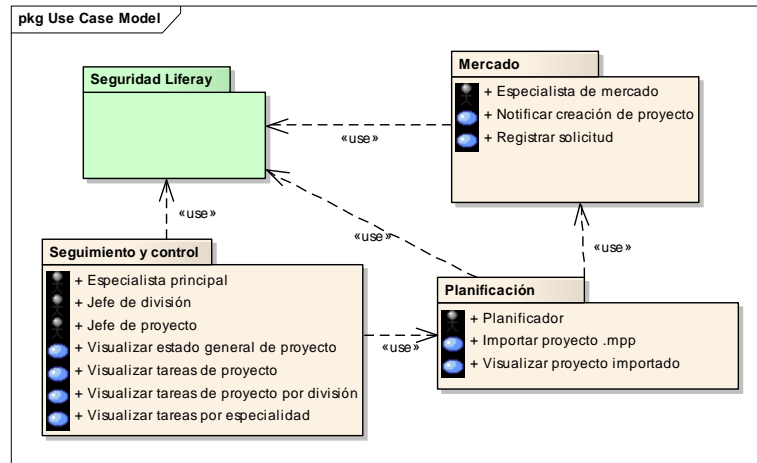


Figura 13 Diagrama de paquetes

Como se puede apreciar, en el diagrama de paquetes, el de Seguridad Liferay ha sido diferenciado de los demás, debido a que en la aplicación no es necesario tratar cuestiones referentes a la seguridad. Liferay Portal ofrece un conjunto de prestaciones para el manejo de usuarios, roles y permisos garantizando de esta manera que la información sea visualizada de acuerdo al rol correspondiente. A partir del diagrama de paquetes del sistema a continuación se presentan los diagramas de casos de uso para cada paquete.

El paquete Seguimiento y control contiene las funcionalidades correspondientes a las actividades de este proceso, incluyendo la visualización del diagrama de Gantt y de los estados de las tareas y los proyectos. Este hace uso del paquete de Planificación.

El paquete de Mercado contiene las funcionalidades del proceso de solicitud de servicios técnicos, y partiendo de este se procede a la planificación, por lo que se utiliza en el paquete de Planificación.

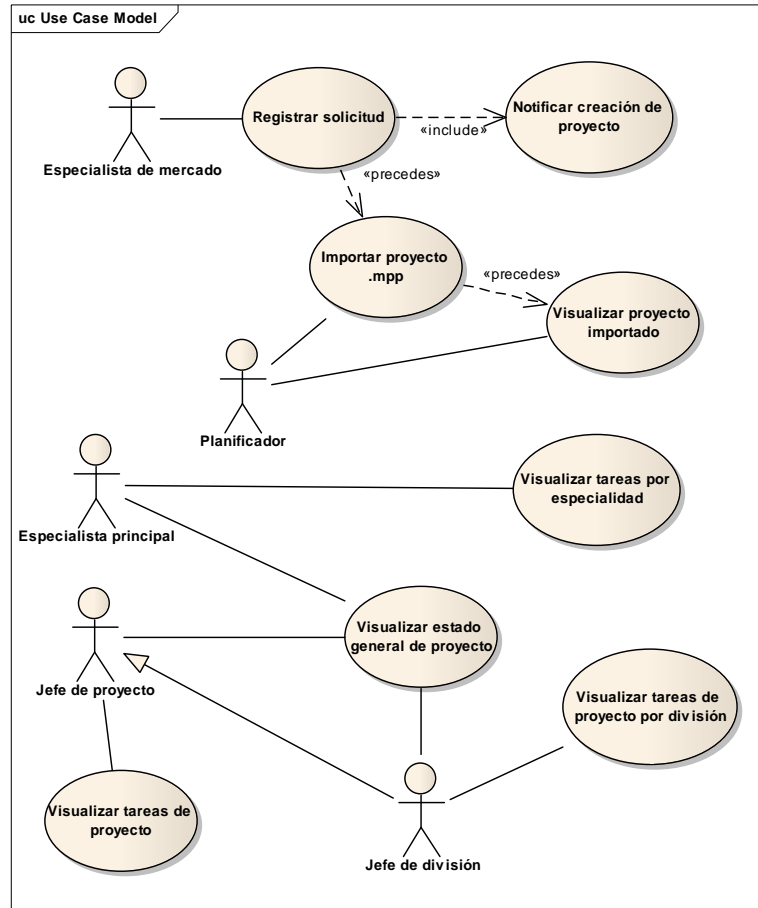


Figura 14 Diagrama de Caso de Uso del sistema

Actores del sistema

Un actor es representado en el diagrama como una figura incrustada y es análogo a un rol que el usuario puede jugar, pero algunas veces el actor solo deberá ser llamado “Usuario”, pero frecuentemente le es dado un nombre de rol específico . A continuación se muestran los actores que se identificaron.

Tabla 3 Actores del sistema

Actor	Definición
Especialista de Mercado	Es el que registra la solicitud de servicios técnicos del cliente.
Jefe de proyecto	Es el encargado de administrar los procesos internos de cada proyecto.
Jefe de división	

Planificador	
--------------	--

2.3.2 Análisis y diseño preliminar

La etapa de análisis y diseño preliminar es un paso intermedio entre el análisis y el diseño. Consiste en hacer un diseño exploratorio para un mejor entendimiento de los requerimientos, refinar y remover ambigüedades que surjan [23].

Descripción de los casos de uso

Los casos de uso permiten describir la utilización del sistema en el contexto del modelo de objeto. Los casos de uso deben ser descritos siguiendo la regla de los dos párrafos, en voz activa, usando un flujo respuesta/evento entre el usuario y el sistema, y utilizando una estructura de oración sustantivo-verbo-sustantivo. En la descripción de los casos de uso se reflejan dos cursos, un curso básico que muestra cómo se debe realizar el proceso y un curso alternativo para describir los posibles errores o sucesos que pueden ocurrir durante la operación [23].

A continuación se muestra la descripción de algunos de los casos de uso (ver en el [Anexo 2](#) las demás descripciones de los casos de uso).

Tabla 4 Caso de uso: Registrar solicitud

Paquete Mercado: Registrar solicitud	
Curso básico	El Especialista de Mercado (EM) selecciona la opción Insertar solicitud de servicios técnicos de la vista principal. El sistema muestra la vista de nueva solicitud. El EM llena todos los campos necesarios y hace clic en guardar. El sistema valida el formulario y luego inserta la solicitud en la base de datos.
Curso alternativo	El Especialista de Mercado ha dejado campos necesarios sin llenar o ha introducido valores incorrectos, el sistema procede a mostrar un mensaje informando del error. La sesión ha expirado, el sistema redirecciona al planificador a la vista de inicio de sesión.

Tabla 5 Caso de uso: Realizar reporte de plan de diseño

Paquete Seguimiento y Control: Importar proyecto en .mpp	
Curso básico	El planificador hace clic en la opción asignar proyecto de la vista. El sistema muestra un explorador para la selección del proyecto en mpp. El planificador abre un proyecto. El sistema carga los datos del fichero seleccionado, los inserta en la base de datos y los muestra en la vista proyecto.
Curso alternativo	La sesión ha expirado. El sistema redirige al planificador a la vista autenticarse. El fichero seleccionado no contiene datos. El sistema muestra un mensaje de error.

2.3.2.2 Análisis de robustez

El análisis de robustez ayuda a identificar los objetos que participan en cada caso de uso mediante un diagrama de robustez. En el diagrama de robustez se debe ilustrar gráficamente las interacciones entre los objetos participantes de un caso de uso. Este diagrama permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos participantes de cada caso de uso [23].

Los objetos que participan en el diagrama de robustez se clasifican dentro de los tres tipos siguientes:

- **Objetos de interfaz:** usados por los actores para comunicarse con el sistema, generalmente como ventanas, pantallas, diálogos y menús.
- **Objetos entidad:** son objetos del modelo del dominio.
- **Objetos de control:** es la unión entre la interfaz y los objetos entidad. Sirve como conexión entre los usuarios y los datos.

A continuación se muestran algunos de los diagramas de robustez del sistema, el resto se encuentra en el [Anexo 3](#).

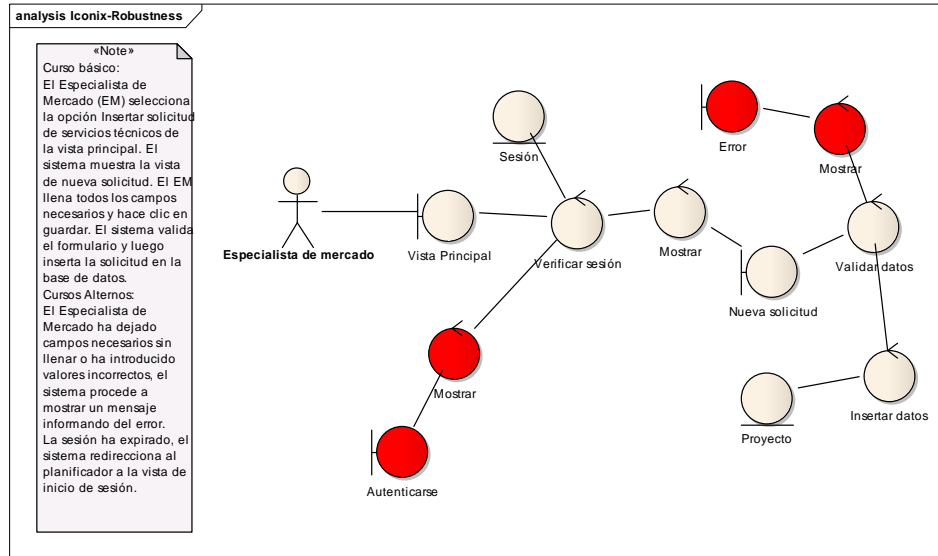


Figura 15 Diagrama de robustez. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.

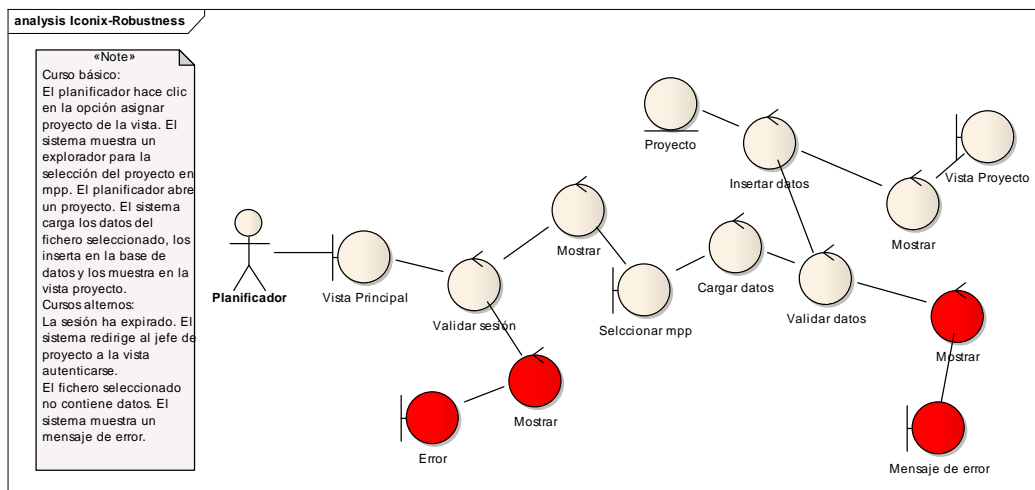


Figura 16 Diagrama de robustez. Paquete: Seguimiento y control. Caso de uso: Importar proyecto en mpp.

2.3.2.3 Arquitectura técnica

El propósito de la Arquitectura Técnica (AT) es tener una noción general del sistema que se va a desarrollar en términos de estructura. Se construye para satisfacer los requerimientos del negocio y del nivel de servicios del sistema que será desarrollado. La AT incluye (pero no se limita) a la topología del sistema, es decir, la localización física en la red y la elección del servidor de aplicación [23].

La arquitectura abarca:

1. Requerimientos no funcionales

2. El modelo de despliegue (red y servidores de aplicación, la topología del sistema y navegadores soportados).

Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener teniendo en cuenta la aceptación de los usuarios finales. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. ICONIX propone que estos deben adaptarse a la arquitectura técnica que va a adoptar la aplicación [14]. Los requerimientos no funcionales determinados para el sistema propuesto en la presente investigación son:

Apariencia o interfaz externas

- 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.
- 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 esencialmente 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 una alta portabilidad.

Seguridad

- Sólo los usuarios autorizados podrán acceder al sistema.

Capítulo 2: Descripción de la solución propuesta

- 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.
- 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 tecnología AJAX, se recomienda Mozilla Firefox 10.0 o superior.
- La máquina computadora servidor debe tener instalado, Servidor Web Tomcat versión 6.0 o superior, servidor de base de datos PostgreSQL 9.1 o superior, *Java Development Kit*, (*JDK* por sus siglas en inglés) versión 1.6.
- Se recomienda una resolución de pantalla de 1360x768

Hardware

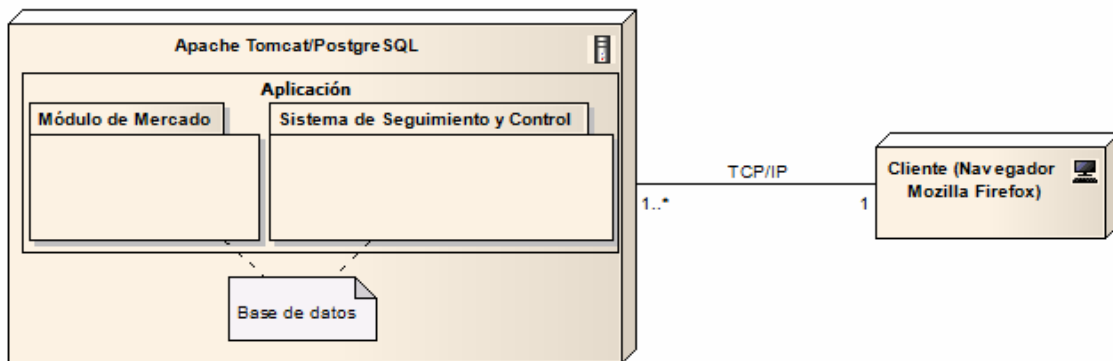
- Para ejecutar el software los requerimientos mínimos de hardware en el cliente son: microprocesador Intel Pentium IV a 2.8 GHz de velocidad de

procesamiento u otro similar, con 1 Gb de memoria RAM y un adaptador de red.

- La máquina computadora servidor debe tener 4 Gb de memoria RAM o superior y debe ser un Pentium IV con un microprocesador cercano a los 3.2 GHz de velocidad.
- La máquina computadora servidor y las computadoras clientes deben estar conectadas a la red.

Modelo de despliegue

Dentro de la arquitectura técnica, un paso muy importante es el modelo 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. Como se muestra en la figura 15 se cuenta con un servidor en el cual están contenidas la aplicación y la base de datos. Este servidor será accedido desde las máquinas clientes a través de un navegador web para procesar las peticiones realizadas por los usuarios.



2.3.3 Figura 17 Diagrama del Modelo de Despliegue Diseño detallado

El análisis y diseño preliminar solo era una teoría de cómo funcionarían las clases de la aplicación, al contrario que con el diseño detallado se logra asignar funciones a cada una de las clases que fueron detectadas. Además es donde se empiezan a ver qué métodos llevarán las clases del sistema. Esto se debe a que, hasta ahora, solo se ha interactuado con los objetos de las clases, con los actores y con otros objetos de manera dinámica, por lo que se tiene suficiente

información como para poder empezar a especificar los métodos de las respectivas clases [23].

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, pero no es hasta los diagramas de secuencias 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 [23]. ICONIX propone realizar un diagrama para cada caso de uso, facilitando entender su diseño.

Los objetivos elementales del diagrama de secuencia son:

- Asignar comportamiento a las clases
- Mostrar en detalle, cómo 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 representan algunos diagramas de secuencia de los casos de uso antes mencionados, los restantes se encuentran en el [Anexo 4](#).

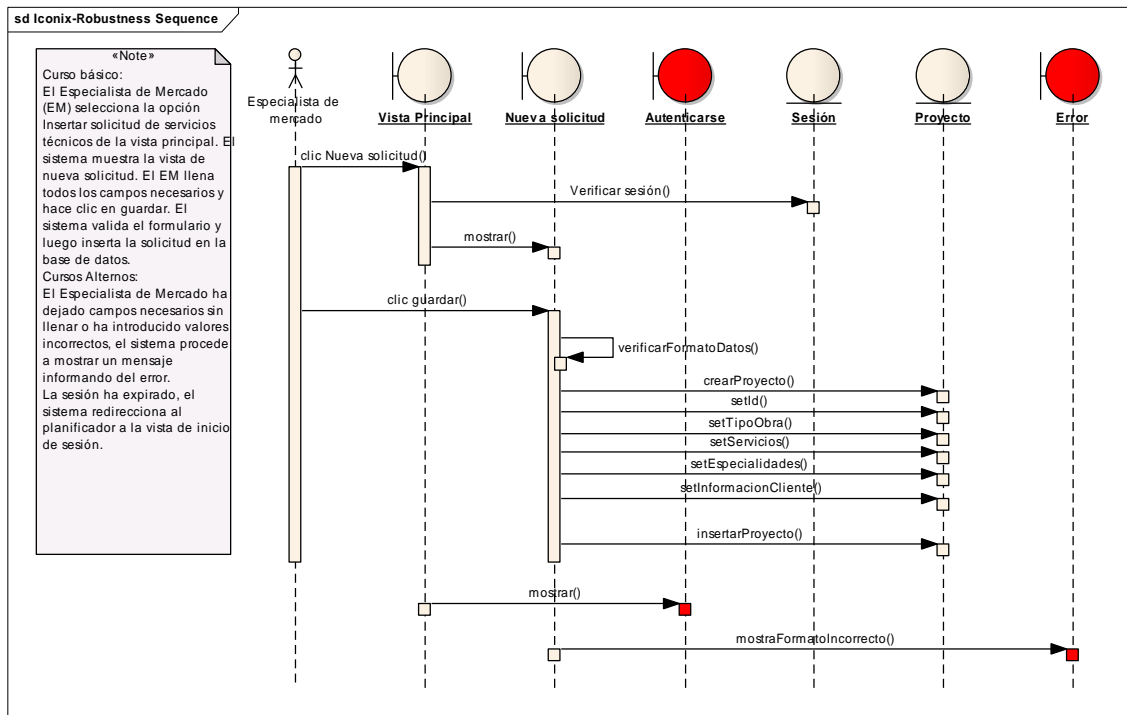


Figura 18 Diagrama Secuencia. Paquete: Mercado. Caso de uso: Insertar solicitud de servicios técnicos.

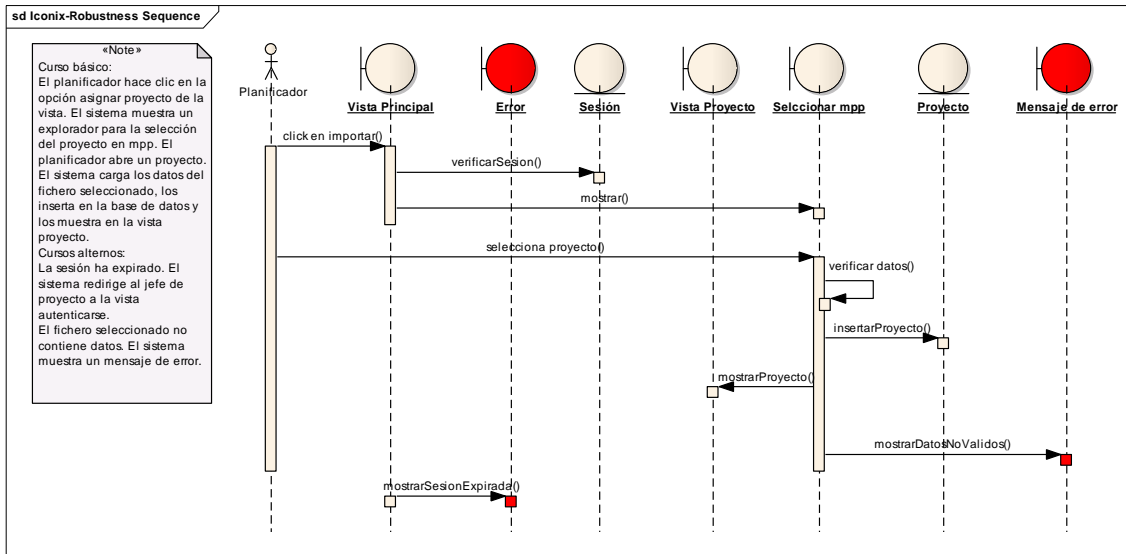


Figura 19 Diagrama Secuencia. Paquete: Seguimiento y control. Caso de uso: Importar proyecto en mpp.

Diagrama de clases persistentes

Las clases persistentes guardan su estado en un medio permanente y representan además un modelo lógico de la base de datos, formado por las tablas que permanecen en la misma. El diagrama de clases persistentes del sistema modela la información que trasciende en el tiempo, incluso después de cerrada la aplicación [24]. En el Anexo 3 se puede apreciar el diagrama de clases persistentes del sistema.

2.3.4 Implementación

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

2.3.4.1 Estándares de código

Cuando se habla de estandarizar dentro del contexto de la programación orientada a objetos, se refiere al seguimiento de un conjunto de reglas para un correcto desempeño y diseño del software en desarrollo:

Comentarios: Los comentarios están escritos en español. Al comentar un método de una clase el mismo se escribirá en mayúscula y en forma de párrafo. Si el comentario es interior, es decir, se encuentra dentro de un método para describir una sentencia o una variable el mismo será en minúscula.

Caracteres a evitar: No se utiliza el carácter “ñ” o “Ñ” en los nombres de variables o métodos debido a que forma parte de los caracteres especiales y esto puede traer consecuencias a la hora de compilar y ejecutar el código.

Nombre para Clases e Interfaces: El nombre de las clases e interfaces están escritos en notación *Camell* y pueden tener incluidos guiones bajos. El nombre de las clases del modelo está asociado a los nombres de las tablas en la base de datos.

Nombre para Objetos: El nombre de los objetos está escrito en minúscula y pueden tener guiones bajos. Los mismos indican propósito, no tienen números incluidos y no exceden los 40 caracteres.

Nombre para Métodos: El nombre de los métodos está en notación *Camell* y pueden tener guiones bajos en algunos casos.

Nombre para Base de Datos: El nombre de la base de datos está en minúscula y solo se aceptan letras en él, su longitud no excede los 20 caracteres.

Nombre para Tablas y Campos: Los nombres de las tablas y campos no requieren prefijo, las letras están todas en minúscula. Solo contienen letras, y las palabras están separadas mediante guiones bajos.

2.3.4.2 Patrones de diseño

Los patrones de diseño (*design patterns*) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un

patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias [16].

A continuación se describirán los patrones de diseño utilizados:

Modelo Vista Controlador

El Modelo Vista Controlador (MVC) separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.
- La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario.
- El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

2.3. Valoración de sostenibilidad

En el presente epígrafe se hace un estudio sobre el impacto que tendrá el sistema propuesto en las dimensiones administrativa, ambiental, socio humanista y tecnológica, con el objetivo de determinar la sostenibilidad del mismo.

Al realizar un producto se debe prever antes de su elaboración que el mismo solucione el problema que le dio origen, que la solución sea adecuada y que genere más beneficios que dificultades. Para ello una vez concretada una

propuesta de solución se realiza una valoración de sostenibilidad del producto teniendo en cuenta el costo, impacto socio ambiental, tecnológico y mejoras en la calidad de vida de usuarios para los que está destinado.

A continuación se realiza la valoración de sostenibilidad del sistema, utilizando un procedimiento que cuenta con 4 dimensiones, administrativa, socio-humanista, ambiental y tecnológica, en el cual se realiza un estudio de factibilidad del producto informático mediante la estimación de costo.



Dimensión administrativa

En la dimensión administrativa se tienen en cuenta el costo de elaboración del producto informático, si ahorra recursos, la generación de ingresos directos o indirectos a la entidad, así como la calidad de la producción y los servicios.

Existen varias alternativas para la estimación del costo y el esfuerzo en Casos de Uso utilizándose el Análisis de Puntos de Función y COCOMO II, o una variante más reciente denominada Análisis de Puntos de Casos de Uso, la cual es en cierta medida similar al análisis de Puntos de Función.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de *Objectory AB*, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo y el costo total estimado para el proyecto a partir de esos factores [26].

A continuación, se detallan algunos de los pasos más importantes a seguir para la aplicación de éste método. En el [Anexo 6](#) se puede apreciar la estimación del sistema mediante el análisis de Puntos de Casos de Uso.

Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. Los criterios se muestran en la siguiente tabla:

Tabla 6 Factor de Peso de los Actores sin ajustar

Tipo de Actor	Descripción	Factor de peso
Simple	Otro sistema que interactúa con el sistema a des-arrollar mediante una interfaz de programación.	1
Medio	Otro sistema que interactúa con el sistema a des-arrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo. Los criterios se muestran en la siguiente tabla:

Tabla 7 Factor de Peso de los Casos de Uso sin ajustar

Tipo de Actor	Descripción	Factor de peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15

Factor de complejidad técnica (TCF)

Capítulo 2: Descripción de la solución propuesta

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Tabla 8 Factores de complejidad técnica

Factor	Descripción	Peso
T1	Sistema distribuido	2
T2	Objetivos de performance o tiempo de respuesta	1
T3	Eficiencia del usuario final	1
T4	Procesamiento interno complejo	1
T6	El código debe ser reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidad de uso	0.5
T8	Portabilidad	2
T8	Facilidad de cambio	1
T10	Concurrencia	1
T11	Incluye objetivos especiales de seguridad	1
T12	Provee acceso directo a terceras partes	1
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1

Factor de ambiente (EF)

Capítulo 2: Descripción de la solución propuesta

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Tabla 9 Factores de ambiente

Factor	Descripción	Peso
E1	Familiaridad con el modelo de proyecto utilizado	1.5
E2	Experiencia en la aplicación	0.5
E3	Experiencia en orientación a objetos	1
E4	Capacidad del analista líder	0.5
E5	Motivación	1
E6	Estabilidad de los requerimientos	2
E7	Personal part-time	-1
E8	Dificultad del lenguaje de programación	-1

- Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).
- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).

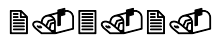
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

A pesar del costo estimado para desarrollar el sistema informático propuesto (ver Anexo 4), no se incurre en gastos debido a que el desarrollador es un estudiante.

Con la realización del mantenimiento propuesto, la empresa Ceproníquel no tendrá que cambiar o mejorar las tecnologías, ya que el PI será realizado en función de los requerimientos técnicos que posee la empresa. El resultado que se esperan con la implantación del sistema es el ahorro de electricidad asociado al funcionamiento del PI, este consumo no incrementará el existente en la empresa, debido a que el PI será implantado en máquinas de actual explotación.

La calidad de la gestión de la información y el flujo de trabajo mejorará considerablemente una vez puesto en marcha el PI propuesto, debido a que la solución que brinda está orientada a dotar de confiabilidad e integridad a la gestión de información del proceso. Las particularidades del PI propuesto podrán minimizar el tiempo de trabajo al mejorar el flujo de información de la empresa, esto se debe a que el PI informatiza una serie de funcionalidades que anteriormente no se habían concebido, apoyará el proceso de toma de decisiones por parte de la dirección de la empresa. Finalmente, las herramientas que se usaron para el desarrollo del sistema en su totalidad son libres, de modo que no es necesario pagar licencia por su uso.

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



Dimensión Socio-Humanista

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

Con respecto al impacto socio-humanista del PI, se puede asegurar que una vez implantada la aplicación, la función de la planificación de proyectos y

Capítulo 2: Descripción de la solución propuesta

seguimiento y control a proyectos en Ceproníquel, contarán con acceso a información íntegro y profundo al flujo de trabajo a desarrollar por cada uno de los especialistas principales y cada actividad a desempeñar, dándole de esta forma, respuesta a la necesidad social que no se tenía propuesta hasta el momento.

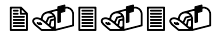
Otro aspecto a valorar es su posible generalización, en dependencia de la necesidad social que resuelve, ya que gracias a sus características de flexibilidad y riqueza de fácil manejo de información ante cambios, pudiera ser adoptado por empresas que posean estilos semejantes en la producción de proyectos de ingeniería, debido a que el sistema será adaptable para cualquier entidad que maneje estos sistemas.

Las condiciones de trabajo se ven mejoradas debido a que se minimiza el tiempo de búsqueda, actualización de datos e intercambio de información, mediante el PI propuesto, el cual permitió un gran nivel de organización y especificación al especialista principal, dotando de confiabilidad e integridad, así como a la gestión de información.

Por otro lado, la implantación del nuevo PI no trae consigo la generación o disminución de empleo en la entidad, su influencia está orientada, por el contrario, a favorecer el trabajo de los proyectistas y directivos que ya son empleadas por la empresa.

El PI influye positivamente en la formación de la responsabilidad de quien lo desarrollará, ya que se trata de un software basado en la producción de proyectos ingenieriles a nivel amplio en la gama de información, lo que trae consigo un alto nivel de compromiso. Durante su elaboración se tuvo en cuenta cómo orientarlo hacia un desarrollo sostenible en su entorno. La experiencia acumulada convierte a su autor en un analista competente en la gestión de proyectos informáticos sostenibles.

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



Dimensión ambiental

El diseño de las interfaces del PI será lo más amigable posible y de fácil uso por parte de los usuarios. Se ahorrarán recursos que provienen del medio ambiente como los materiales de oficina, principalmente el papel, no se contribuye de modo alguno al deterioro del medio ambiente por causa de contaminación por ruido, interferencia, etc., todo lo contrario, su implantación ayuda en gran medida al cuidado del mismo en las diferentes esferas antes mencionadas.

Se utilizarán colores adecuados y un tamaño de letra que no dañen ni cansen la vista, lográndose una mayor aceptación del PI por parte del usuario. La implantación del PI aumentará el uso de la computadora lo que puede causar enfermedades profesionales a los usuarios. Por lo que se recomienda una correcta postura de los usuarios en las sillas, así como tener el monitor a la altura de los ojos y utilizar protector de pantalla además de una adecuada iluminación en el área de trabajo. Además, con el mantenimiento se incorporarán elementos visuales a las aplicaciones que facilitarán

Todos los software que se utilizaron en el desarrollo del PI son libres, esto contribuye a que se puedan reutilizar códigos existentes en el mundo y que sirvan para disminuir el tiempo de desarrollo. Igualmente el mismo código a generar podrá reutilizarse con vista a la implementación de otros módulos o nuevos requerimientos del sistema.

Por todo lo antes expuesto se concluye que el PI es sostenible en la dimensión ambiental.



Dimensión tecnológica

En la dimensión tecnológica se analiza el producto informático (PI) según aspectos del uso de tecnología adecuada y asimilable con el usuario.

Las empresas de proyectos ingenieriles cuentan con una estructura que le permitirá la implantación y correcta explotación del PI. Todas las computadoras están conectadas a la red por lo que pueden acceder y manipular la información de manera rápida. Para el empleo del sistema se capacitará a los usuarios, y se adjuntará la ayuda al sistema.

El servidor donde se encuentra la aplicación Web y la BD puede inesperadamente dañarse, lo que hace al sistema vulnerable, por lo cual se debe realizar copias del mismo y de la BD.

El producto brinda facilidades para la correcta explotación por parte del usuario, con independencia del productor, pues el usuario tiene los privilegios para actualizar la información según el rol que tenga en el sistema.

Por todo lo antes expuesto se concluye que el PI es sostenible en la dimensión tecnológica.

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

Para cumplir con las características que plantean las preguntas científicas es necesario validar cada una de las funcionalidades del sistema, para su procesamiento estadístico fue empleado el método Delphi, el cual permitió la emisión de un pronóstico que ratificó la validez de la solución. Este método emplea una serie de variables que, a través de cálculos, permiten obtener un resultado final en cuanto a la relevancia de la solución.

Se realizó primeramente una encuesta para conocer el grado de competencia de los posibles expertos (Anexo 5). Para ello se consultaron profesionales con experiencia, creatividad, capacidad de análisis y de pensamiento en el tema tratado. Luego de seleccionar 15 expertos de la población de 17 encuestados, se realiza una encuesta para saber el nivel de satisfacción con respecto al producto informático propuesto (Anexo 6).

Después de realizar el procesamiento estadístico de los aspectos analizados en la encuesta, se comprobó en los resultados finales obtenidos son satisfactorios, contribuyendo positivamente en la calidad del sistema, lo cual permitió percibir la satisfacción de las necesidades iniciales.

Al valorar de forma general los resultados obtenidos para los diferentes criterios analizados, se puede concluir que es factible la implantación del sistema por relevancia obtenida en los aspectos encuestados. Estos resultados se pueden apreciar en el **Anexo 7**.

2 . 5 . Conclusiones del capítulo

La utilización de la metodología ICONIX para modelar y construir el sistema propuesto, permitió resolver una gran parte de su desarrollo, sin descartar ninguna de las etapas que lo componen, logrando desarrollar eficientemente el diseño del problema planteado en la investigación.

El análisis de sostenibilidad arrojó que el proyecto es sostenible y perdurable en el tiempo debido a la necesidad que existe de una herramienta informática que les permita gestionar de una forma rápida y fiable el flujo de información referente a la planificación de proyectos.

Mediante la encuesta aplicada, se pudo observar la satisfacción de los usuarios en cuanto a las funcionalidades de la aplicación, las cuales permitieron resolver el problema previamente planteado en la investigación.

CONCLUSIONES GENERALES

- El mantenimiento sobre la base de corrección de errores y extensión de funcionalidades realizado al módulo de seguimiento y control a proyectos del portal Ceproníquel, posibilitó favorecer este proceso, cumpliéndose el objetivo de la investigación.
- Las herramientas y tecnologías seleccionadas constituyeron una buena opción para el desarrollo del sistema, ya que poseen un alto grado de integración entre sí.
- Con la valoración del sistema en las cuatro dimensiones de gestión de sostenibilidad, se determinó que el producto informático es sostenible y perdurable en el tiempo.
- El sistema desarrollado presenta un nivel muy apropiado de aceptación por los usuarios, y una opinión favorable por parte de los expertos, lo cual favorecerá la implantación del mismo.
- El uso de la herramienta Service Builder que brinda Liferay, posibilitó la construcción de servicios para la comunicación con otros portlets.

RECOMENDACIONES

Una vez concluida esta investigación se recomienda continuar el mantenimiento sobre la base de extensión de funcionalidades, como brindar apoyo gráfico para facilitar la toma de decisiones de los diferentes roles involucrados en el proceso de seguimiento y control a proyectos en la empresa Ceproníquel.

REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

Anexo 1: Proceso de Solicitud de servicios técnicos

Es política de la entidad que cualquier trabajador puede hacerle saber a Mercado que un cliente X está interesado en los servicios que se ofertan. Una solicitud puede hacerse a través de varias formas, por vía telefónica, a través del director, de los trabajadores de la entidad, y que el cliente realice personalmente dicha solicitud en Mercado.

Desde que el cliente realiza la solicitud hasta que dicha solicitud del proyecto es Aceptado o Rechazado o Cancelado, intervienen diferentes áreas. A continuación se describe el flujo de este proceso:

1. El Cliente efectúa la solicitud de servicios técnicos a Mercado.
2. Mercado le envía dicha solicitud al Consejo de Producción, el cual está compuesto por el director del proyecto, directores de divisiones y las áreas de apoyo (transporte, economía).
3. Los Directores de las Divisiones son los encargados de informarle a los especialistas principales y a los controladores de calidad sobre dicha solicitud.
4. El jefe de Proyecto como los jefes de las Divisiones, de acuerdo con el proyecto que se quiera realizar pueden seleccionar las especialidades implicadas en dicha solicitud. A lo largo del desarrollo del proyecto puede incluirse una especialidad.
5. El jefe de Proyecto le informa a los jefes de División, que para cuándo se puede fijar la fecha de la reunión.
6. En la visita a la zona se sabe realmente cuáles son las especialidades que tienen que participar en dicho proyecto.
7. Una vez que se sabe cuáles son las especialidades que van a participar, los controladores de calidad de dichas especialidades realizan el Plan de Diseño y el documento de Alcance.
8. Cada plan de diseño y documento de Alcance es revisado por el especialista principal, luego que este los aprueba los envía al jefe de

División, este los revisa y cuando los aprueba los envía al jefe de Proyecto.

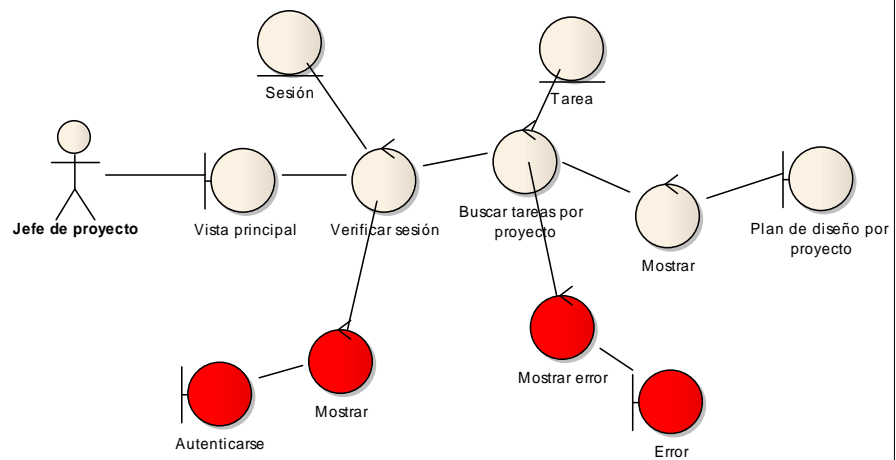
9. El jefe de Proyecto es el encargado de revisar todos los planes de diseño con sus respectivos documentos de Alcances, una vez aprobado, son enviados a Mercado.
10. Mercado a partir de los planes de diseños, confecciona un Plan de Diseño General para oferta el cual es anexado al Documento de Oferta (Suplemento de Contrato).

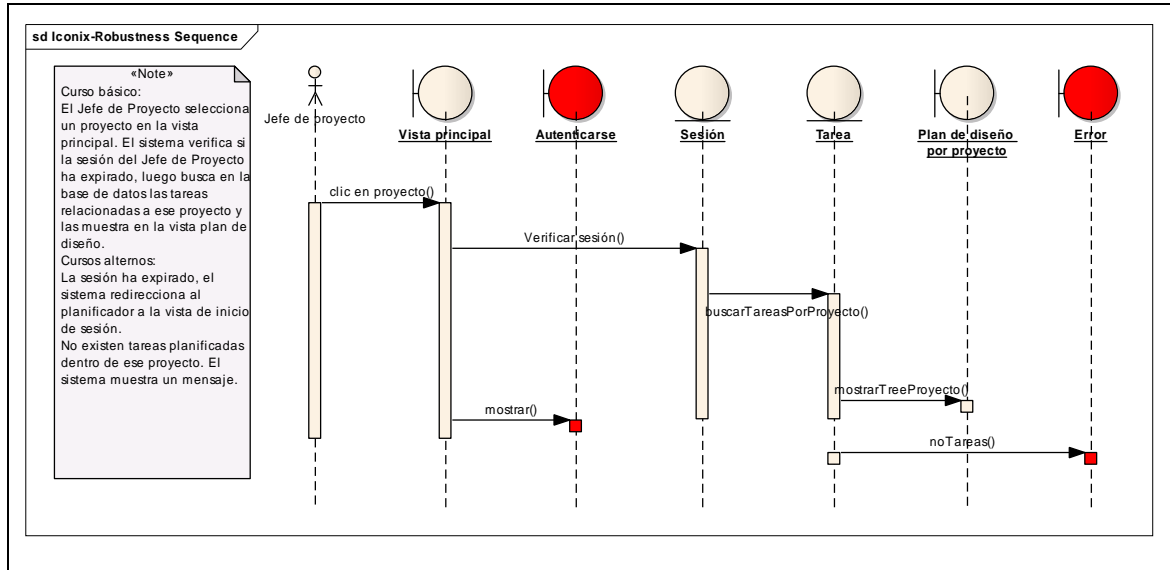
Anexo 2 Casos de uso del sistema

Caso de uso:	Visualizar tareas de proyecto
Descripción:	<p>Curso básico: El Jefe de Proyecto selecciona un proyecto en la vista principal. El sistema verifica si la sesión del Jefe de Proyecto ha expirado, luego busca en la base de datos las tareas relacionadas a ese proyecto y las muestra en la vista plan de diseño.</p> <p>Cursos alternos: La sesión ha expirado, el sistema redirecciona al planificador a la vista de inicio de sesión. No existen tareas planificadas dentro de ese proyecto. El sistema muestra un mensaje</p>

analysis Iconix-Robustness

«Note»
 Curso básico:
 El Jefe de Proyecto selecciona un proyecto en la vista principal. El sistema verifica si la sesión del Jefe de Proyecto ha expirado, luego busca en la base de datos las tareas relacionadas a ese proyecto y las muestra en la vista plan de diseño.
 Cursos alternos:
 La sesión ha expirado, el sistema redirecciona al planificador a la vista de inicio de sesión.
 No existen tareas planificadas dentro de ese proyecto. El sistema muestra un mensaje.

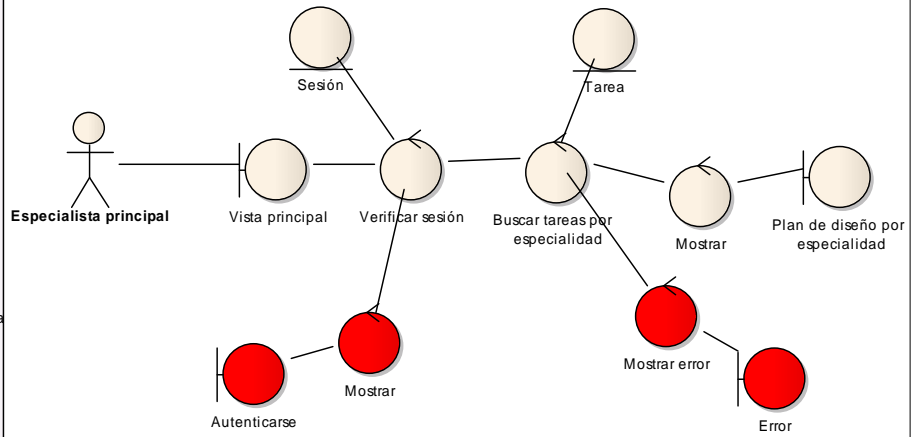




<p>Caso de uso:</p>	<p>Visualizar tareas por especialidad</p>
<p>Descripción:</p>	<p>Curso básico: El Especialista principal selecciona un proyecto en la vista principal. El sistema verifica si la sesión del Especialista principal ha expirado, luego busca en la base de datos las tareas relacionadas a esa especialidad y las muestra en la vista plan de diseño.</p> <p>Cursos alternos: La sesión ha expirado, el sistema redirecciona al Especialista principal a la vista de inicio de sesión.</p> <p>No existen tareas planificadas para la especialidad dentro de ese proyecto. El sistema muestra un mensaje</p>

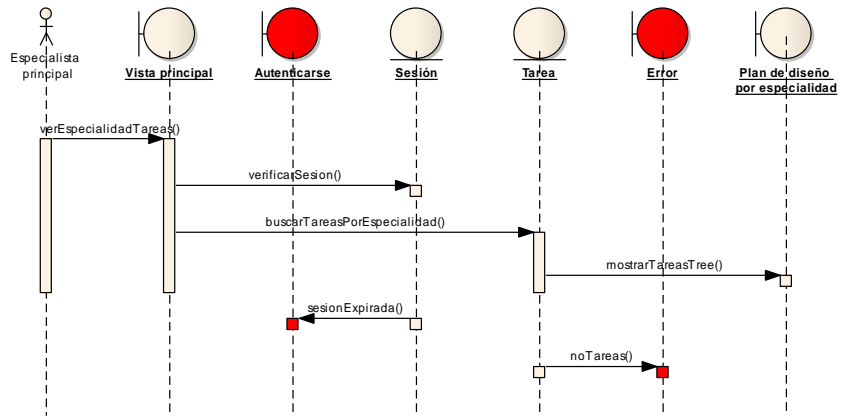
analysis Iconix-Robustness

«Note»
 Curso básico:
 El Especialista principal selecciona un proyecto en la vista principal. El sistema verifica si la sesión del Especialista principal ha expirado, luego busca en la base de datos las tareas relacionadas a esa especialidad y las muestra en la vista plan de diseño.
 Cursos alternos:
 La sesión ha expirado, el sistema redirecciona al Especialista principal a la vista de inicio de sesión.
 No existen tareas planificadas para la especialidad dentro de ese proyecto. El sistema muestra un mensaje

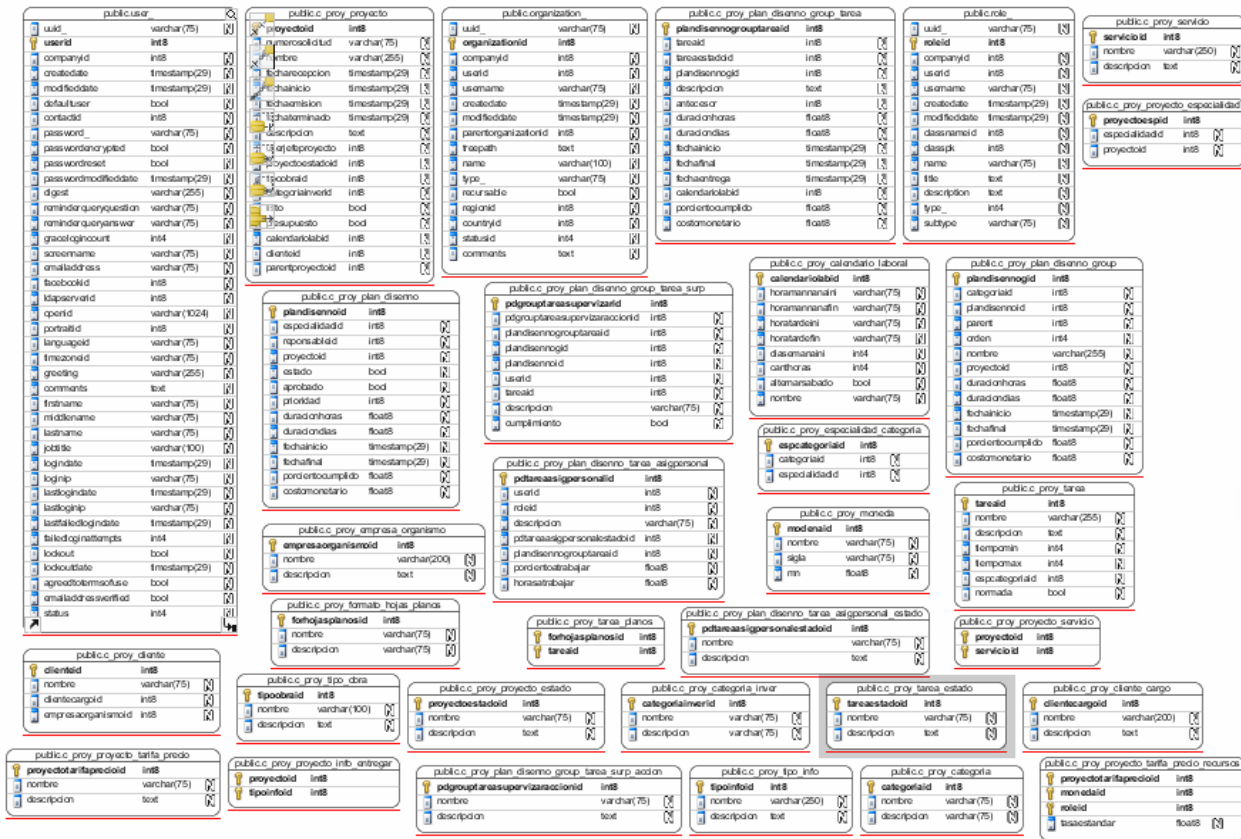


sd Iconix-Robustness Sequence

«Note»
 Curso básico:
 El Especialista principal selecciona un proyecto en la vista principal. El sistema verifica si la sesión del Especialista principal ha expirado, luego busca en la base de datos las tareas relacionadas a esa especialidad y las muestra en la vista plan de diseño.
 Cursos alternos:
 La sesión ha expirado, el sistema redirecciona al Especialista principal a la vista de inicio de sesión.
 No existen tareas planificadas para la especialidad dentro de ese proyecto. El sistema muestra un mensaje



Anexo 3 Diagrama de clases persistentes



Anexo 4 Estimación de esfuerzo basa de casos de uso

Item	Value
Total Use Cases	7
Unique Use Case Points (UUCP)	85,00
Technical Complexity (TCF)	1,07
Environmental Complexity(ECF)	0,75
Use Case Points (UUCP * TCF * ECF) = UCP	68,00
Estimated Hours per UUCP (HRS)	6,00
Total Hours (HRS * UCP)	408,00
Total Cost	12240,00

Factores de Complejidad Técnica

Metric	Description	Weight	Value	TCF
TCF01	Distributed System	2,00	5,00	10,00
TCF02	Response or throughput performance objectives	1,00	4,00	4,00
TCF03	End user efficiency (online)	1,00	2,00	2,00
TCF04	Complex internal processing	1,00	4,00	4,00
TCF05	Code must be re-usable	1,00	2,00	2,00
TCF06	Easy to install	0,50	5,00	2,50
TCF07	Easy to use	0,50	3,00	1,50

Capítulo 1: Fundamentos teóricos

TCF08	Portable	2,00	3,00	6,00
TCF09	Easy to change	1,00	3,00	3,00
TCF10	Concurrent	1,00	2,00	2,00
TCF11	Includ special security features	1,00	2,00	2,00
TCF12	Provide direct access for third parties	1,00	5,00	5,00
TCF13	Special user training faciities are required	1,00	3,00	3,00
			Total:	47,00

Factor	Value
Unadjusted TCF value (UTV)	47,00
TCF Weighting (TWF)	0,01
TCF Constant (TC)	0,60
Technical Complexity Factor (TCF) = TC + (UTV * TWF)	1,07

Factores de Complejidad Ambiental

Metric	Description	Weight	Value	TCF
ECF01	Familiar with Rational Unified Process	1,50	4,00	6,00
ECF02	Application experience	0,50	3,00	1,50
ECF03	Object-oriented experience	1,00	4,00	4,00
ECF04	Lead analyst capability	0,50	4,00	2,00
ECF05	Motivation	1,00	3,00	3,00
ECF06	Stable requirements	2,00	4,00	8,00
ECF07	Part-time workers	-1,00	0,00	-0,00
ECF08	Difficult programming language	-1,00	3,00	-3,00
			Total:	21,50

Factor	Value
Unadjusted ECF value (UEV)	21,50
ECF Weighting (EWF)	-0,03
ECF Constant (EC)	1,40
Environmental Complexity Factor (ECF) = EC + (UEV * EWF)	0,75

Anexo 5 Encuesta para la determinación del grado de competencia

Usted ha sido escogido como posible experto para ser consultado respecto al grado de utilidad del módulo de control y seguimiento a proyectos para el portal empresarial Ceproníquel, un sistema informático que favorecerá la gestión del proceso anteriormente mencionado.

Necesitamos, antes de realizarle la consulta correspondiente, como parte del método empírico de la 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 1 hasta 10.

1	2	3	4	5	6	7	8	9	10

2. Realice una autovaloración del grado de influencia de 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 6 Encuesta aplicada a expertos seleccionados

Estimado señor(a), con motivo de evaluar el módulo para el control y seguimiento a proyectos para el portal empresarial Ceproníquel, desarrollado recientemente en la Facultad de Informática y Matemática de la Universidad de Holguín, se aplica esta encuesta. Por lo que la información que brinde será decisiva para estos objetivos; rogamos que al responder estas preguntas lo haga de la manera más explícita posible. De antemano gracias.

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) en la celda que se corresponda a la categoría que usted otorga a cada aspecto referente al módulo.

Criterios	MR	BR	R	PR	NR
1. ¿Cómo evalúa la estructura organizativa de la aplicación?					
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?					
3. ¿Cómo aprecia el uso de los colores e					

imágenes?					
4. ¿La aplicación facilita el acceso a la información referente a los proyectos en la empresa?					
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?					
6. ¿Cómo evalúa la disponibilidad de los datos brindado por la aplicación?					
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?					
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?					
9. ¿Cómo evalúa la seguridad de la información?					

Anexo 7 Procesamiento de las encuestas aplicadas a los expertos

Tabla 1. Frecuencia absoluta.

Criterios	MR	BR	R	PR	NR	TOTAL
1. ¿Cómo evalúa la estructura organizativa de la aplicación?	8	5	2			15
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?	9	3	3			15
3. ¿Cómo aprecia el uso de los colores e imágenes?	6	5	4			15

4. ¿La aplicación facilita el acceso a la información referente a los proyectos en la empresa?	8	5	2			15
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?	7	5	3			15
6. ¿Cómo evalúa la disponibilidad de los datos brindado por la aplicación?	6	4	3	2		15
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?	7	4	2	2		15
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?	11	2	2			15
9. ¿Cómo evalúa la seguridad de la información?	6	6	1	2		15

Tabla 2. Frecuencia absoluta acumulada.

Crterios	MR	BR	R	PR	NR
1. ¿Cómo evalúa la estructura organizativa de la aplicación?	8	13	15	15	15
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?	9	12	15	15	15
3. ¿Cómo aprecia el uso de los colores e imágenes?	6	11	15	15	15
4. ¿La aplicación facilita el acceso a la información	8	13	15	15	15

referente a los proyectos en la empresa?					
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?	7	12	15	15	15
6. ¿Cómo evalúa la disponibilidad de los datos brindado por la aplicación?	6	10	13	15	15
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?	7	11	13	15	15
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?	11	13	15	15	15
9. ¿Cómo evalúa la seguridad de la información?	6	12	13	15	15

Tabla 3. Tabla del inverso de la frecuencia absoluta acumulada.

Criterios	MR	BR	R	PR
1. ¿Cómo evalúa la estructura organizativa de la aplicación?	0,5333	0,8667	1	1
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?	0,6	0,8	1	1
3. ¿Cómo aprecia el uso de los colores e imágenes?	0,4	0,7333	1	1
4. ¿La aplicación facilita el acceso a la información referente a los proyectos en la empresa?	0,5333	0,8667	1	1
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?	0,4667	0,8	1	1

6. ¿Cómo evalúa la disponibilidad de los datos brindado por la aplicación?	0,4	0,6667	0,8667	1
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?	0,4667	0,7333	0,8667	1
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?	0,7333	0,8667	1	1
9. ¿Cómo evalúa la seguridad de la información?	0,4	0,8	0,8667	1

Tabla 4. Tabla de determinación de los puntos de corte.

Criterios	MR	BR	R	PR	Suma	Promedio
1. ¿Cómo evalúa la estructura organizativa de la aplicación?	0,08	1,11	3,49	3,49	0	2,04
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?	0,25	0,84	3,49	3,49	8,07	2,02
3. ¿Cómo aprecia el uso de los colores e imágenes?	-0,25	0,62	3,49	3,49	7,35	1,84
4. ¿La aplicación facilita el acceso a la información referente a los proyectos en la empresa?	0,08	1,11	3,49	3,49	8,17	2,04
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?	-0,08	0,84	3,49	3,49	7,74	1,94
6. ¿Cómo evalúa la disponibilidad de los datos brindado por la	-0,25	0,43	1,11	3,49	4,78	1,2

aplicación?						
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?	-0,08	0,62	1,11	3,49	5,14	1,29
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?	0,62	1,11	3,49	3,49	8,71	2,18
9. ¿Cómo evalúa la seguridad de la información?	-0,25	0,84	1,11	3,49	5,19	1
Suma	0,12	7,52	24,27	31,41	63,32	15,55
Punto de corte	0,01	0,84	2,7	3,49	6,13	1,76

Tabla 5. Conclusiones generales.

Criterios	MR	BR	R	PR	NR
1. ¿Cómo evalúa la estructura organizativa de la aplicación?	Si	-	-	-	-
2. ¿Cómo aprecia el diseño de las interfaces de la aplicación?	Si	-	-	-	-
3. ¿Cómo aprecia el uso de los colores e imágenes?	Si	-	-	-	-
4. ¿La aplicación facilita el acceso a la información referente a los proyectos en la empresa?	Si	-	-	-	-
5. ¿Se logró, con la aplicación, mejorar el proceso de control y seguimiento a proyectos llevado a cabo actualmente en la entidad?	Si	-	-	-	-

Capítulo 1: Fundamentos teóricos

6. ¿Cómo evalúa la disponibilidad de los datos brindado por la aplicación?	-	SI	-	-	-
7. ¿Qué tan útil es la aplicación para el control de la información en este campo?	-	SI	-	-	-
8. ¿El sistema favorece aspectos vinculados al registro de nuevas solicitudes de servicios técnicos?	Si	-	-	-	-
9. ¿Cómo evalúa la seguridad de la información?	-	SI	-	-	-