Universidad de las Ciencias Informáticas

Facultad 10

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Título: Perfeccionamiento de la herramienta dotProject para viabilizar la planificación en los módulos Proyectos y Tareas

Autor:

Ernesto Camilo Acosta Labrada

Tutores:

Ing. Yanko Hernández Valdés

Ing. Yadira Morales Alamo

Ciudad de La Habana, Cuba, 2010

"Año 52 de la Revolución"

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

Aristóteles.

Declaración de Autoría

Declaramos que somos los únicos autores del trabajo “Perfeccionamiento de la herramienta dotProject para viabilizar la planificación en los módulos Proyectos y Tareas” y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2010.

Autor:

_____

Ernesto Camilo Acosta Labrada

Tutores:

_____

Ing. YankoHernández Valdés

_____

Ing. Yadira Morales Alamo

Datos de contacto.

Tutor: Yanko Hernández Valdés: Ingeniero Industrial, Instructor, graduado en el año 2005. Su dirección de correo es yanko@uci.cu.

Tutor: Yadira Morales Alamo: Ingeniera en Ciencias Informáticas, Instructor recién graduado, graduada en el año 2008. Su dirección de correo es yalamo@uci.cu.

Quiero dedicarle esta tesis ***a mi abuela*** *que esté donde esté, quisiera decirle que todo lo que hago es por ella y que le doy muchas gracias por haber existido en mi vida y que este sueño mío lo comparto con ella porque sé que hubiera sido su gran felicidad.*

A mamita*, por ser la mejor de todas las madres, apoyarme y enseñarme el camino correcto, por darme su vida sin pensarlo y malcriarme en todo momento. Te quiero con mi vida mami.*

A papito*, que es mi ejemplo a seguir como hombre, profesional y padre. Por sus consejos, por su paciencia, por ayudarme en todos los momentos y seguir conmigo hasta el final. Te quiero con mi vida papi.*

A mi hermana*, porque juntos hemos estado compartiendo las mismas cosas y apoyándonos en todo y aunque no esté aquí hoy, quisiera que se sintiera orgullosa de mi como yo me siento de ella. Por aguantar mis pesadeces y seguir queriéndome. Discúlpame los momentos malos que te he hecho pasar, nunca dudes lo mucho que te quiero.*

A mi abuelo *Acasio que siempre está a mi lado, aconsejándome, apoyándome, que lo quiero y que significa mucho para mí.*

A mi padrastro Pepe, *por ayudarme siempre que lo he necesitado y apoyarme en momentos difíciles. Por cuidarme como si fuera otro hijo y por ser parte de mi familia.*

A mi pilluelo Anaivys*, por estar a mi lado en todos los momentos, por ser la novia más linda que pueda tener un hombre. Por tratarme siempre con el cariño y amor que te caracteriza. Por compartir a mi lado cada momento de felicidad y por enseñarme que el amor existe.*

A mi familia, *por enseñarme a ser como soy e inculcar en mí los más profundos sentimientos de amor y amistad. Por entregarme con el más sincero cariño todo ese amor incomparable.*

A mis amigos *de estos cinco años que han compartido conmigo momentos difíciles: Nane, Adalberto, Javier, Jose, Hansel, Ballester, Victor, Osmel, Kendry, Camilo, Ever, Curvelo, Jose Carlos, Rolando, Héctor, Noa, Roberto, Pepe, Ernesto, George, Merallo, Manuel, Victor García, Fidel, Bradier, Magdiel, Fabián, Ramiro, Alexis, Alfredo, Juan José, bueno en fin a todo el piquete.*

A mis amigos de toda la vida, *a Reinier y Eugenio, que desde lejos comparten conmigo mis buenos y malos tiempos, que siempre me han apoyado y que cada día me demuestran su incondicional amistad.*

Agradecer *especialmente a Javier y Yadira por todo su apoyo en esta lucha por conseguir mi sueño. Por ser además muy bellas personas con las cuales compartí buenos momentos. Por ser mis amigos.*

A mis tutores *por su apoyo, dedicación, comprensión y por sobre todas las cosas por ser mis amigos. Por darme el placer de tener los mejores tutores de la UCI. Yanko, por ser más que mi amigo mi hermano.*

A todos los profesores *que han contribuido en mi formación para convertirme en un profesional.*

Resumen

Mediante el empleo de las tecnologías web se han automatizado diversos procesos en distintas esferas tanto informáticas como de cualquier otro campo. La gestión y planificación de proyectos forman parte de la alta gama de actividades inmersas en dichas automatizaciones, fundamentalmente mediante el uso de sistemas gestores de proyecto (SGP). Debido a un estudio detallado sobre el SGP dotProject se detectan algunas deficiencias en los módulos base de la herramienta (Proyectos y Tareas), que traen como consecuencia una serie de inconvenientes a los líderes y planificadores de proyectos. La presente investigación se encarga del estudio e implementación de funcionalidades de la herramienta gestora de proyectos dotProject con el objetivo de perfeccionanar el empleo de la misma. El desarrollo estuvo guiado por las especificaciones que propone la metodología RUP y el lenguaje de modelado UML, obteniendo los artefactos de los diferentes flujos de trabajo.

Para la implementación se utilizaron diferentes herramientas y lenguajes libres, cuya selección está basada teniendo en cuenta las tecnologías que se emplearon en el desarrollo original del dotProject. Para el trabajo se utilizó la herramienta IDE NetBeans, los lenguajes de programación PHP 5 y JavaScript, y el sistema gestor de bases de datos MySQL server. Una vez terminado el trabajo la herramienta posibilita una eficiente planificación de los proyectos, racionalizando el esfuerzo necesario para el desarrollo de esta actividad. Con la propuesta se contribuye a que el sistema dotProject adquiera funcionalidades que incrementan su calidad, en el sentido de un fácil manejo y eficiente planificación de proyectos.

Palabras Clave:

DotProject, Proyectos, Tareas, Planificación

ÍNDICE DE CONTENIDOS

Introducción

Disímiles han sido los problemas enfrentados y resueltos por el hombre, encaminados a garantizar procesos necesarios en diversas esferas de la vida. El desarrollo acrecentado de la humanidad y con ella la industria ha propiciado la aparición de procesos ejecutivos y productivos cada vez más complejos. Situación que ha conllevado a profundos estudios acerca de cómo planificar y controlar actividades que complementan estos procesos. Debido a que este problema se ha presentado ante todos, diversidad de autores han brindado en este campo sus criterios. Se han encontrado modelos, normas, se han escrito libros, guías pero aún así se hace difícil, tedioso y consume abundante tiempo todo el proceso de planificación y control. Los estudios no culminan en este punto, aparecen métodos mejor optimizados como la automatización de modelos y normas.

Para las empresas actualmente es común y necesario hacer uso de diversos SGP (Sistemas de Gestión de Proyectos), logrando un uso racional y controlado del tiempo y costo, garantizando la calidad de sus productos y manteniendo un mayor seguimiento de sus proyectos, lo que permite conocer realmente el estado de las secuencias de actividades o tareas que conforman los procesos. Las empresas productoras de software no exentas de esta necesidad han hecho uso de diversos SGP tales como GanttPv, GanttProject, MsProject, TeamWork, dotProject, entre otros.

La planificación cada día se hace más imprescindible debido a las crecientes dimensiones de los mercados en que se sumergen las empresas, es la vista futurista de las empresas para mejorar constantemente sus procesos y así aumentar su reputación, lograr mayores índices de ganancias y satisfacer la sociedad eficazmente.

La facultad 10 de la Universidad de las Ciencias Informáticas (UCI), como parte de una entidad formadora de ingenieros y a la vez productora de software, lleva entre sus misiones la migración al uso de software libre. Para cumplir su propósito es necesario emplear herramientas que posibiliten que el proceso de desarrollo de software sea planificado y controlado. Debido a sus características de ser completamente gratuito y de código abierto, permitir la creación de módulos e instalarlos al sistema, entre otras características que serán mencionadas en capítulos posteriores, se determinó que debía ser usado el dotProject.

DotProject, a pesar de presentar las ventajas antes mencionadas, carece de validaciones al insertar datos cuando se crean o modifican los proyectos, presenta deficiencias en cuanto a dependencias de tareas, cálculo de duración de ejecución de las tareas y tiempo asignado a un rol que realizará tareas simultáneas. Con el uso y reconocimiento de la herramienta se ha hecho necesario crear una estructura organizativa de desglose del trabajo que facilite segmentar el proyecto en tantos niveles como sean necesarios hasta llegar al mínimo de estos.

Las deficiencias presentes en la herramienta hacen engorroso el empleo de la misma, además de entorpecer la imagen del dotProject en aspectos vitales como la calidad del producto y satisfacción al cliente.

Por tanto, se plantea el ***problema científico*** mediante la interrogante: ¿Cómo perfeccionar el trabajo con la herramienta dotProject a partir de la viabilización en la planificación de los módulos Proyectos y Tareas?

Para adentrarse en lo que se pretende lograr es necesario hacer un levantamiento de la herramienta para conocerla en su totalidad, enmarcándose el ***objeto de estudio*** en la gestión de proyectos y como ***campo de acción*** la planificación de proyectos y tareas en la herramienta dotProject para proyectos productivos en la UCI.

Se plantea como ***objetivo general*** de la investigación: Perfeccionar la planificación en los módulos Proyectos y Tareas en la herramienta dotProject.

Por consiguiente, se persiguen los siguientes ***objetivos específicos*****:**

- ✓ Sistematizar el estado actual de la gestión de proyectos.
- ✓ Realizar el análisis y diseño de las funcionalidades a modificar.
- ✓ Validar las funcionalidades descritas.

Idea a defender**:** Si se perfeccionan los módulos Proyectos y Tareas de la herramienta dotProject, se facilita el trabajo con la misma, eliminándose los errores que eventualmente se presentan en su uso.

En el desarrollo de la investigación se utilizaron los siguientes métodos científicos:

Métodos teóricos:

- ✓ **Histórico-lógico**: Sirvió para analizar histórica y evolutivamente los procesos por los cuales ha pasado la gestión y planificación de proyectos dentro de la herramienta dotProject, enriqueciendo conocimientos sobre el tema.
- ✓ **Analítico-sintético**: Una vez interpretadas y analizadas las ideas plantadas en los diferentes documentos estudiados, se pueden obtener los elementos necesarios y de forma más exacta para lo que se quiere lograr.

El documento está estructurado en tres capítulos, además de las secciones de Glosario de Términos, Referencias Bibliográficas, Bibliografía y Anexos.

***Capítulo 1. Fundamentación Teórica*:** En este se definen una serie de conceptos acerca de la gestión de proyectos y de los SGP, incluyendo un estudio sobre las características de algunos SGP, centrando la investigación en la herramienta dotProject. Se describen las tecnologías, lenguajes y herramientas a utilizar en la elaboración de la solución.

***Capítulo 2. Características, análisis y diseño del sistema*:** A lo largo de este capítulo se recogen los requerimientos que deben ser modificados como propuesta de solución. Además, se precisa el análisis y el diseño de la aplicación mediante el modelado de diagramas generados por los flujos de trabajo involucrados.

***Capítulo 3. Implementación y pruebas*:** En este capítulo se lleva a cabo la implementación de la propuesta descrita en el capítulo anterior. Además, se generan y efectúan las pruebas que validan la solución eficiente al problema planteado.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 - Introducción

En este capítulo se muestran los conceptos fundamentales a tener en cuenta sobre gestión y planificación de proyectos. Se realiza un estado del arte sobre SGP con el objetivo de conocer sus características. Además, se hace un estudio general sobre la herramienta dotProject teniendo en cuenta sus funcionalidades. Se relacionan las herramientas, lenguajes y metodologías a utilizar.

1.2 - Gestión de proyectos

Antes de enfocarse en lo que es la gestión de proyectos es conveniente precisar el concepto de proyecto que se asume en la investigación.

Según el PMBOK (Guía de los Fundamentos de la Dirección de Proyectos por sus siglas en inglés “Project Management Body of Knowledge”) (1), un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único.

Las características o atributos comunes a la mayoría de los proyectos son:

- ✓ Objetivos. La naturaleza de un proyecto debe ser real y debe tener sustento.
- ✓ Presentan un calendario de actividades y deben tener un plan de trabajo.
- ✓ Son complejos, están compuestos por múltiples elementos.
- ✓ Demandan recursos tales como habilidades, conocimientos, capital y equipo de diversas áreas de una organización.
- ✓ Tienen una estructura organizacional, en la cual hay roles y responsabilidades.
- ✓ Requieren un sistema de control e información.
- ✓ Son temporales. Tiene un principio y un fin.
- ✓ Son únicos. Se inician por un único propósito.
- ✓ Requieren de elaboración progresiva. Los procesos y ciclos de vida del proyecto se inician como una idea general y se detallan conforme transcurre el proyecto.

A continuación se muestra un estudio detallado acerca de la gestión de proyectos, sus funciones y de qué forma se realiza en las diferentes empresas o compañías.

Según el PMBOK (1), la gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos establecidos. Se logra mediante la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento, control, y cierre. Se refiere a la organización y administración de recursos de manera tal que se pueda culminar todo el trabajo requerido en un proyecto dentro del alcance, el tiempo, y los costos definidos.

El primer desafío de la gestión de proyectos es asegurarse de que el proyecto sea entregado dentro de los parámetros definidos. El segundo es la asignación y la integración de las acciones necesarias para resolver esos objetivos predefinidos. El proyecto, por lo tanto, es un sistema cuidadosamente seleccionado de actividades definidas para utilizar los recursos (tiempo, dinero, recursos humanos, materiales, energía, espacio, provisiones, comunicación, calidad, riesgo, entre otros) para resolver los objetivos predefinidos.

La gestión de proyectos muchas veces es responsabilidad del gerente de proyecto, quien raramente participa de manera directa en las actividades que producen el resultado final. En vez de eso se esfuerza por mantener el progreso y la interacción mutua productiva de las varias partes de manera que el riesgo general de fracasar se disminuya.

1.3 - Planificación de proyectos

La planificación de proyectos es la organización en el tiempo de las tareas necesarias para la consecución del proyecto, tratando de aprovechar al máximo los recursos disponibles y minimizando el costo, respetando las fechas de entrega pactadas en el contrato. Debe afrontarse de manera adecuada para que al final del proyecto se pueda hablar de éxito. No se trata de una etapa independiente abordable en un momento concreto del ciclo del proyecto. O sea, no se puede hablar de un antes y un después al proceso de planificación puesto que según el avance del proyecto será necesario modificar tareas y reasignar recursos, entre otros.

Se debe tener claro que sí se puede hablar de una "etapa de planificación", llamada así porque agrupa la mayor parte de los esfuerzos para planificar todas las variables que se darán cita. Cada vez que se intenta prever un comportamiento futuro y se toman las medidas necesarias se está planificando.

Se encontraron dos grandes fases en las que la planificación cobra el máximo protagonismo. La primera es necesaria para estudiar y establecer la viabilidad de un proyecto, ya sea interno o externo a la organización. Hay que hacer los estudios necesarios con el objetivo de conocer todos los detalles posibles del proyecto y poder hacer una estimación aceptable de los recursos necesarios y los costos generados. Todo ello constituye el elemento fundamental en el que se apoya el cliente (que puede ser la propia organización en el caso de proyectos internos) para decidir sobre la realización o no del proyecto.

La segunda fase importante de planificación tiene lugar una vez decidido ejecutar el proyecto. Es el momento de realizar una planificación detallada punto por punto. Uno de los errores más importantes y graves en gestión de proyectos es querer arrancar con excesiva premura la obra, sin haber prestado la debida atención a una serie de tareas previas de preparación, organización y planificación que son imprescindibles para garantizar la calidad de la gestión y el éxito posterior.

1.4 - Sistemas de gestión de proyectos (SGP)

Para hacer más organizado y estructurado este proceso se utilizan herramientas de planificación de proyectos o sistemas de gestión de proyectos. Una herramienta gestora de proyectos es aquel software que permite llevar un control minucioso de cada uno de los proyectos de una empresa. El control implica todas las partes del trabajo, como la planificación, desarrollo y producción, así como el trato con el cliente. Este debe ser capaz de gestionar varios trabajos, ya sean internos de la empresa o contratados por clientes.

De una forma u otra un sistema de gestión de proyectos se encarga de agilizar todo el proceso de gestión y planificación de los proyectos, pues esto involucra gran cantidad de esfuerzo e información.

Según PMBOK (1), los SGP son herramientas que se orientan a la administración de forma automatizada de los recursos existentes para desarrollar un producto, cuya producción requiere de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente y para las cuales una herramienta de este tipo cuenta con funcionalidades que simplifican y humanizan el trabajo.

Dentro de cada proyecto o trabajo, deben poderse administrar aspectos tales como:

- ✓ **Proyectos:** Con descripciones, fechas de entrega, tiempos estimados, entre otros.
- ✓ **Usuarios:** Generalmente los trabajadores que están implicados en las tareas de los proyectos.

- ✓ **Tareas:** Manejando especificaciones, plazos, prioridades, planificaciones de tiempo, entre otras y además debe poder asignar recursos de la empresa (generalmente empleados) a cada tarea.

La utilización de las herramientas de gestión de proyectos no solo brinda información a quien es encargado de gestionar el proyecto si no también a cada miembro del proyecto, informándole acerca de los recursos, tareas y progreso de las actividades pendientes de cada persona. Además, debe mostrar de forma clara el tiempo y recursos que de manera general se emplearán en el desarrollo del proyecto. Una herramienta de gestión de proyectos es imprescindible para obtener una mayor organización, sobre todo en empresas de alta complejidad que requieren del manejo de un gran número de proyectos y actividades. Aunque de forma general es muy útil para cualquier grupo de trabajo del tamaño que sea.

Las herramientas de gestión de proyectos en cuanto a su interfaz se clasifican en:

- ✓ **Aplicaciones de escritorios:** Son aquellas que funcionan directamente sobre el sistema operativo que las soporta basados en modelos SDI (Single Document Interface) o MDI (Multiple Document Interface).
- ✓ **Aplicaciones Web:** Son aquellas que dependen de un servidor Web donde deben estar instaladas para poder ejecutarse correctamente. Están formadas por al menos dos capas, la parte servidora (que se ejecuta en el servidor web) y la parte cliente (se ejecuta en el navegador del usuario).

En la tabla que se presenta a continuación se recoge información del estudio realizado sobre algunos de los sistemas para la gestión de proyectos, con el objetivo de enriquecer conocimientos acerca de estas herramientas y obtener informaciones imprescindibles, que puedan servir de guía para los cambios a realizar en el dotProject.

Tabla 1. Características de los SGP.

SGP	Funcionalidades que presenta
Activecollab	✓ Gestión de calendarios de trabajo. ✓ Existencia de diferentes vistas del proyecto (Gantt, diagramas de red, de recursos, histogramas, entre otras). ✓ Gestión de tareas, con niveles de jerarquía y todas las posibilidades de dependencia necesarias. ✓ Gestión de recursos (tantos humanos como materiales) asignados a un proyecto. ✓ Gestión de costos, aunque de forma básica. ✓ Gestión de líneas base para la replanificación controlada del proyecto. ✓ Funciones de seguimiento de proyectos, ya sea por dedicación de recursos o simplemente por avance de porcentaje (introducción manual).
Ganttproject	✓ Permite importar y exportar archivos de MS Project. ✓ Soporta añadir y quitar columnas personalizadas. ✓ Puedes exportar tus proyectos a páginas HTML (Web), GanttProject usa conversión XSL. ✓ Permite trabajar con proyectos almacenados en servidores Web. Si el servidor soporta WebDAV, puedes guardar/publicar el proyecto en él. ✓ Incluye funciones personalizables. ✓ Es posible organizar tareas de forma jerárquica. ✓ Permite publicar un informe en formato PDF (por si se quiere incluir diagrama de Gantt en Internet) (2).
Microsoft Project	✓ Rápida y fácilmente de crear y administrar planes de proyecto con la nueva guía de proyectos. ✓ Nuevos asistentes se han agregado a Microsoft Project para que sea más rápido y fácil al configurar proyectos. ✓ La interfaz de Microsoft Project es una interfaz de Microsoft Office XP intuitiva y familiar.

	✓ Indicadores de etiqueta inteligente proporcionan comentarios cuando se producen ciertos cambios en el plan del proyecto. ✓ Se pueden convertir listas de tareas desde Microsoft Excel, Microsoft Outlook o Microsoft Project Web Access en planes de proyecto en Microsoft Project. ✓ Puede exportar las fechas y tareas de Microsoft Project para crear escalas de tiempo de alto nivel y diagramas de Gantt. ✓ Se han realizado una gran variedad de mejoras a vistas de datos de proyecto (3).
Planner	Es una herramienta escrita en lenguaje de programación C para gestión de proyectos, diseñada para el escritorio de GNOME. Utiliza los diagramas gantt para esquematizar las actividades del proyecto y su interacción entre éstas y otros elementos (4). Permite la descomposición en tareas y subtareas, dependencias, identificación de la ruta crítica. Entre sus funcionalidades permite definir un calendario, marcar eventos importantes, asignar recursos tanto humanos y materiales, establecer prioridades, entre otras más. En un principio fue desarrollada para GNU/Linux, pero consta de una versión beta disponible para Windows.

1.5- DotProject

Creado en el año 2000 con el fin de construir una herramienta para la gestión de proyectos. Es una aplicación multiusuario basada en Web construida por aplicaciones de código abierto. Está programada en PHP, y utiliza MySQL como sistema gestor de base de datos (aunque otros motores como Postgres también pueden ser utilizados). La plataforma recomendada para utilizar dotProject se denomina LAMP (Linux + Apache + MySQL + PHP). De todas formas, existen binarios para instalar dotProject bajo otros sistemas operativos tales como Microsoft Windows (NT, 2000, XP) y Mac.

El dotProject basa su trabajo en proveer a los usuarios una herramienta con una interfaz de usuario simple, claro y consistente. Se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela, dependiente o independientemente unas de otras. Es importante saber que las tareas independientes son tareas que normalmente se llevan a cabo en un tiempo estimado sin importar las actividades restantes del proyecto, cuando se dice paralela se refiere a las actividades que se

desarrollan simultáneamente o en un mismo período de tiempo, no siendo así en el caso de tareas dependientes, donde una tarea es condicionada por otra u otras. Existen 4 tipos de dependencias entre tareas más utilizados, aunque el dotProject solo presenta uno de estos (final a inicio):

- ✓ **Final a Inicio:** El inicio de la actividad sucesora depende de la finalización de la actividad predecesora.
- ✓ **Final a Final:** La finalización de la actividad sucesora depende de la finalización de la actividad predecesora.
- ✓ **Inicio a Inicio:** El inicio de la actividad sucesora depende del inicio de la actividad predecesora.
- ✓ **Inicio a Fin:** La finalización de la actividad sucesora depende del inicio de la actividad predecesora.

La aplicación consta de un grupo de módulos ordenados jerárquicamente los cuales permiten brindar un gran número de funcionalidades que logran la utilidad del software, entre los más importantes se encuentran:

Compañías: Son las entidades que agrupan proyectos, actividades y usuarios.

Departamentos: Son áreas dentro de las compañías que permiten agrupar usuarios en dicho nivel.

Usuarios/Contactos: DotProject tiene usuarios los cuales son capaces de entrar al sistema Web mediante el uso de un usuario y una clave secreta, y trabajar dentro del esquema de permisos que posea el rol de dicho usuario. Los contactos son usuarios especiales que asignados a un determinado proyecto pueden recibir, por ejemplo, correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema dotProject. Los usuarios y contactos pertenecen a una compañía.

Proyectos: Es la entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto.

Actividades: Son las tareas asignadas dentro de un proyecto. Son los componentes sobre los cuales se controla la duración, dependencias, recursos asignados y progreso. Las actividades deben pertenecer a un único proyecto.

Diagramas de Gantt: Permite ver en forma gráfica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.

Tickets: Para administrar todos los problemas relacionados a un proyecto.

Archivos: Permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.

Foros: Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto.

Administración del Sistema: Contiene las actividades relacionadas a la administración de usuarios, roles y configuración del sistema.

Recursos: Permite asignar recursos no humanos (oficinas, equipamiento) a un proyecto.

DotProject es una herramienta que permite gestionar las distintas tareas que componen un proyecto. Además, se perfila como una interesante herramienta para trabajar en entornos colaborativos, permitiendo a los integrantes del equipo trabajar compartiendo información relativa a los proyectos. A continuación se describen las características por las cuales se decide utilizar el dotProject para la planificación de proyectos:

Características del dotProject:

Permite la gestión y planificación de proyectos en entornos colaborativos: Basado en plataforma Web lo que posibilita la participación online de los miembros de un proyecto.

Permite la gestión de varios proyectos en el mismo ambiente y con distintas compañías: Ésta característica lo hace apto para ser implementado en oficinas de proyectos, donde se gestionan varios proyectos para distintas empresas y con muchos trabajadores en proyectos distintos o compartidos.

Permite la asignación de recursos (equipamiento, mobiliario, entre otros) a un proyecto o varios, así como la descomposición en tareas: Para la gestión del plan de cada proyecto permite la asignación de recursos tanto humanos como materiales.

- ✓ Permite clasificar y/u ordenar los proyectos en función de su estado: En curso, pendientes, cerrados, como plantilla, archivados, entre otros.
- ✓ Permite la gestión de usuarios basado en roles, permisos por proyectos y funcionalidades del sistema.

- ✓ Permite el almacenamiento de archivos (documentos o aplicaciones), versionados, por proyectos, tareas y categorías.
- ✓ Permite la gestión de tickets: Sistema que utiliza dotProject como ayuda, donde se puede tener un log de problemas o pedidos de los usuarios del sistema.
- ✓ Permite la configuración de una agenda de contactos: Cada contacto puede estar vinculado a un usuario de algún proyecto.

Permite vista de eventos y tareas en calendario, filtrado por:

- ✓ Estado actividad.
- ✓ Proyecto.
- ✓ Empresa.

Permite la visualización de informes y estadísticas sobre los proyectos registrados, como por ejemplo:

- ✓ Las horas asignadas (por usuario o proyecto) para un periodo de tiempo.
- ✓ Las horas asignadas y las realmente utilizadas, para poder extraer porcentajes de trabajos realizados y porcentajes de eficiencia en base a tareas completadas.
- ✓ Estado de un proyecto: Tareas completas, tareas que sufren desviaciones, entre otras.

Contraste de las características de la aplicación dotProject con las de algunas áreas de conocimientos del Project Management (5).

Administración del Alcance del Proyecto

El sistema permite asignar la visibilidad de las tareas, por lo tanto las actividades de más alto nivel (hitos) serán vistas por los clientes o gerentes y las tareas operativas o propias de desarrollo vistas por los desarrolladores y el administrador del proyecto. De igual forma se implementa la gestión de costos generales del proyecto.

Administración de Tiempo del Proyecto

El sistema permite asignar estimación de esfuerzos (en horas/días) por tarea, como también el calendario de cada tarea, asignar dependencias y recursos afectados (con el porcentaje de esfuerzo de ese recurso). Cada tarea puede ser actualizada según su estado (activa/inactiva) y progreso en porcentaje. Una tarea puede ser un hito (importante para establecer objetivos - gestión del alcance). El sistema implementa un grafico gantt resumen del plan por proyecto, por usuario o por departamento, lo que es muy ventajoso ya que cada integrante del proyecto visualiza sus tareas dentro del proyecto o cómo éstas afectan o se ven afectadas por otras tareas, además de visualizar la carga de trabajo de los departamentos de la organización que colabora en el proyecto. A cada tarea se le pueden adjuntar archivos (documentos o aplicaciones), esta característica podría ser muy significativa en hitos o entregables. Cada tarea puede ser actualizada mediante un log de novedades, estas novedades incluyen esfuerzos reales, redefinición de necesidades, re-planificación y los costos (codificados) asumidos por esas razones. Mediante este log es posible tener una importante base de datos para futuras estimaciones en proyectos similares, lo que repercute en un aumento de la efectividad de esas estimaciones.

Administración de Costos del Proyecto

Mediante la asignación de costos por tareas es posible estimar costos del proyecto, además de registrar los costos reales durante el desarrollo del proyecto (según progreso de la tarea) mediante un log por tareas; por lo mencionado anteriormente, es muy ventajoso tener un historial pues permite realizar análisis.

Administración de Recursos Humanos del Proyecto

Es posible mediante la gestión de usuarios, administrar básicamente los recursos humanos, otorgándole el rol que corresponda, con esto se tiene un seguimiento del esfuerzo por persona. En la configuración en cada tarea se especifica el porcentaje asignado del recurso humano (usuario) y la duración de la tarea, con lo cual se puede conocer en el calendario el trabajo de cada usuario y el horario en que trabaja en esa tarea ya que se especifica horario de inicio y fin. Dado que cada usuario puede pertenecer a un departamento de una empresa, también se tienen los esfuerzos por departamento y por compañía, para un proyecto o para los proyectos que lleve adelante el departamento y la empresa.

Administración de las Comunicaciones del Proyecto

Correo electrónico:

Todos los usuarios de cada proyecto están comunicados mediante correo electrónico, pues es un atributo obligatorio para cada usuario. Las notificaciones que pueden incluirse son asignaciones de tareas, vencimientos de plazos, log de notificaciones, tickets por proyectos, entre otros.

Foros de Discusión:

Cada proyecto da la posibilidad de habilitar varios foros de discusión donde se pueden plantear temas a fines de cada etapa o tarea, la amplitud de posibilidades da la flexibilidad de una comunicación abierta y fluida en éste sentido.

Tickets:

El administrador de un proyecto particular o el administrador del sistema, tiene la posibilidad de ver un registro de pedidos sobre problemas, dudas o sugerencias respecto del proyecto o del sistema. Cada ticket tiene un estado con lo que ayuda a tener un seguimiento de pedidos no resueltos, pendientes o cerrados.

1.6 - Herramientas, lenguajes y tecnologías a utilizar

1.6.1 - Lenguajes de programación

Para el desarrollo del trabajo se utilizaron los mismos lenguajes de programación empleados en la implementación del dotProject, los cuales son PHP 5 y JavaScript, esto permite entre otras cosas la reutilización de código. Además, se empleó el lenguaje de modelado UML para confeccionar cada uno de los diagramas que comprende el desarrollo de software. A continuación se describen algunas de las características de los lenguajes mencionados.

PHP 5

PHP (acrónimo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular, diseñado originalmente por Rasmus Lerdorf en 1994, especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. Puede ser desplegado en la mayoría de los servidores web y plataformas sin costo alguno.

Está disponible en casi todos los sistemas operativos del mercado, entre estos Linux/Unix, HP-UX, Solaris, OpenBSD, Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. Soporta ODBC (el Estándar Abierto de Conexión con Bases de Datos), así que puede conectarse a cualquier base de datos que soporte tal estándar.

Teniendo en cuenta las características mencionadas, y conociendo que la herramienta dotProject está programada en PHP 5, se determinó emplear dicho lenguaje para la programación.

JavaScript

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript van generalmente incrustados en los documentos HTML y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos, entre otros (6).

Se utiliza JavaScript para validar las deficiencias que sean de menor complejidad y no necesiten de una implementación más compleja con PHP 5.

Lenguaje de modelado UML

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80s y principios de los 90s. El UML fusiona los conceptos de la orientación a objetos aportados por Booch, OMT y OOSE (Booch, G. et al., 1999). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos, incluido el modelado de sistemas distribuidos y concurrentes, para asegurar que funcionen adecuadamente estos dominios.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. UML posee formas de modelar conceptos como son los procesos de negocio y funciones de sistema, además de aspectos concretos como escribir clases en un lenguaje determinado, esquemas de bases de datos y componentes de software reusables.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere

discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no el proceso que se siguió para obtenerlo (7).

1.6.2 - Herramientas a utilizar

Las herramientas que se muestran a continuación forman parte de la plataforma LAMP (Linux + Apache + MySQL + PHP) recomendada para la instalación del dotProject. Para un mejor conocimiento se mencionan algunas de las características del servidor de páginas web Apache y del sistema gestor de bases de datos MySQL. Se describe además la Herramienta CASE: Visual Paradigm para el modelado de diagramas.

Apache HTTP Server

Es un servidor de páginas web, programa que permite publicar páginas web alojadas en un ordenador. Es el servidor http más usado en el mundo directa o indirectamente. El servidor Apache es empleado por múltiples razones, como son la seguridad, fiabilidad, disponibilidad, facilidad de instalación, pocos recursos de hardware necesarios, disponibilidad del código fuente, entre otras. Está programado en C y fue el generador de uno de los principales sitios de proyectos de código abierto (www.apache.org) que existen actualmente. Dispone de versiones para los principales sistemas operativos y está disponible por defecto en todas las distribuciones GNU/Linux que existen hoy.

Algunas características:

- ✓ Apache es un software gratuito y libre.
- ✓ Es altamente configurable y de diseño modular. Es muy sencillo ampliar sus capacidades. Actualmente existen muchos módulos y están disponibles para toda la comunidad. Otro aspecto importante es que cualquier persona que posea una experiencia avanzada en la programación en C o Perl puede escribir un módulo para realizar una función determinada.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- ✓ Tiene una alta configuración en la creación y gestión de logs. Apache permite la creación de ficheros de log a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la licencia GNU/GPL, este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- ✓ Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, entre otros.).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y contraseñas, manteniendo un buen nivel de seguridad en los datos.

NetBeans

Este es un entorno de desarrollo integrado, desarrollado y programado en Java por la empresa Sun Microsystems, actualmente propiedad de Oracle. Inicialmente elaborado para facilitar el desarrollo de aplicaciones en lenguaje Java. Actualmente ha evolucionado para soportar otros lenguajes de amplio uso en el mundo como PHP, XML y HTML. Al soportar estos y tener facilidades como el completamiento de código y otras como un editor de páginas HTML, humaniza un poco el trabajo del programador y garantiza un mejor acabado del sistema.

Herramienta CASE Visual Paradigm

Visual Paradigm utiliza el UML como lenguaje de modelado. Está disponible en varias ediciones: Enterprise, Profesional, Community, Standard, Modeler y Personal, cada una destinada a necesidades específicas. Es una herramienta de gran alcance, fácil de utilizar y que apoya el ciclo de vida completo del software (análisis, diseño, codificación, prueba y despliegue), además permite dibujar diagramas UML, generar código de diagramas de clases y viceversa. Genera documentación y soporta todos los diagramas de la más reciente versión de UML.

Fue seleccionada la edición Community de Visual Paradigm que es gratuita y soporta la versión 2.0 de UML. Esta edición soporta el ciclo de vida completo del desarrollo de software. Además, ayuda a una más rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

1.6.3 - Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado.

PMBOK

PMBOK es un estándar en la gestión de proyectos desarrollado por el PMI. Se encuentra disponible en 11 idiomas: inglés, español, chino simplificado, ruso, coreano, japonés, italiano, alemán, francés, portugués de Brasil y árabe. En 1987, el PMI publicó la primera edición del PMBOK en un intento por documentar y estandarizar la información y prácticas generalmente aceptadas en la gestión de proyectos. La edición actual (la tercera) provee de referencias básicas a cualquiera que esté interesado en la gestión de proyectos. Posee un léxico común y una estructura consistente para el campo de la gestión de proyectos. La Guía del PMBOK es ampliamente aceptada por ser el estándar en la gestión de proyectos.

Metodología SXP

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva, fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de manera eficiente y de tener siempre medidos los progresos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Rational Unified Process (RUP)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

Como RUP es un proceso, en su modelación define como sus principales elementos:

Trabajadores (“quién”)	Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
Actividades (“cómo”)	Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
Artefactos (”qué”)	Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
Flujo de actividades (“Cuándo”)	Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

Está compuesto por 4 fases:

Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Elaboración: En esta etapa el objetivo es determinar la arquitectura base del sistema.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios release del producto que han pasado las pruebas. Se ponen estos release a consideración de un conjunto de usuarios.

Transición: El release ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores, cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Debido a que RUP es una metodología de desarrollo robusta, adaptable a cualquier proyecto, presenta gran cantidad de documentación tanto en la universidad como en internet y teniendo en cuenta las características antes mencionadas, se decide utilizar RUP.

1.7 - Conclusiones

- ✓ Se realizó un estado del arte sobre gestión y planificación de proyectos.
- ✓ Se estudiaron algunos de los SGP, donde se muestran sus funcionalidades y características fundamentales, lo que brinda información imprescindible a tener en cuenta para los cambios que se llevarán a cabo en la herramienta dotProject.
- ✓ Se hizo un estudio detallado sobre la herramienta dotProject que arroja como resultado un conjunto de características que servirán de apoyo para el entendimiento de la herramienta y la interacción con la misma.
- ✓ Se concluyó que el sistema se desarrollará utilizando RUP como metodología de desarrollo de software y sobre las tecnologías utilizadas en la creación original del dotProject:
    - Gestor de base de datos: MySQL.
    - Servidor de páginas web: Apache HTTP Server con soporte para PHP.
    - IDE de desarrollo: Netbeans.
    - Lenguajes de programación: PHP y JavaScript.
- ✓ Para apoyar el proceso de implementación del software se utilizará la herramienta de desarrollo Visual Paradigm.

Capítulo 2. Características, análisis y diseño del sistema

2.1 - Introducción

Para dar solución al problema científico planteado se precisa conocer las diferentes características del sistema, centrándose en aquellas que deben ser modificadas. Se realiza una descripción del dominio así como el levantamiento de requisitos, donde se muestran detalladamente las mejoras que se aplicarán en el producto. Además, se realiza el análisis y el diseño de la aplicación donde se modelan los casos de uso seleccionados para garantizar las soluciones propuestas. Se definen los diferentes diagramas generados en estos flujos de trabajo, para así lograr una mayor comprensión del funcionamiento de la herramienta.

2.2 - Descripción del dominio

Debido a la relativa simplicidad del entorno donde está enmarcado el sistema y el conocimiento que se posee acerca de su funcionamiento, no es necesario realizar un Modelo de Negocio completo para comprender la problemática que ha de resolverse, siendo suficiente el Modelo de Dominio. Este último captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, y se considera en RUP un subconjunto del llamado modelo de objetos del negocio.

El modelo de dominio no es más que una representación visual de los conceptos u objetos significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de componentes de software (8).

Conceptos fundamentales:

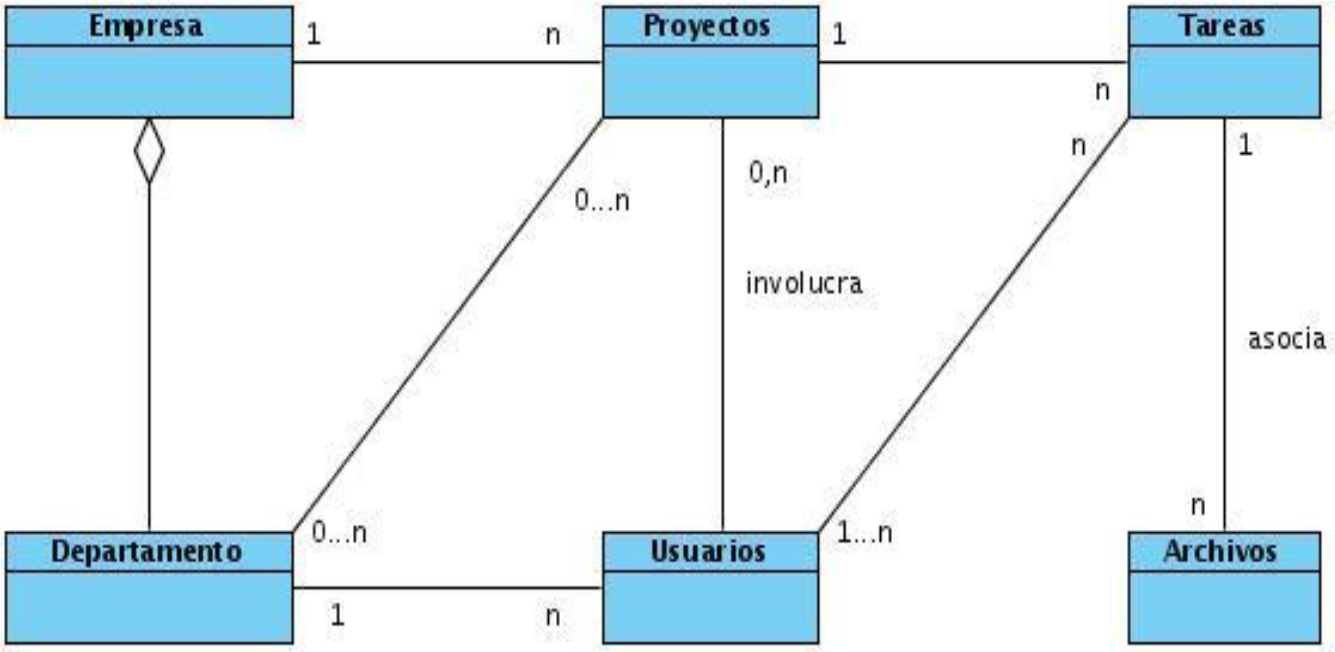
Proyecto: Es un esfuerzo temporal que se lleva a cabo para obtener un servicio, producto o resultado único.

Tarea: Un término que reemplaza a trabajo, cuyo significado y ubicación dentro de un plan estructurado para un trabajo del proyecto varía de acuerdo con el área de aplicación, industria y marca del software de gestión de proyectos.

Usuario: La persona u organización que usará el producto o servicio del proyecto (1).

Planificador de proyecto: Es el encargado de realizar el cronograma de tareas así como las estimaciones de tiempo y recursos en las tareas.

Modelo de dominio dotProject



2.3 - Modelo del sistema

2.3.1 - Requisitos del sistema

Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Los requisitos se clasifican en funcionales y no funcionales.

Requisitos funcionales

Son capacidades o condiciones que el sistema debe cumplir. Los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. A continuación se muestra una tabla donde se explican cada uno de los requisitos funcionales trazados teniendo en cuenta los cambios que se van a realizar.

Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

En la siguiente tabla se recogen los requísitos funcionales

Tabla 2. Requísitos funcionales

Requisitos Funcionales	**Descripción detallada de cambios en la herramienta**
RF - 1. Agregar proyecto.	El sistema debe adicionar proyectos teniendo en cuenta los requisitos funcionales 1.1 y 1.2.
RF - 1.1 Verificar fechas válidas en el proyecto.	Actualmente en la herramienta un proyecto puede adicionarse con los campos de las fechas de forma incorrecta, se validará esta funcionalidad mostrando un error en dicho caso. ***Anexo 1.***
RF - 1.2 Asignar proyecto padre.	La herramienta permite la parcelación de un proyecto en tareas, restringiéndose solo a estos dos niveles (proyectos y tareas) lo que impide un desglose más amplio del trabajo, es por ello que se agrega la opción de "Asignar proyecto padre" para brindar los niveles que se necesiten, ya sea proyecto, sub-proyecto 1…n, tareas. ***Anexo 2.***
RF - 2. Modificar proyecto.	El sistema debe modificar las fechas de los proyectos teniendo en cuenta el requisito 2.1.
RF - 2.1 Editar fecha del proyecto.	Actualmente en la herramienta cuando se edita la fecha de inicio de un proyecto las tareas previamente planificadas se mantienen con la fecha antigua, lo que provoca un desfase de estas respecto a la fecha de inicio del proyecto. Debido a esta deficiencia se plantea realizar cambios en la funcionalidad de forma que las tareas de un proyecto se trasladen a la fecha estimada para inicio del proyecto siempre y cuando haya una modificación.
RF - 4. Agregar tarea.	El sistema debe agregar tareas teniendo en cuenta los requisitos funcionales 4.1, 4.2, 4.3 y 4.4.
RF - 4.1 Calcular duración de tareas.	En la herramienta el campo duración de una tarea sale con valor número 1 por defecto obligando al planificador a oprimir el botón "Duración" lo que calcula y actualiza dicho valor. Se propone optimizar

	esta funcionalidad eliminando el botón mencionado y realizar el cálculo de manera automática una vez llenados los campos de fecha de inicio y finalización de la tarea. ***Anexo 3.***
RF - 4.2 Establecer dependencias de tareas.	Cuando se establecen dependencias de fin a inicio, la tarea dependiente no debe comenzar antes de la previa culminación de la tarea predecesora, actualmente la herramienta ejecuta de forma errónea. Se plantea validar la funcionalidad de manera que las tareas se acoplen a las dependencias asignadas como es debido. ***Anexo 4.***
RF - 4.3 Asignar recursos a las tareas.	En la herramienta es posible asignar recursos humanos ocupando el 100 % de su tiempo disponible en varias tareas que ocurren de forma simultánea. Se propone validar que no sea posible realizar dicha planificación a no ser que sea compartido el por ciento disponible del usuario en las tareas concurrentes que debe cumplir. ***Anexo 5.***

Seguidamente se hace una descripción de los requísitos no funcionales.

Requisitos de Software:

Para el servidor web donde estará instalado el sistema debe estar habilitado con el sistema operativo Linux, servidor web se recomienda Apache 1.3.27 o superior, gestor de base de datos MySQL 3.23.51. Los nodos clientes deberán acceder al sistema mediante algún navegador web (se recomienda Mozilla).

Requisitos de Hardware:

- ✓ Memoria RAM 256 MB.
- ✓ Servidor a 250 MHz.
- ✓ Capacidad de disco duro 8 GB.

Restricciones en el diseño y la implementación:

- ✓ Lenguaje de programaciónse utilizara PHP 4.1.x o superior, Java Script.
- ✓ Para la implementación NetBeans 6.8.

- ✓ Mantener la arquitectura que propone el dotProject.

Requisitos de apariencia o interfaz externa:

Los nuevos cambios que se realicen en las interfaces deberán mantener el mismo estándar de diseño del dotProject.

Requisitos de Seguridad:

El local donde estará el servidor que contiene al sistema debe regirse por las políticas de acceso de la entidad.

Requisitos de Usabilidad:

- ✓ En la implementación de los cambios al dotProject se deben manejar los errores de tal forma que se informe al usuario de un suceso adverso.
- ✓ Aptitud para la tarea: los diseñadores deben ofrecer exactamente la información que el usuario necesita.

Requisitos de Soporte:

Capacitar a los clientes que usarán esta nueva versión del dotProject sobre las modificaciones que se realizaron en el mismo.

Actor del sistema

Un actor es quién interactúa con el sistema, puede ser cualquier individuo, grupo, organización o máquina que utilice el sistema para él beneficiarse de sus resultados. Todo caso de uso es inicializado por un actor, un caso de uso define la manera en que se comportará el sistema. Se define como actor del sistema al planificador de proyectos, que es el responsable de crear y modificar los proyectos y tareas, así como asignar duración, dependencias y recursos humanos a las actividades.

2.3.2 - Patrones de casos de uso

¿Qué es un patrón?

“Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución de ese problema, de tal modo que se pueda aplicar esa solución un millón de veces, sin hacer lo mismo dos veces” (9).

En general, un patrón está compuesto por 4 elementos esenciales:

1. **Nombre**: Es lo que se usa para describir el problema, sus soluciones y consecuencias en una palabra o dos, por lo que definir un buen nombre es algo esencial.
2. **El problema**: Describe cuándo hay que aplicar el patrón, explicando lo que sucede y el contexto en el que se desarrolla y especifica condiciones, si existieran, que deben cumplirse para que tenga sentido aplicar el patrón seleccionado.
3. **La solución**: Especifica los elementos que se usan para resolver el problema, su orden de uso, responsabilidades y colaboraciones que se establecen entre estos.
4. **Las consecuencias**: Son los resultados de aplicar el patrón, por lo que son de especial relevancia para evaluar las diferentes alternativas y para entender la relación costo/beneficio de la aplicación del patrón.

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Con estos patrones se pueden lograr mejores resultados y de forma más rápida. Entre los muchos patrones existentes relacionados con los casos de uso se pueden enumerar los siguientes:

- Reglas de negocio
- Concordancia (Commonality)
- Componente jerárquico (Component hierarchy)
- Extensión concreta o Inclusión (CRUD)
- Caso de uso grande (Large Use case)
- Sistema de Capas
- Múltiples actores
- Servicio opcional
- Vistas ortogonales
- Inicio de sesión
- Secuencia de casos de uso.

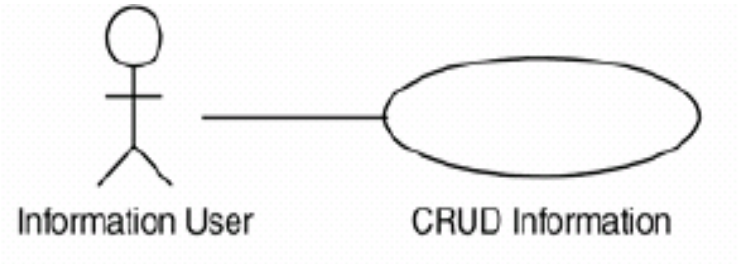
El patrón CRUD es el de vital importancia en este caso ya que permite trabajar con varios casos de usos. Para que se tenga una mejor comprensión de esto se define a continuación:

CRUD (Creating, Reading, Updating, Deleting)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

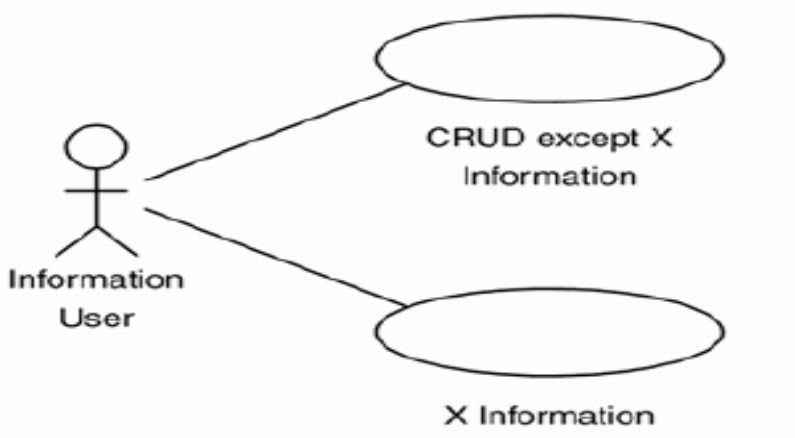
CRUD Completo

Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples (10).



CRUD Parcial

Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras. En este caso es conveniente el uso del patrón CRUD Parcial debido a que el trabajo modelará solo las operaciones de adicionar y modificar, para ello se nombra de esta forma: Administrar información (10).



2.3.3 - Listado de casos de uso del sistema

Administrar proyectos

CU-1	Administrar proyectos.
Actor	Planificador de proyecto.
Descripción	El planificador de proyecto es encargado de adicionar y modificar los proyectos.
Referencia	RF 1, RF 1.1, RF 1.2, RF 2, RF 2.1

Administrar tareas

CU-2	Administrar tareas.
Actor	Planificador de proyecto.
Descripción	El planificador de proyecto es encargado de adicionar y modificar tareas.
Referencia	RF 4, RF 4.1, RF 4.2, RF 4.3

Diagrama de casos de uso del sistema

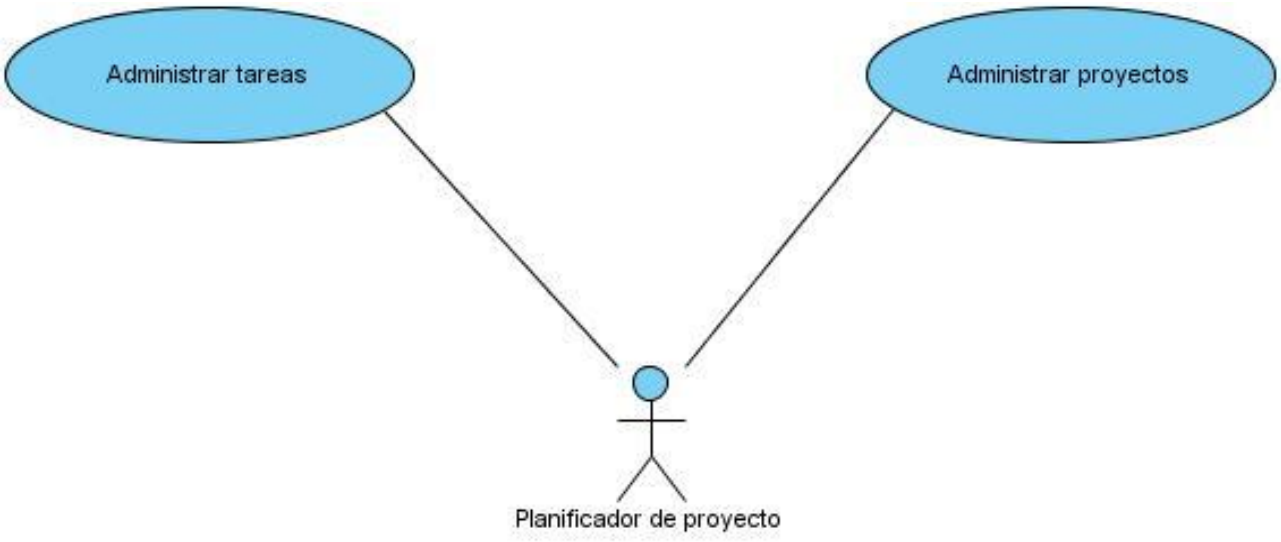


Tabla 3. Descripción detallada de casos de uso del sistema. Caso de uso Administrar proyectos.

CU-1	Administrar proyectos
Actor	Planificador del proyecto

<table>
<tr><td>Descripción</td><td colspan="2">El caso de uso se inicia cuando el planificador del proyecto desea realizar una de las siguientes operaciones:

Insertar proyecto: El planificador del proyecto introduce los datos del proyecto que desea adicionar, el sistema valida que los datos sean correctos para que el proyecto sea adicionado con éxito, finalizando así el caso de uso.

Modificar proyecto: El planificador del proyecto selecciona un proyecto de la lista de proyectos, el sistema muestra los datos del mismo permitiendo modificarlos, finalizando así el caso de uso.</td></tr>
<tr><td>Referencia</td><td colspan="2">RF 1, RF 1.1, RF 1.2, RF 2, RF 2.1</td></tr>
<tr><td>Precondiciones</td><td colspan="2">El planificador del proyecto debe estar autenticado con el rol de administrador en el sistema dotProject.</td></tr>
<tr><td colspan="3">Curso Normal de los Eventos</td></tr>
<tr><td colspan="2">Acciones del Actor</td><td>Respuesta de la herramienta</td></tr>
<tr><td colspan="2">1. El planificador del proyecto puede realizar las siguientes opciones.

a) Insertar Proyecto

b) Modificar Proyecto

Si el planificador escoge la opción Insertar Proyecto ir a la sección “Insertar Proyecto”, si escoge la opción Modificar Proyecto ir a la sección “Modificar Proyecto.”</td><td></td></tr>
<tr><td colspan="3">Sección “Insertar”: Flujo Normal de Eventos</td></tr>
<tr><td colspan="2">Acciones del Actor</td><td>Respuesta del Sistema</td></tr>
</table>

1. El planificador del proyecto se autentica en el dotProject.	2. El sistema comprueba el rol que desempeña en el dotProject.
3. El planificador del proyecto selecciona el módulo de Compañías.	4. El sistema muestra las opciones del módulo Compañía.
5. El planificador del proyecto oprime el botón “Nuevo Proyecto”.	6. El sistema muestra la interfaz para adicionar un nuevo proyecto.
7. El planificador del proyecto incorpora los datos necesarios para la introducción del proyecto, seleccionando o no un proyecto padre.	8. El sistema verifica si el proyecto existe.
	9. El sistema verifica que los datos estén correctos y genera un proyecto padre si previamente fue seleccionado.
	10. El sistema registra el nuevo proyecto, terminando así el caso de uso.
Sección “Insertar”: Flujo Alternativo	
	2.1. Si el usuario no está registrado en la base de datos del dotProject o no tiene el rol indicado para utilizar este recurso no podrá incorporar nuevos proyectos.
	8.1 Si el proyecto existe, el sistema muestra un mensaje de error.
	9.1 En caso de que no sean válidas las fechas se mostrará el mensaje ¨Fechas incorrectas¨.
Sección “Modificar”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El planificador del proyecto se autentica en el dotProject.	2. El sistema comprueba el rol que desempeña en el dotProject.

3. El planificador del proyecto selecciona el módulo de Proyectos.	4. El sistema muestra el listado de proyectos y las opciones del módulo Proyectos.
5. El planificador selecciona el proyecto que desea modificar y oprime la pestaña ¨Editar este proyecto¨.	6. El sistema muestra la interfaz para modificar proyecto.
7. El planificador del proyecto cambia los datos necesarios del proyecto y oprime el botón ¨Salvar¨.	8. El sistema verifica que los datos estén correctamente.
	9. El sistema establece los nuevos datos al proyecto cambiando si es necesario las tareas que comprende.
	10. El sistema registra el nuevo proyecto, terminando así el caso de uso.
Sección “Modificar”: Flujo Alternativo	
	2.1 Si el usuario no está registrado en la base de datos del dotProject o no tiene el rol indicado para utilizar este recurso no podrá incorporar nuevos proyectos.
	8.1 En caso de que no sean válidas las fechas se mostrará el mensaje “Fechas incorrectas”.
	9.1 En caso de que se cambie la fecha de inicio del proyecto se establecen nuevas fechas a sus tareas, teniendo en cuenta las modificaciones realizadas.

Caso de uso Administrar tareas.

CU-2	Administrar tareas
Actor	Planificador del proyecto
Descripción	El caso de uso se inicia cuando el planificador del proyecto desea realizar una de las siguientes operaciones:

	Insertar tarea: El planificador del proyecto introduce los datos de la tarea que desea adicionar, el sistema valida que los datos sean correctos para que la tarea se adicione con éxito, finalizando así el caso de uso. Modificar tarea: El planificador del proyecto selecciona una tarea de la lista de tareas, el sistema muestra los datos de dicha tarea permitiendo modificar los datos, finalizando así el caso de uso.
Referencia	RF 4, RF 4.1, RF 4.2, RF 4.3
Precondiciones	El planificador de proyecto debe estar autenticado con el rol de administrador en el dotProject.

Curso Normal de los Eventos	
Acciones del Actor	Respuesta de la herramienta
2. El planificador del proyecto puede realizar las siguientes opciones. c) Insertar tarea. d) Modificar tarea. Si el planificador escoge la opción Insertar tarea ir a la sección “Insertar tarea”, si escoge la opción Modificar tarea ir a la sección “Modificar tarea.”	

Sección “Insertar”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El planificador del proyecto se autentica en el dotProject.	2. El sistema comprueba el rol que desempeña en el dotProject.
3. El planificador del proyecto selecciona el módulo Tareas.	4. El sistema muestra las opciones del módulo Tareas.
5. El planificador del proyecto oprime el botón ¨Nueva Tarea¨.	6. El sistema muestra la interfaz para agregar tarea.

7. El planificador del proyecto incorpora los datos necesarios para la introducción de la tarea, seleccionando o no una tarea padre.	8. El sistema verifica si la tarea existe.
	9. El sistema verifica que los datos estén correctamente llenos y genera el campo duración de la tarea.
	10. El sistema registra la nueva tarea, terminando así el caso de uso.
Sección “Insertar”: Flujo Alternativo	
	1.1 Si el usuario no está registrado en la base de datos del dotProject o no tiene el rol indicado para utilizar este recurso no podrá incorporar nuevas tareas.
	8.1 En caso de existir la tarea se muestra un mensaje de error.
	9.1 En caso de que no sean válidas las fechas se mostrará el mensaje ¨Fechas incorrectas¨.
Sección “Modificar”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El planificador del proyecto se autentica en el dotProject.	2. El sistema comprueba el rol que desempeña en el dotProject.
3. El planificador del proyecto selecciona el módulo de Tareas.	4. El sistema muestra el listado de tareas y las opciones del módulo Tareas.
5. El planificador selecciona la tarea que desea modificar y oprime la pestaña ¨Editar esta tarea¨.	6. El sistema muestra la interfaz para modificar tarea.
7. El planificador del proyecto cambia	8. El sistema verifica que los datos estén

los datos necesarios de la tarea.	correctamente llenos.
	9. El sistema establece los nuevos datos a la tarea.
	10. El sistema actualiza la tarea, terminando así el caso de uso.
Sección “Modificar”: Flujo Alternativo	
	1.1 Si el usuario no está registrado en la base de datos del dotProject o no tiene el rol indicado para utilizar este recurso no podrá incorporar nuevos proyectos.
	8.1 En caso de que no sean válidas las fechas se mostrará un mensaje de error especificando el problema.

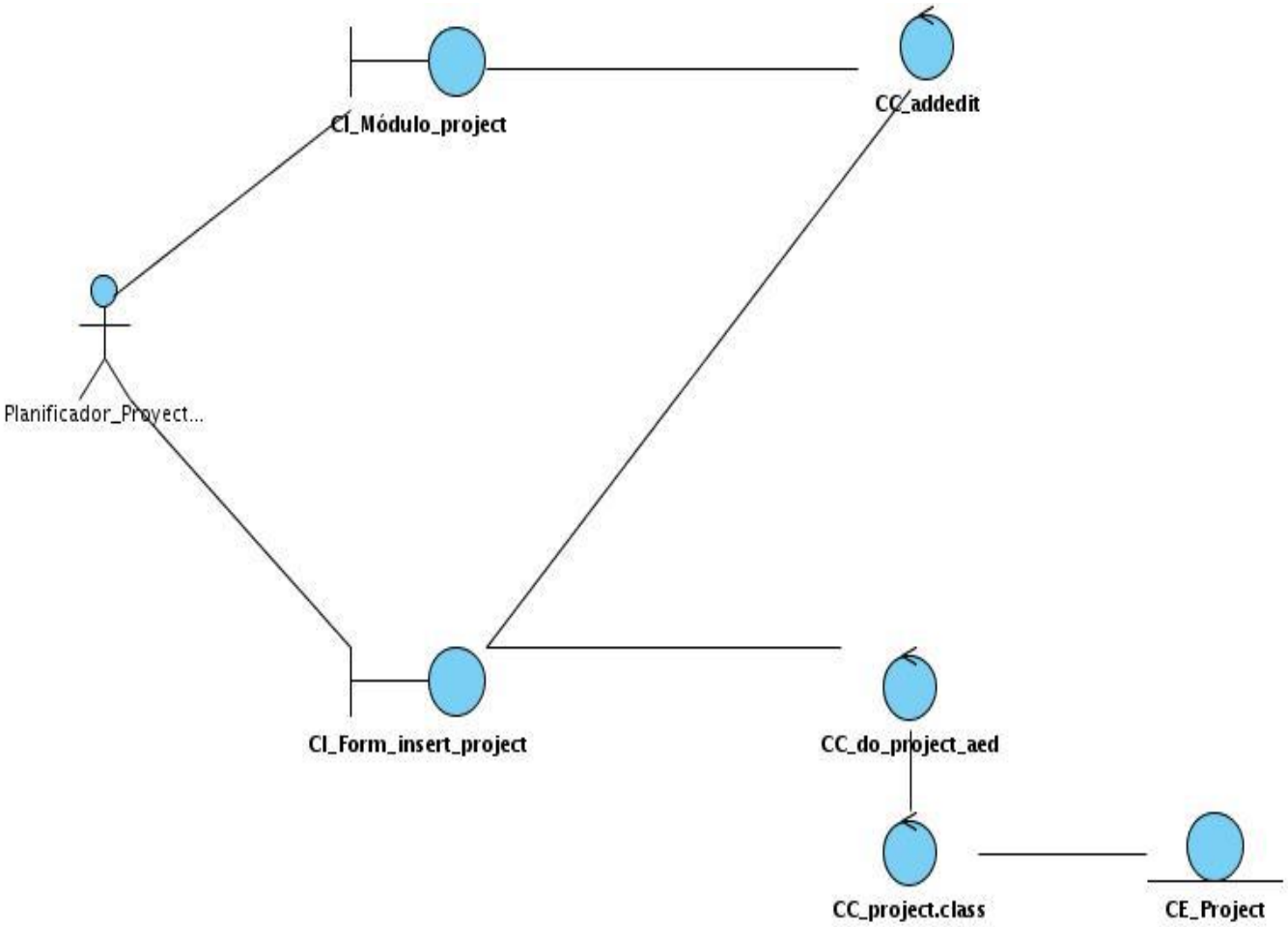
2.4 Análisis del Sistema

Durante el análisis se examinan los requerimientos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema completo.

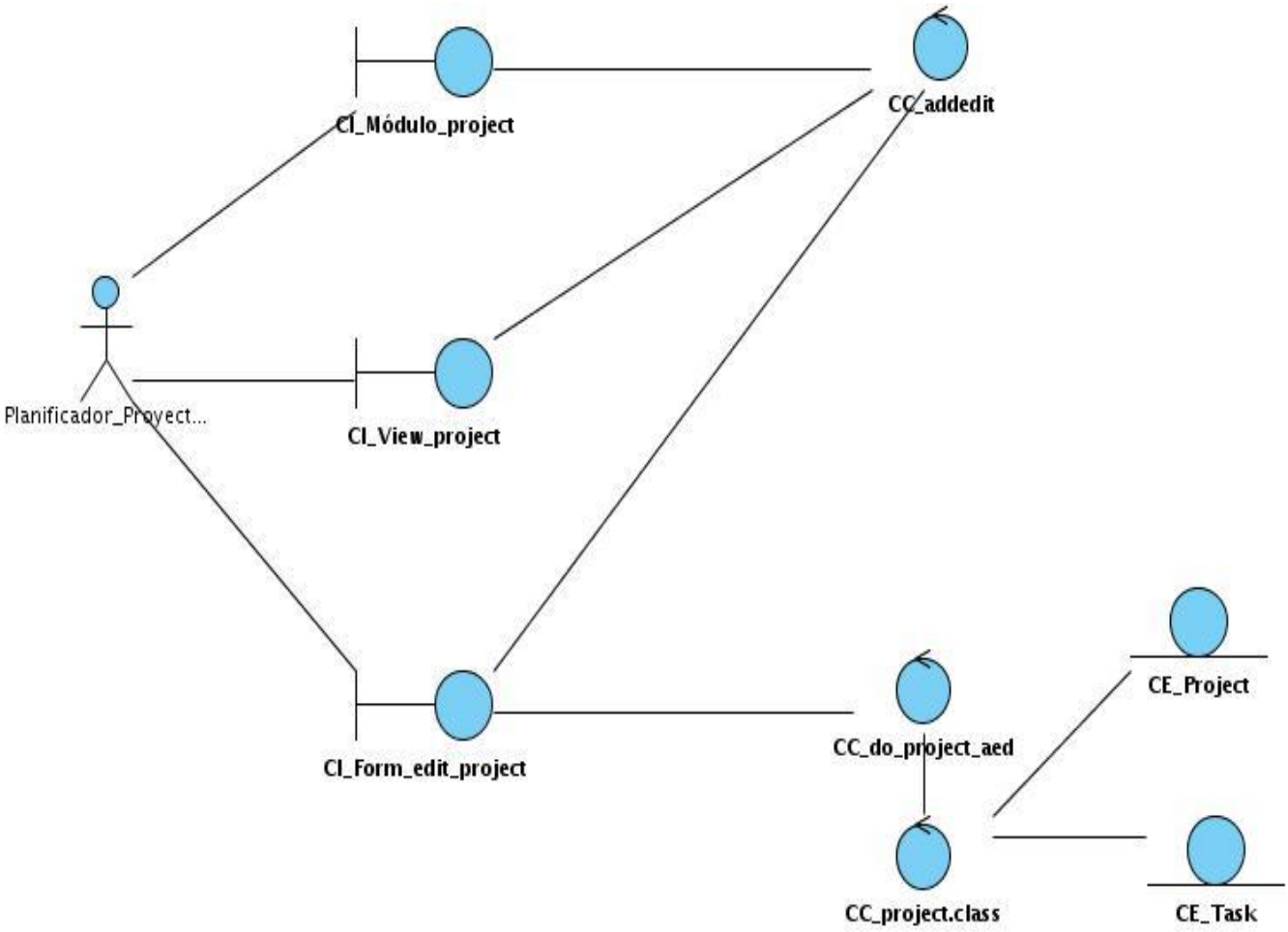
2.4.1 Diagramas de clases del análisis

Uno de los principales artefactos generados en el análisis es el Diagrama de clases del análisis. En este se representan las clases de análisis (clase interfaz, clase controladora y clase entidad) y las relaciones existentes entre ellas.

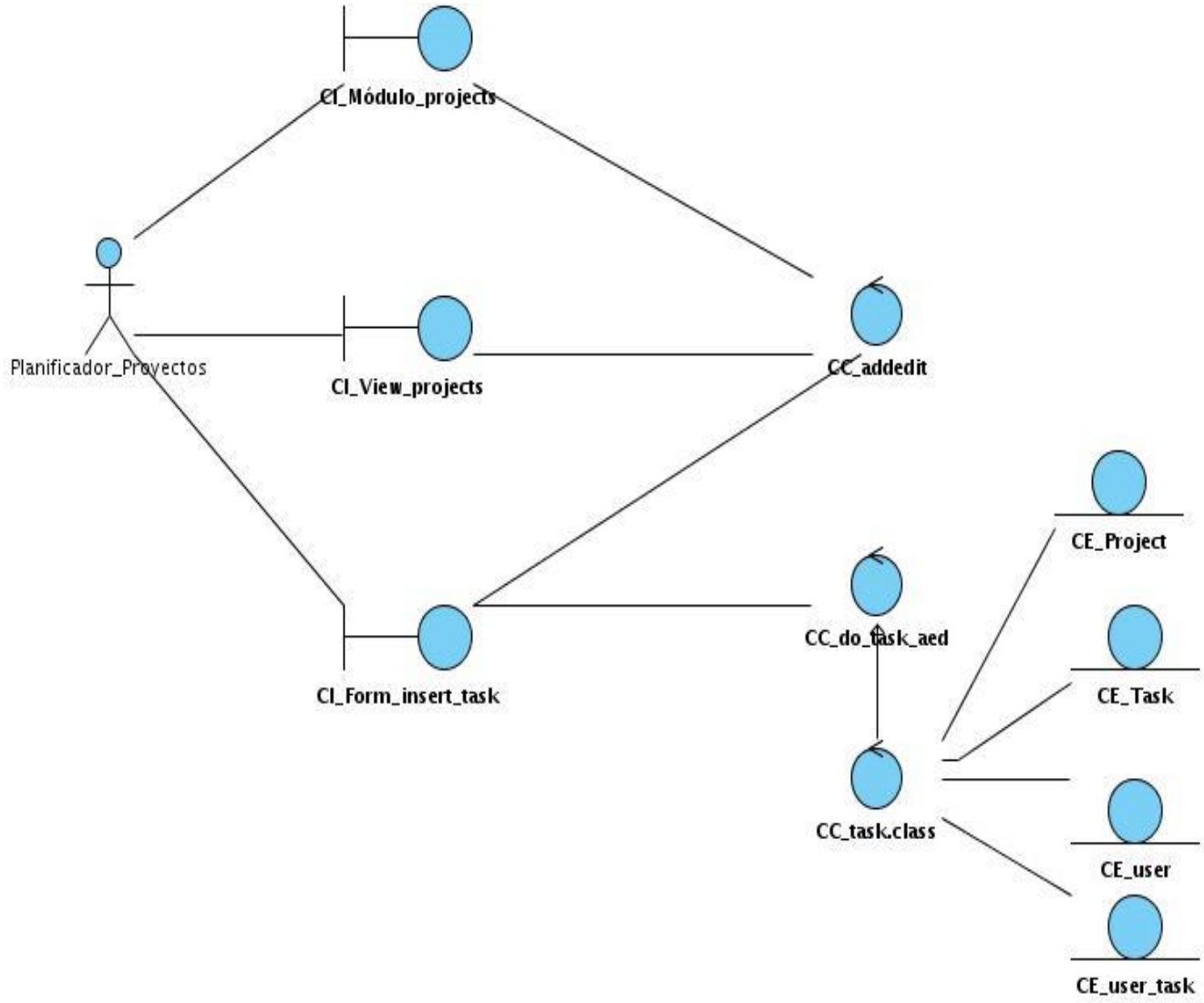
Diagramas de clases del análisis, Administrar proyectos. Escenario Insertar proyecto.



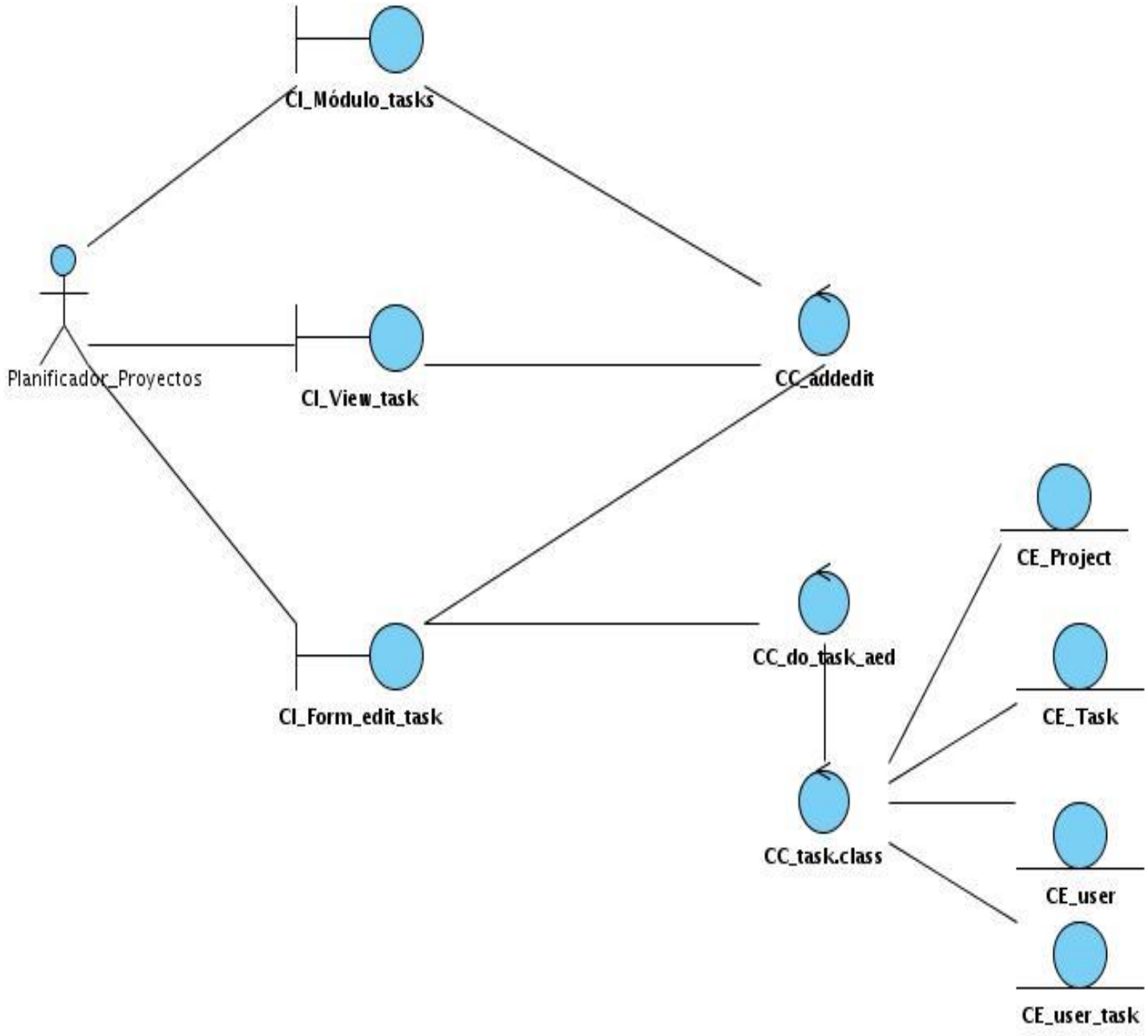
Diagramas de clases del análisis, Administrar proyectos. Escenario Editar proyecto.



Diagramas de clases del análisis, Administrar tareas. Escenario Insertar tareas.

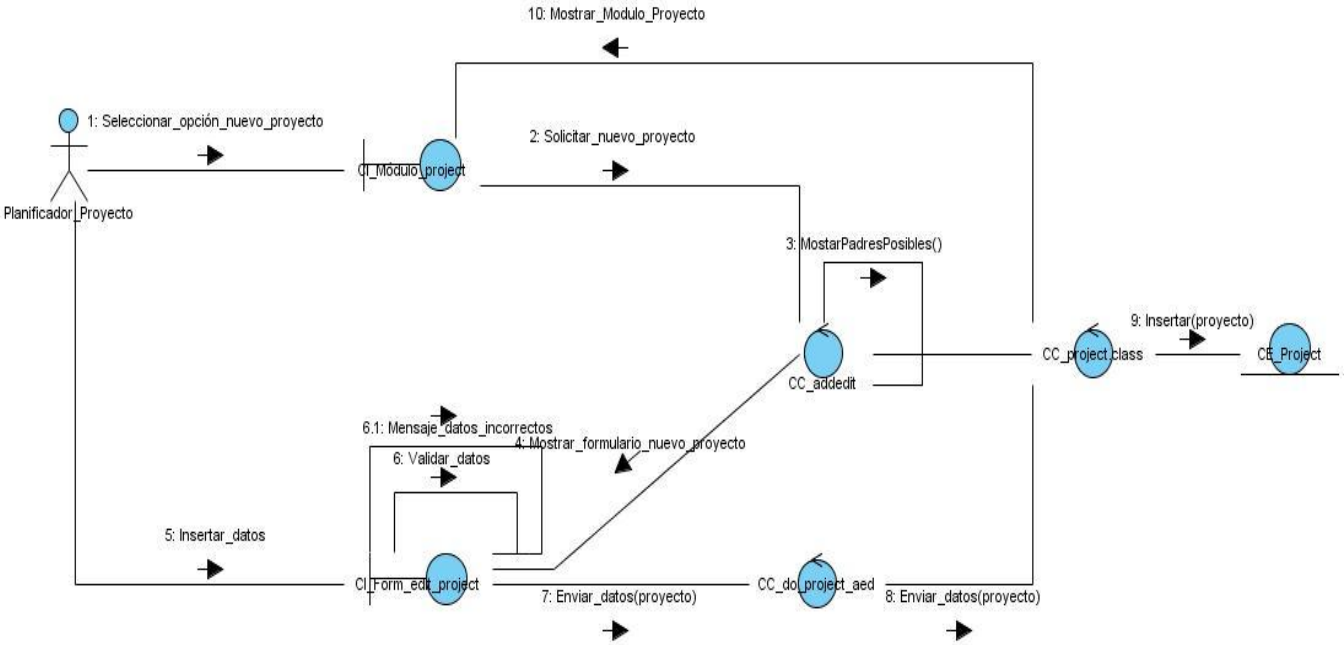


Diagramas de clases del análisis Administrar tareas. Escenario Editar tareas.

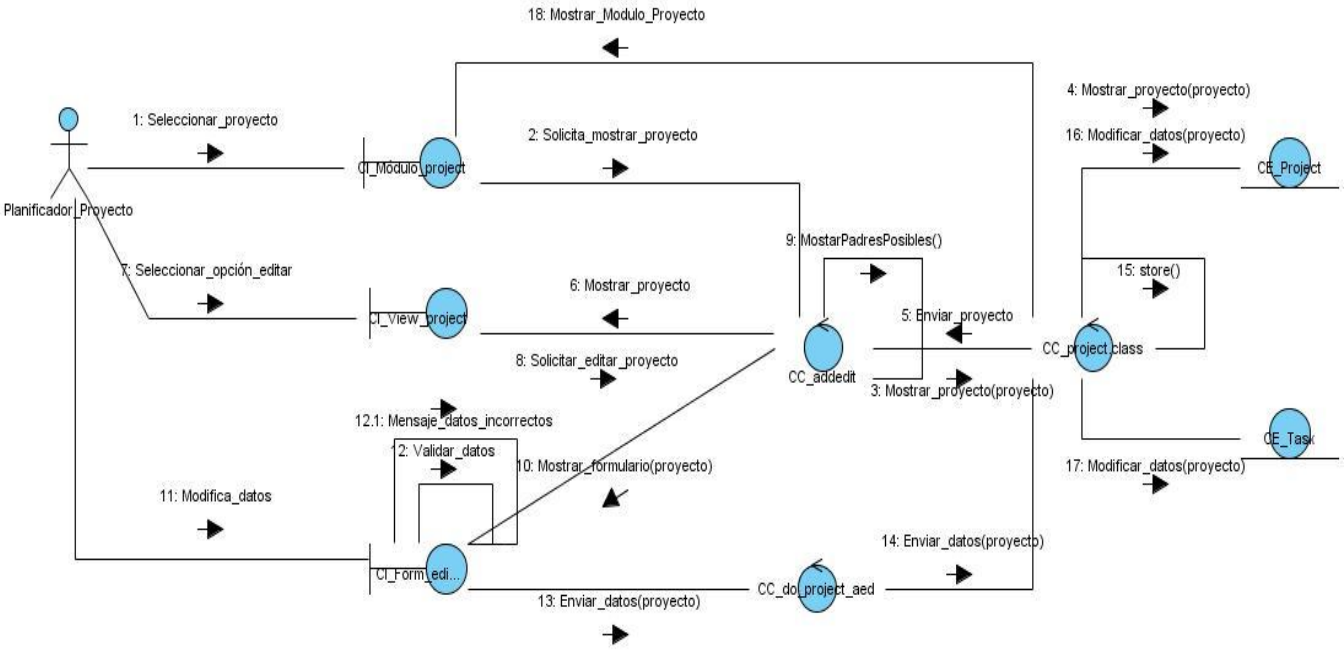


2.4.2 Diagramas de colaboración

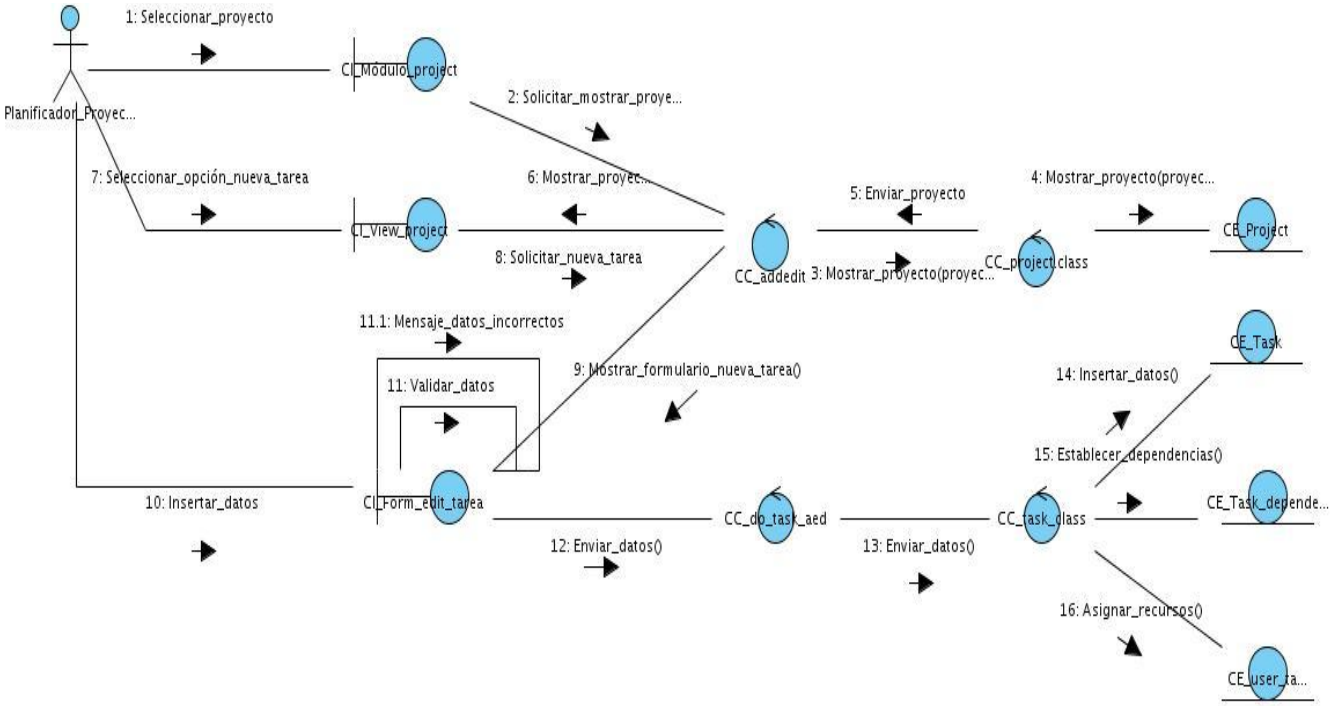
Diagramas de colaboración, Administrarproyectos. Escenario insertar proyecto.



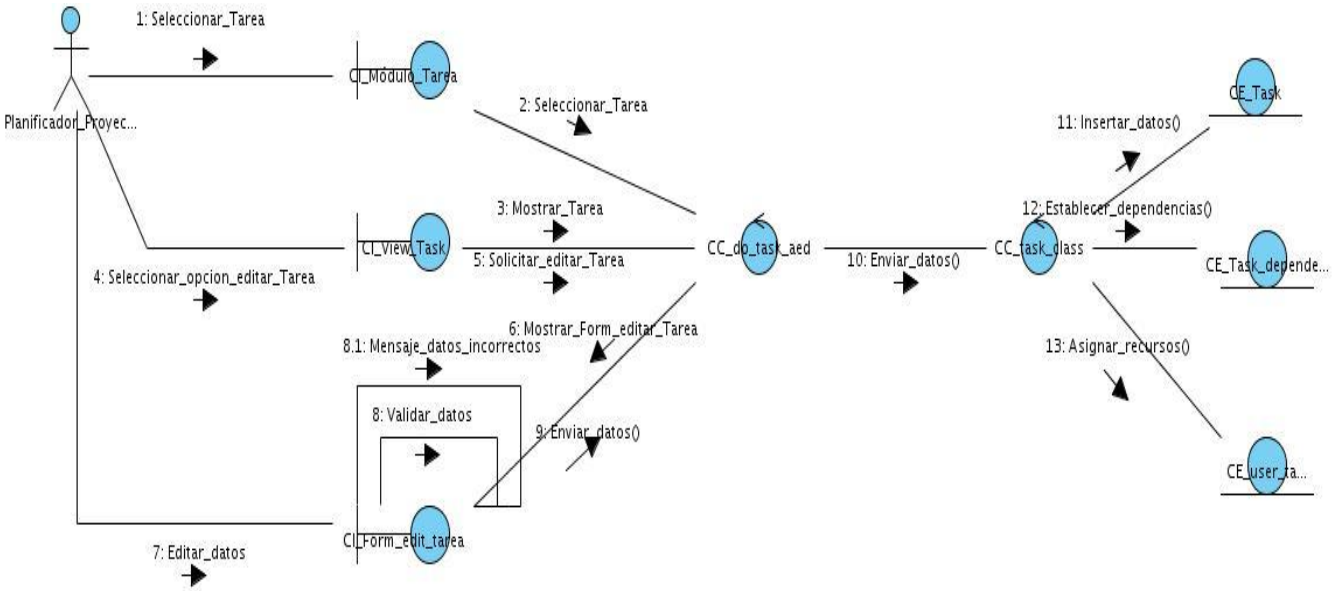
Diagramas de colaboración, Administrarproyectos. Escenarioeditar proyecto.



Diagramas de colaboración, Administrar tareas. Sección insertar tareas.



Diagramas de colaboración, Administrar tareas. Sección editar tareas.



2.5 - Diseño del sistema

El propósito del diseño es modelar el sistema y encontrar la forma para que soporte todos los requisitos. Se aterrizan todos los aspectos relacionados con las restricciones y características del sistema como lo son el lenguaje de programación a utilizar, el sistema operativo donde se podrá ejecutar la aplicación, las tecnologías de interfaz de usuario, en fin, agrupar en el diseño los requerimientos no funcionales definidos. Es un modelo físico que crea una entrada apropiada y un punto de partida para la implementación.

2.5.1 - Diagramas de clases del diseño

Diagrama de clases del diseño, Administrar proyectos. Escenario Insertar proyecto.

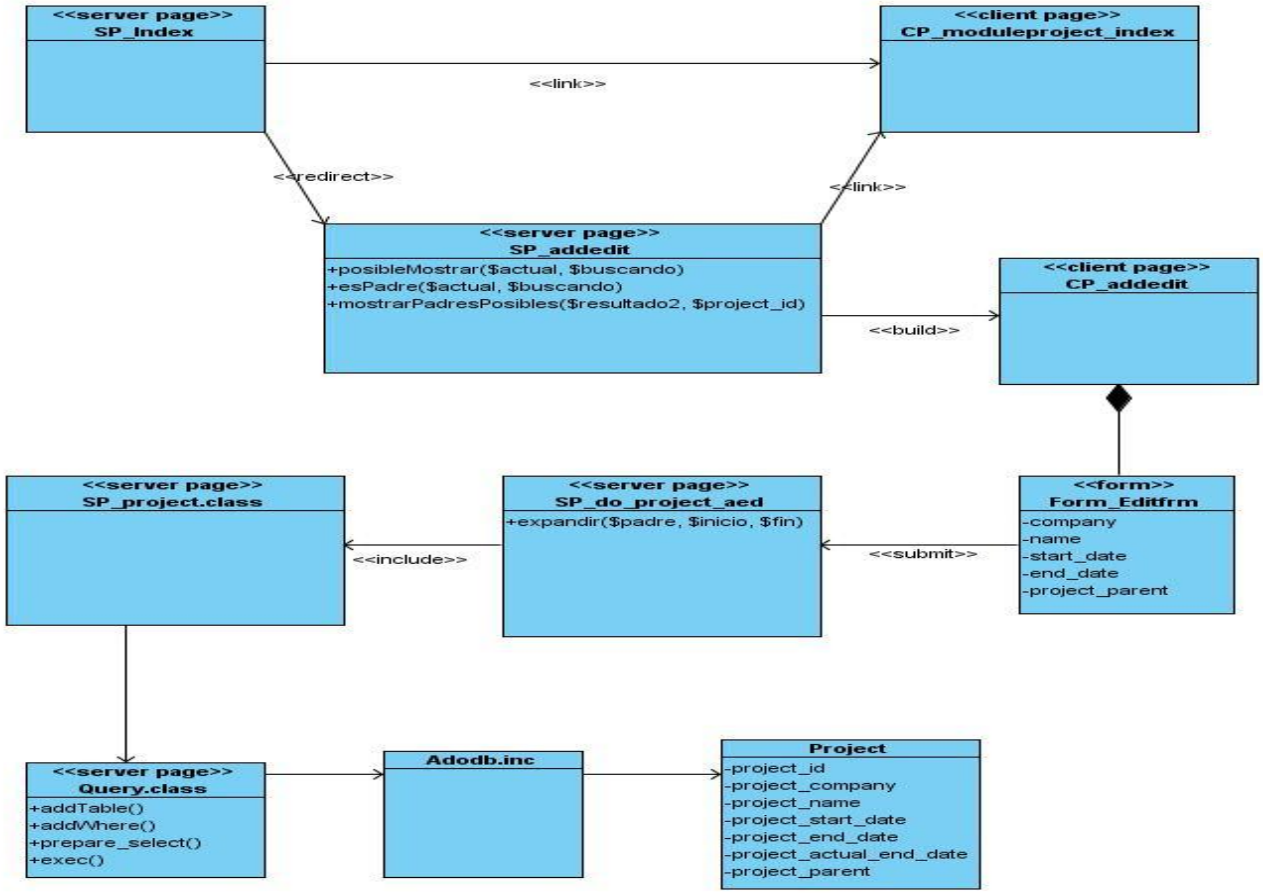


Diagrama de clases del diseño, Administrar proyectos. Escenario Editar proyecto.

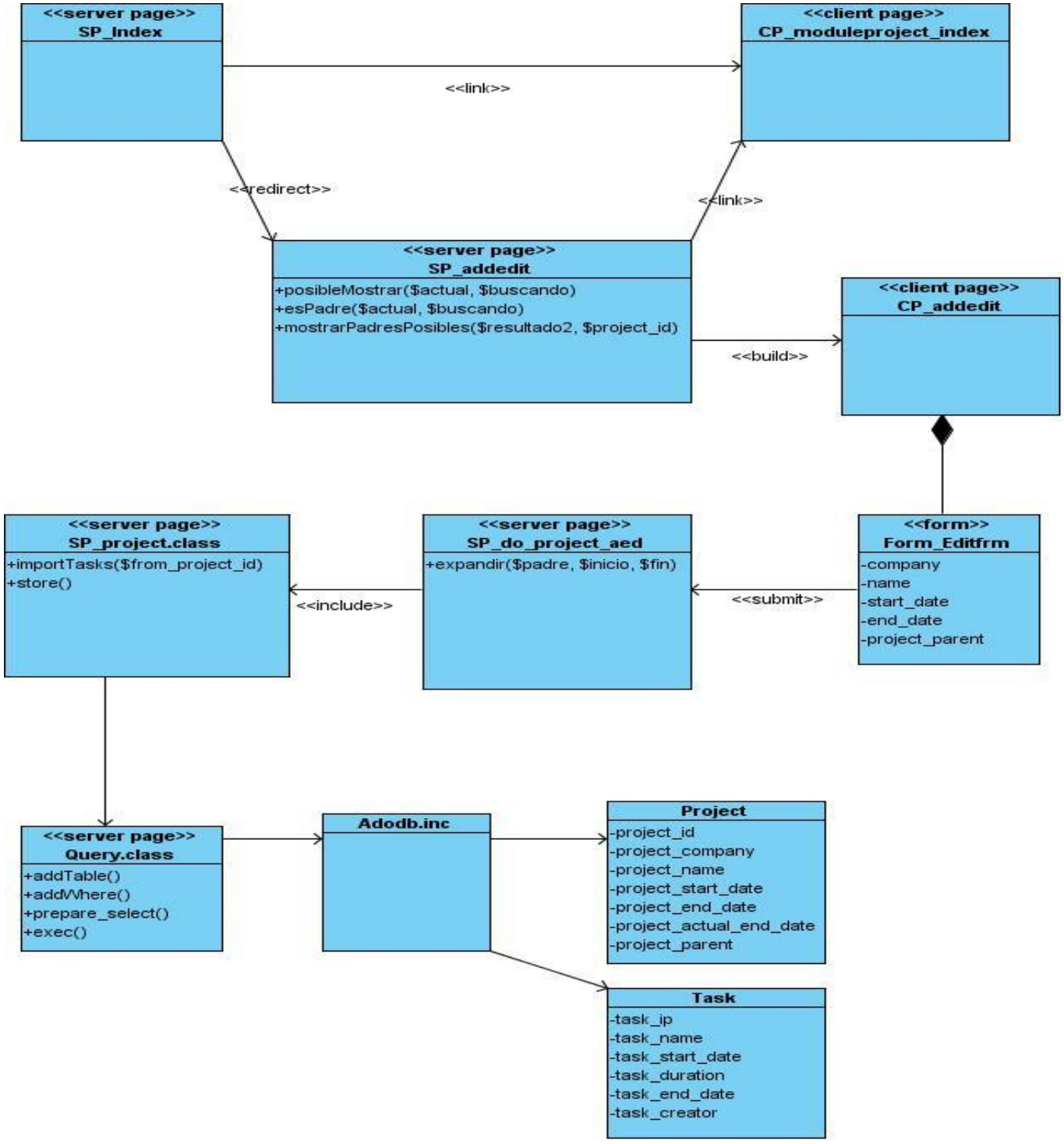


Diagrama de clases del diseño, Administrar tareas. Escenario Insertar tarea.

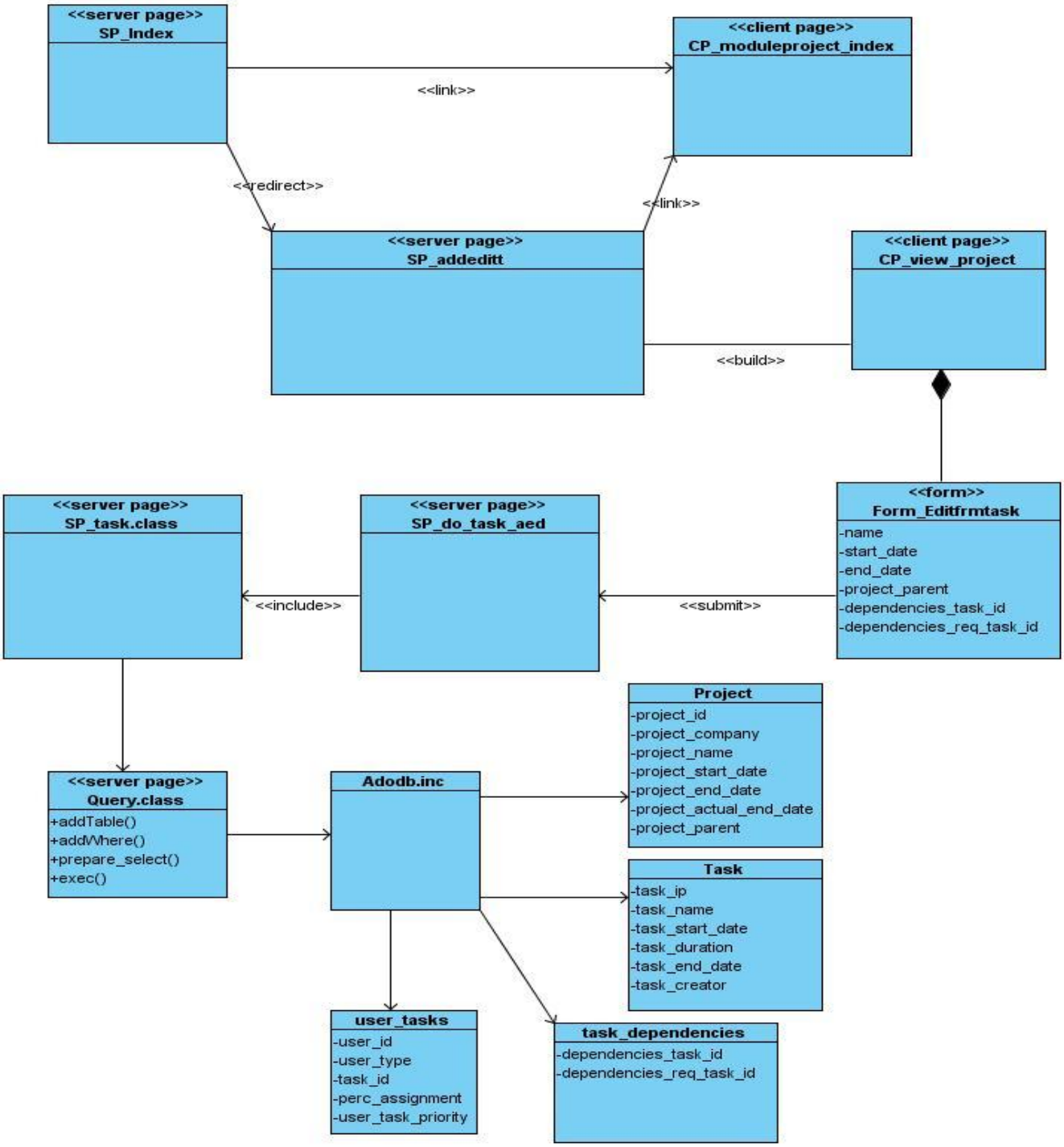
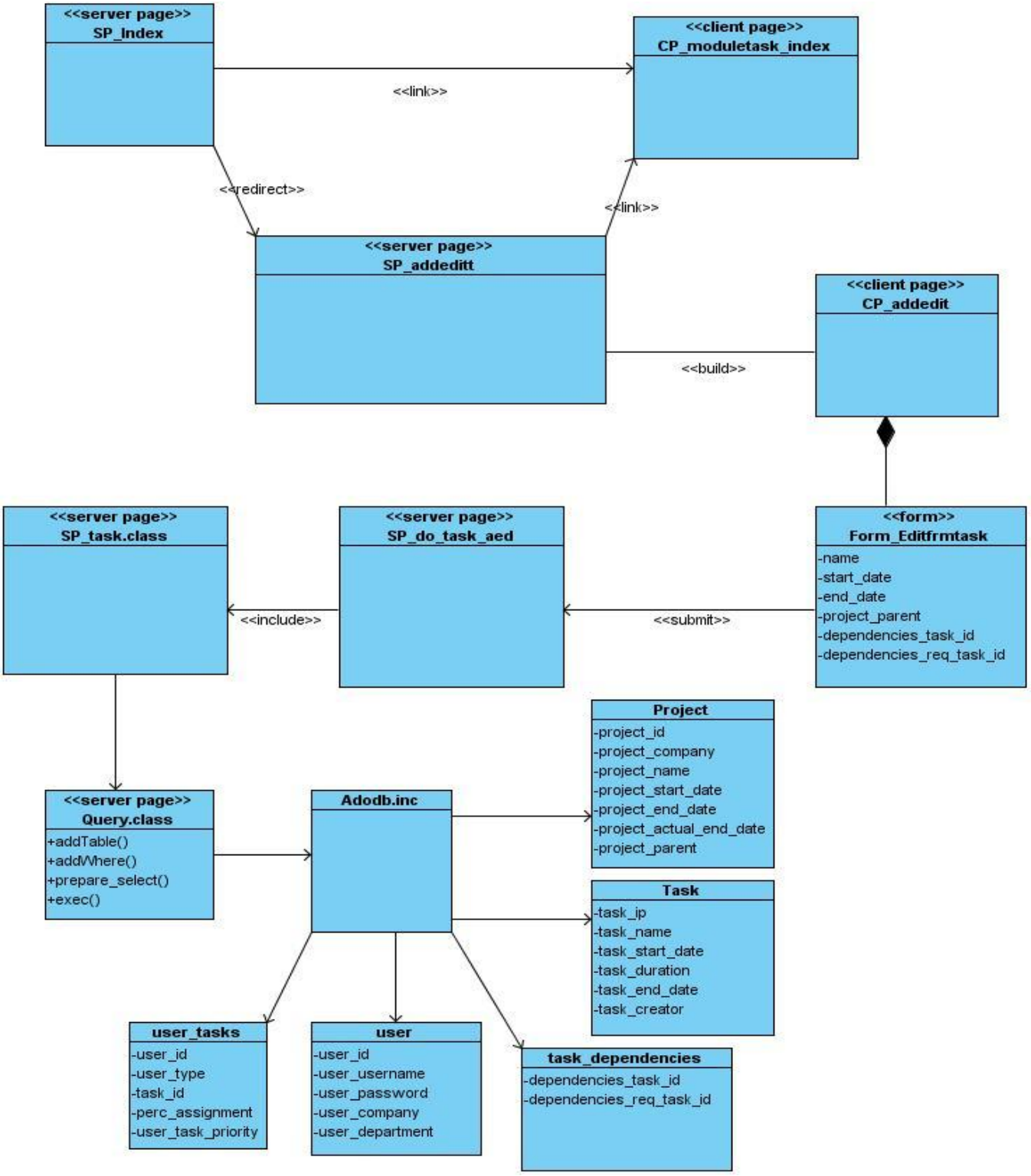


Diagrama de clases del diseño, Administrar tareas. Escenario Editar tarea.



2.6 - Conclusiones

- ✓ Se obtuvo un listado de funcionalidades que debe tener el sistema, expresados en los requisitos funcionales con una descripción detallada sobre las mejoras obtenidas por cada uno.
- ✓ Se realizó un modelado de los diagramas de clases del análisis de los casos de usos y los diagramas de colaboración para cada escenario, permitiendo un mayor entendimiento del funcionamiento de la herramienta.
- ✓ Se efectuó el diseño de la aplicación, esforzándose por conservar la estructura de la herramienta dotProject.

Capítulo 3. Implementación y pruebas

3.1 - Introducción

En el presente capítulo se trata el flujo de trabajo de implementación y pruebas, en el cual se construye el modelo necesario para desarrollar el proceso de implementación del sistema con el diagrama de componentes definido. También se elabora el diagrama de despliegue donde se representan los nodos necesarios, en los que se distribuye la aplicación. Además, se abordan los temas relacionados con las pruebas aplicadas al software.

3.2 - Modelo de Implementación

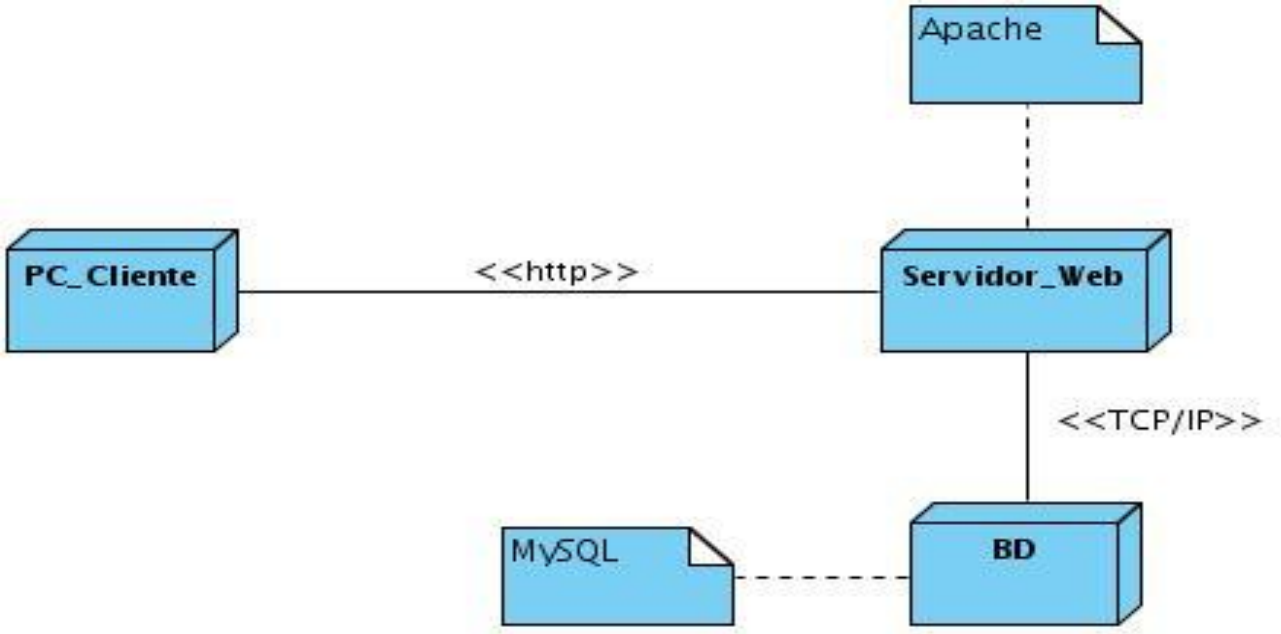
El modelo de implementación describe cómo los elementos del modelo del diseño y las clases se implementan partiendo de componentes, ficheros de código fuente, ejecutables, entre otros. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al describir los componentes y construir su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

3.2.1 - Modelo de Despliegue

El modelo de despliegue agrupa los objetos que describen la distribución física entre los nodos. Cada nodo representa un recurso que interviene en el despliegue del sistema. Describe la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Describe la topología del sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.

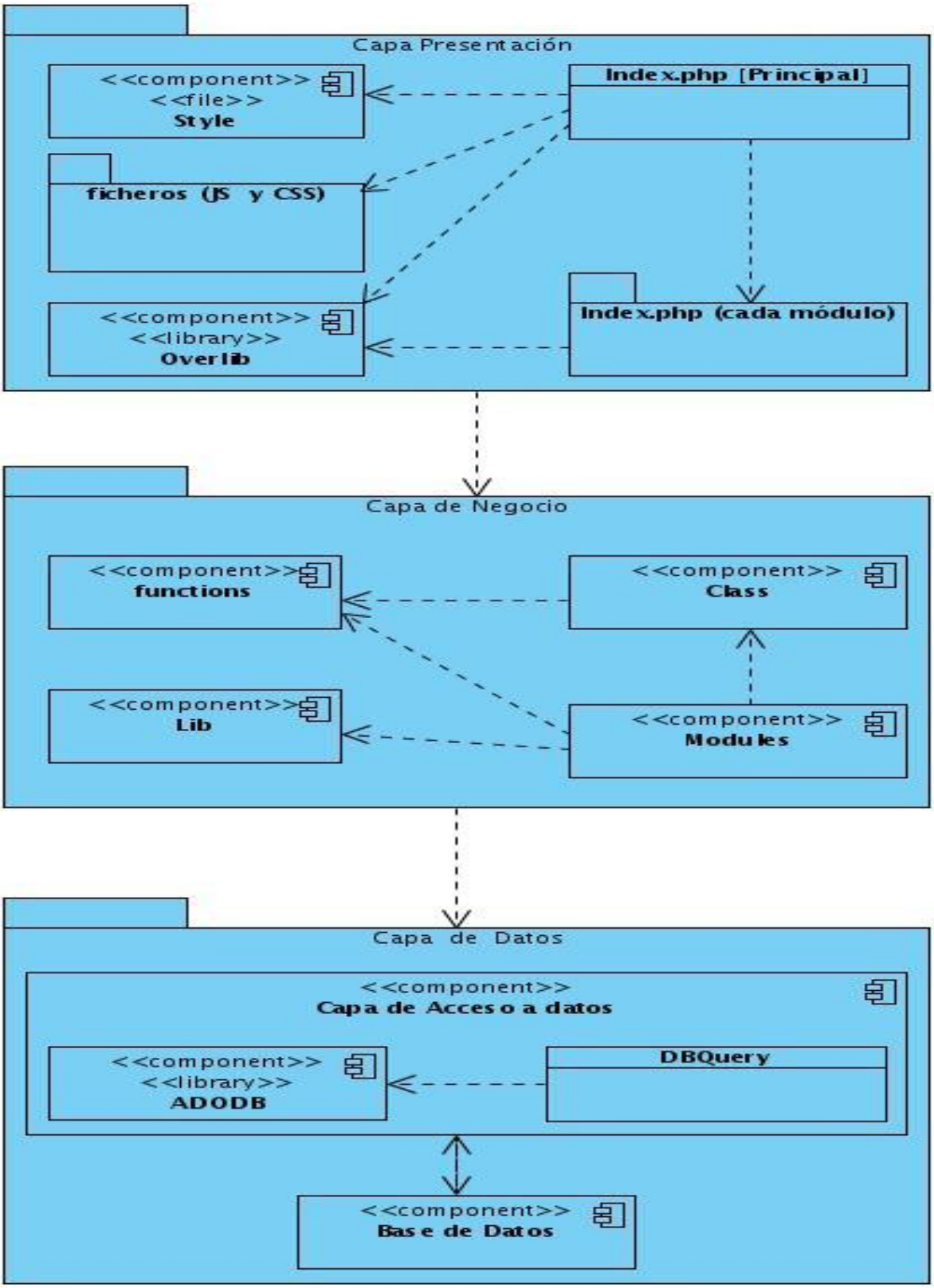
Diagrama de despliegue

El diagrama de Despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Es un grafo de nodos unidos por conexiones de comunicación donde cada nodo puede contener instancias de componentes. En general, un nodo puede ser una unidad de computación de algún tipo, desde un sensor hasta un mainframe y las instancias de componentes de software pueden estar unidas por relaciones de dependencia (11).



3.2.2 - Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables. Los elementos de modelado que lo conforman son los componentes y paquetes y muestran la estructura del sistema en términos de implementación a un alto nivel (12). Se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión de software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.


Capa Presentación
<<component>>
<<file>>
Style
Index.php [Principal]
ficheros (JS y CSS)
<<component>>
<<library>>
Overlib
Index.php (cada módulo)
Capa de Negocio
<<component>>
functions
<<component>>
Class
<<component>>
Lib
<<component>>
Modules
Capa de Datos
<<component>>
Capa de Acceso a datos
<<component>>
<<library>>
ADODB
DBQuery
<<component>>
Base de Datos

3.3 - Pruebas

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Todas las estrategias de prueba del software proporcionan al ingeniero del software una plantilla para la prueba y tienen las siguientes características generales:

- ✓ Las pruebas comienzan a nivel de módulos clase hacia afuera.
- ✓ En grandes proyectos es necesario un equipo de pruebas independientes, no así en pequeños proyectos donde el máximo responsable de ejecutar las pruebas es el jefe del proyecto.
- ✓ La prueba y la depuración de un proyecto son fases distintas, se debe tener en cuenta además que en el segundo caso se debe incluir una estrategia de prueba.

En una estrategia de prueba se deben incluir pruebas de bajo nivel para validar los pequeños segmentos de código y otras de alto nivel para los requisitos del cliente. En toda estrategia de prueba se debe tener en cuenta la verificación y la validación del producto que ha solicitado el cliente, esto es:

- Verificación: ¿Se construye correctamente el producto?
- Validación: ¿Se construye el producto correcto?

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Cualquier producto de ingeniería puede probarse usando uno o ambos enfoques:

- ✓ Conociendo la función específica para la que fue diseñado el producto se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores en cada función, conocida como las pruebas de caja negra. En este tipo de prueba se intenta demostrar que los datos se introducen y se procesan correctamente obteniéndose un resultado apropiado para estos datos introducidos.

- ✓ Conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que todas las piezas encajan, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han probado de forma adecuada, denominada como prueba de caja blanca, que tiene como fin comprobar los caminos lógicos del programa y conocer cómo funciona el programa en determinados sitios y bajo ciertas condiciones.

Estos enfoques de prueba se complementan entre sí, aunque pueden realizarse de manera independiente (13). Para la validación del sistema propuesto se llevará el método de caja negra.

3.3.1 - Pruebas de caja negra

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software (13).

Permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes; errores de interfaz, en estructuras de datos o en acceso a bases de datos externas; errores de rendimiento, de inicialización y de terminación (13).

3.3.2 - Casos de pruebas

Un Caso de Prueba (CP) es una especificación, usualmente formal, de un conjunto de entradas de prueba, condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de aspectos particulares de un elemento objeto de prueba (14).

- ✓ Los CP reflejan trazabilidad con los CU (Funcionalidad), ya que estos muestran una secuencia ordenada de eventos, al describir flujos básicos, flujos alternos, precondiciones y postcondiciones.
- ✓ Las especificaciones suplementarias de requerimientos pues existen otras características de calidad a evaluar, además de la funcionalidad, como usabilidad, confiabilidad, eficiencia, mantenibilidad y portabilidad.
- ✓ Las especificaciones de diseño del sistema, ya que se debe verificar que el software fue implementado según el diseño y que los elementos arquitectónicos garantizan la calidad del software.

Tabla 4. Diseño del caso de prueba, Administrar proyectos. Escenario Adicionar proyecto .

Condición de entrada	Casos válidos	Casos no válidos
Nombre	Nombre del proyecto	Campo vacío
Compañía	Seleccionar nombre de una compañía.	Campo vacío
Fechas de inicio y fin	Seleccionar fechas en orden lógico.	Seleccionar fechas de forma fecha de inicio posterior a fecha de fin.
Proyecto padre	Seleccionar proyecto padre.	

Caso de uso	**Administrar proyecto "Adicionar proyecto"**
Caso de prueba	Permitir insertar un proyecto en la base de datos introduciendo correctamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Calidad Compañía: UCI Fechas estimadas: 01/01/2010 - 01/02/2010
Resultado	El sistema introduce el proyecto en la base de datos.
Condiciones	No debe existir ningún campo vacío y los datos deben de ser del tipo especificado.

Caso de uso	**Administrar proyecto "Adicionar proyecto"**
Caso de prueba	Adicionar un proyecto introduciendo mal los datos o dejando algún campo vacío.
Entrada	Los camposNombre y Compañía están vacíos o las fechas no son correctas.
Resultado	El sistema muestra un mensaje de error especificando que los datos introducidos no son los correctos.
Condiciones	Deben existir campos vacíos o fechas con datos incorrectos.

Diseño del caso de prueba, Administrar proyectos. Escenario Editar proyecto.

Condición de entrada	Casos válidos	Casos no válidos

Nombre	Nombre del proyecto.	Campo vacío.
Compañía	Seleccionar nombre de una compañía.	Campo vacío.
Fechas de inicio y fin	Seleccionar fechas en orden lógico y ubicar las tareas con respecto a la fecha de inicio..	Seleccionar fechas de forma fecha de inicio, posterior a fecha de fin.
Proyecto padre	Seleccionar proyecto padre.	

Caso de uso	**Administrar proyecto "Editar proyecto"**
Caso de prueba	Modificar un proyecto en la base de datos introduciendo correctamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Calidad Compañía: UCI Fechas estimadas: 01/01/2010 - 01/02/2010
Resultado	El sistema modifica el proyecto y traslada sus tareas con respecto a la fecha de inicio en la base de datos.
Condiciones	No debe existir ningún campo vacío y los datos deben de ser del tipo especificado.

Caso de uso	**Administrar proyecto "Editar proyecto"**
Caso de prueba	Modificar un proyecto introduciendo mal los datos o dejando algún campo vacío.
Entrada	Los camposNombre y Compañía están vacíos o las fechas no son correctas.
Resultado	El sistema muestra un mensaje de error especificando que los datos introducidos no son los correctos y traslada las tareas a la fecha de inicio del proyecto.
Condiciones	Deben existir campos vacíos o fechas con datos incorrectos.

Diseño del caso de prueba, Administrar tareas. Escenario Adicionar tarea.

Condición de entrada	Casos válidos	Casos no válidos
Nombre	Nombre de la tarea.	Campo vacío.
Fechas de inicio y fin	Seleccionar fechas en orden lógico.	Seleccionar fechas de forma fecha de inicio posterior a fecha de fin.
Dependencias	Seleccionar tarea para establecer dependencias.	
Recursos humanos	Asignar recursos humanos.	

Caso de uso	**Administrar proyecto "Adicionar tarea"**
Caso de prueba	Permitir insertar una tarea en la base de datos introduciendo correctamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Diseño del sistema Fechas estimadas: 01/01/2010 - 14/01/2010 Dependencias: Análisis del sistema Recursos humanos: Juán González
Resultado	El sistema introduce la tarea en la base de datos, trasladando la fecha de inicio a la fecha de fin de la tarea que depende.
Condiciones	Los datos deben de ser del tipo especificado.

Caso de uso	**Administrar proyecto "Adicionar tarea"**
Caso de prueba	Permitir insertar unatarea en la base de datos introduciendo incorrectamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Diseño del sistema Fechas estimadas: 01/01/2010 - 14/01/2010 Dependencias: Análisis del sistema Recursos humanos: Juán González
Resultado	El sistema introduce la tarea en la base de datos, trasladando la fecha de inicio a la fecha de fin de la tarea que depende.
Condiciones	Los datos deben de ser del tipo especificado.

Diseño del caso de prueba Administrar proyectos escenario"Editar tarea ".

Condición de entrada	Casos válidos	Casos no válidos
Nombre	Nombre de la tarea	Campo vacío
Fechas de inicio y fin	Seleccionar fechas en orden lógico.	Seleccionar fechas de forma fecha de inicio posterior a fecha de fin.
Dependencias	Seleccionar tarea para establecer dependencias.	
Recursos humanos	Asignar recursos humanos	

Caso de uso	Administrar proyecto "Editartarea"
Caso de prueba	Modificar unatarea en la base de datos introduciendo correctamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Diseño del sistema Fechas estimadas: 01/01/2010 - 14/01/2010 Dependencias: Análisis del sistema Recursos humanos: Juán González
Resultado	El sistema introduce la tarea en la base de datos, trasladando la fecha de inicio a la fecha de fin de la tarea que depende.
Condiciones	Los datos deben de ser del tipo especificado.

Caso de uso	Administrar proyecto "Editartarea"
Caso de prueba	Modificar unatarea en la base de datos introduciendo incorrectamente los datos.
Entrada	El usuario introduce los datos del proyecto correctamente de la forma: Nombre: Diseño del sistema Fechas estimadas: 14/01/2010 - 01/01/2010 Dependencias: Análisis del sistema Recursos humanos: Juán González
Resultado	El sistema muestra un mensaje de error especificando que los datos introducidos no son los correctos.
Condiciones	Los datos deben de ser del tipo especificado.

3.4 - Conclusiones

- ✓ Se modelaron los diagramas de despliegue y componentes del sistema, ilustrando la relación existente entre los componentes del sistema y cómo se distribuye este.
- ✓ Se realizaron las pruebas pertinentes al producto final arrojando resultados satisfactorios.
- ✓ El sistema quedó desarrollado en su totalidad de manera factible y obteniendo una alta calidad como SGP.

Conclusiones generales:

En el presente trabajo de diploma se perfeccionó el SGP dotProject solucionando los problemas existentes a través de las siguientes acciones:

- ✓ La sistematización teórica realizada permitió constatar el estado actual de la gestión de proyectos constribuyendo a obtener una solución eficiente al problema de la investigación.
- ✓ Se perfeccionó la herramienta con un diseño de las funcionalidades descritas, obteniendo una nueva estructura de desglose, de fácil manejo y mejor optimizada.
- ✓ Se realizaron pruebas de caja negra a las funcionalidades modificadas verificando que las deficiencias fueron erradicadas satisfactoriamente.

Recomendaciones:

- ✓ Emplear el dotProyect para la planificación de proyectos en la UCI.
- ✓ Hacer extensiva la propuesta de este trabajo para las empresas y universidades cubanas productoras de software.

Referencias bibliográficas:

1. **Project Management Institute.***Guía de los Fundamentos de la Dirección de Proyectos (Guía del PMBOK®) Tercera Edición.* Newtown Square, Pennsylvania: PMI Publications, 2004. 19073-3299.

2. **Pecos, Daniel.** http://danielpecos.com. *PostGreSQL vs. MySQL.* [En línea] 2009. [Citado el: 18 de Febrero de 2010.] http://danielpecos.com/docs/mysql_postgres/x57.html.

3. Soporte Microsoft. *Lista de las nuevas características en Microsoft Project 2002.* [En línea] 22 de Marzo de 2007. [Citado el: 28 de Febrero de 2010.] http://support.microsoft.com/kb/322276/es#2. 322276.

4. **Dueñas, Joel Barrios.** alcance libre. [En línea] 16 de Marzo de 2007. [Citado el: 23 de Febrero de 2010.] http://www.alcancelibre.org/article.php/conoce-planner.

5. **Proyectos, Lic. Fernando Princich – Maestría en Ingeniería de Software – Administración de.** Principales aplicaciones, de código abierto bajo plataforma WEB, para gestión de proyectos. *Dot Project Review.* [En línea] 21 de 12 de 2009. [Citado el: 15 de mayo de 2010.] http://www.scribd.com/doc/24356996/Dot-Project-Review.

6. **Gracia., Lola Cárdenas y Joaquin.** WebEstilo. [En línea] 2000. [Citado el: 19 de Febrero de 2010.] http://www.webestilo.com/javascript/js00.phtml.

7. **Cornejo, José Enrique González.** docIRS. *¿Qué es UML?* [En línea] [Citado el: 18 de Febrero de 2010.] http://www.docirs.cl/uml.htm.

8. Modelo de Dominio. [En línea] [Citado el: 20 de Marzo de 2010.] <http://iie.fing.edu.uy/ense/asign/desasoft/practico/hoja8/ejemplos_clase2.pdf>.

9. **Alexander, C. A., Ishikawa, S., Silverstein, M., Jacobson.***A Pattern Language.* New York: Oxford Press, 1977.

10. EVA UCI. *CTP 1: Profundización en la disciplina de Requisitos Vista de los CU de la arquitectura. Patrones de CU. Descripción de los CU. Seguimiento y traceabilidad.* [En línea] Pág 8 - 9., Ingeniería de Software I., 2009-2010. [Citado el: 14 de Marzo de 2010.] http://eva.uci.cu/mod/resource/view.php?id=22431.

11. **Vilas, Ana Fernandez.** Diagrama de Despliegue. [En línea] 2001. [Citado el: 25 de Marzo de 2010.] <http://www-gris.det.uvigo.es/~avilas/UML/node50.html>.

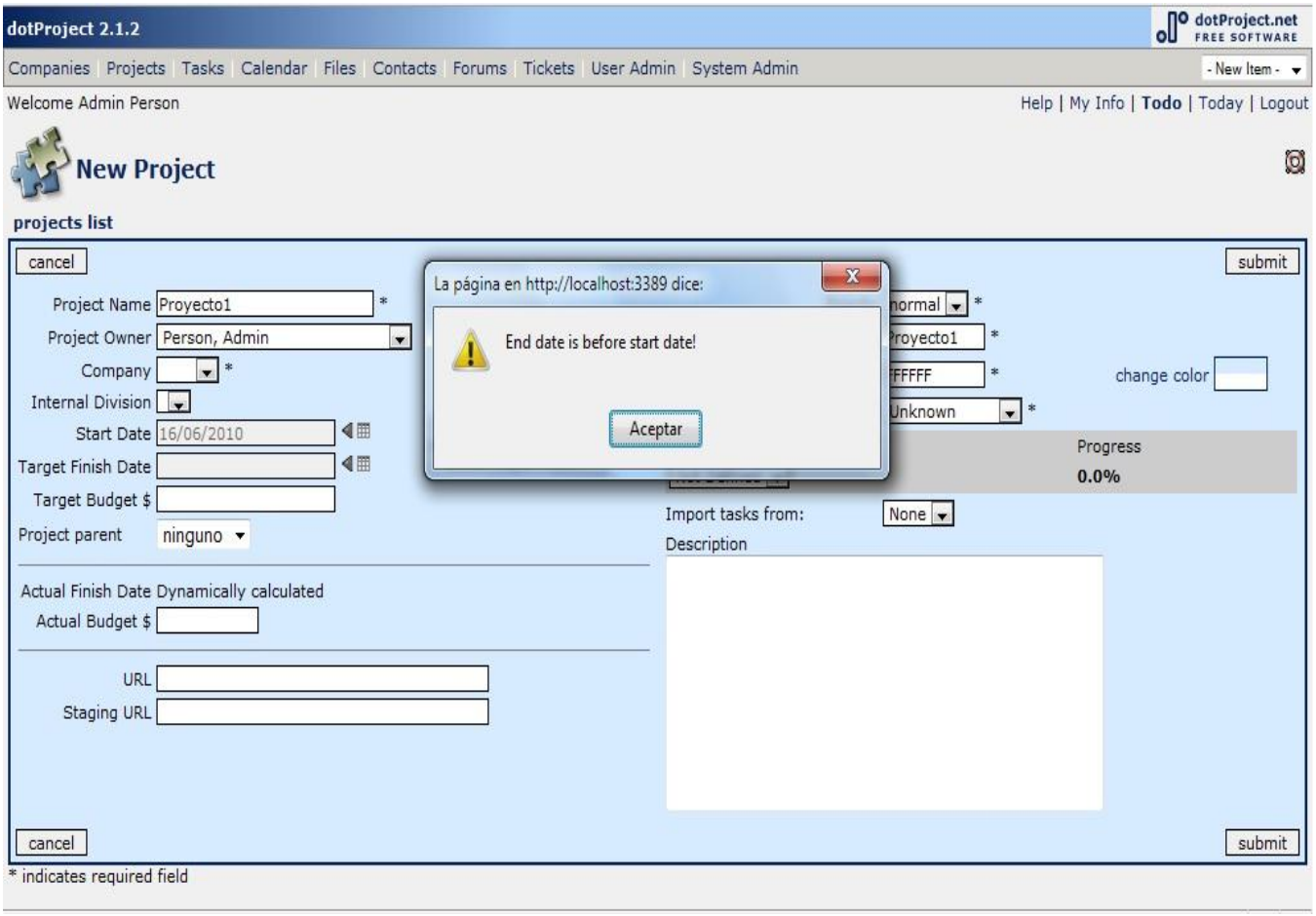
12. —. Diagrama de Componentes. [En línea] 2001. [Citado el: 25 de Marzo de 2010.] <http://www-gris.det.uvigo.es/~avilas/UML/node49.html>.

13. **Pressman, Roger S.** Ingeniería de software, un enfoque práctico. s.l.: McGraw-Hill, 2001, Técnicas de prueba del software, págs. 281–299.

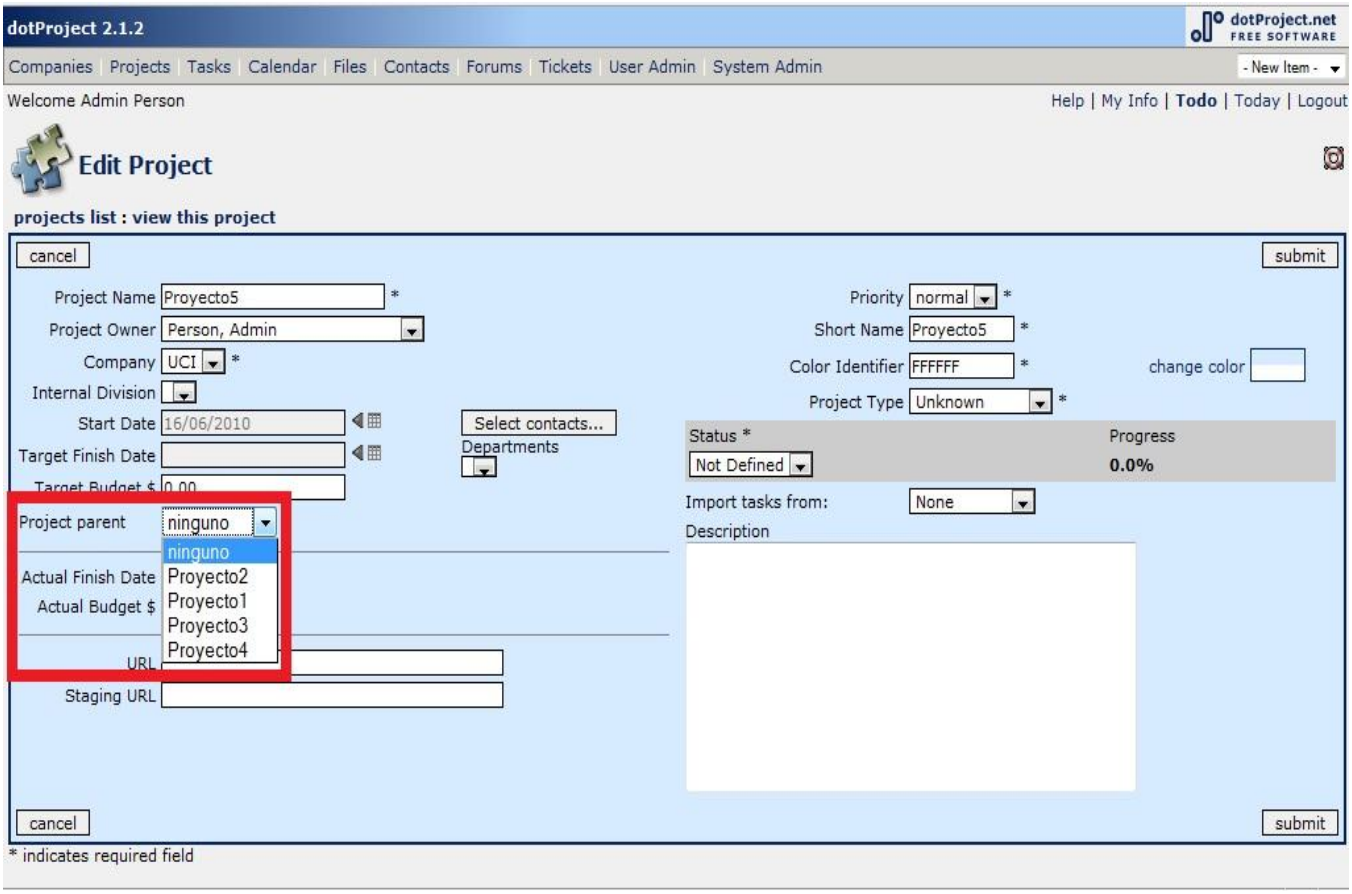
14. **D., Leffingwell D. and Widrig.***Managning Software Requirements, a Use Case Approach. Second Edition.* s.l.: Addison-Wesley, Pearson Education., 2006.

1. **Mario Torres**. PMBOK: Guía de los Fundamentos de la Dirección de Proyectos | ConstrucGeek.2008. [cited 18 May 2010]. Available from world wide web:<http://www.construcgeek.com/blog/pmbok-guia-de-los-fundamentos-de-la-direccion-de-proyectos>.
2. **php.net**. PHP: Hypertext Preprocessor. 2009. [cited 15 May 2010]. Available from world wide web: <http://www.php.net/>.
3. **Álvarez, Rubén Llovera. Febrero 2007.** *Planificación de proyecto.* Universidad de Salamanca: s.n., Febrero 2007.
4. **Franco, JoseAngel Navarro. 2008.** *UML en acción. Modelado de Aplicaciones Web.* [doc] Cuba: CUJAE, 2008.
5. **González, Francy, Rodriguez, Ronald y Peérez, Kelly. Agosto 2007.** *Manual de DotProject.* [pdf] Venezuela: Ministerio del poder Popular para la educación (MPPE), Agosto 2007.
6. **Gonzalez, Joel Estrada. 2007.** Desarrollo Web con PhP y MySQL. *La Web del programador.* [En línea] 2007. http://www.lawebdelprogramador.com.
7. **Pérez, YoandyVillazon.** *Curso de DotProject.* [ppt] Cuba: Proyecto Unicornios-Facultad 10.
8. **Paulo Nunes**. Concepto de Gestión de Proyectos. 2008. [cited 19 May 2010]. Available from world wide web: <http://www.knoow.net/es/cieeconcom/gestion/gestiondeproyectos.htm>.
9. **Ernesto Serrano**. dotProject para planificación de proyectos | AbartiaTeam. 2009. [cited 13 May 2010].Available from world wide web: <http://www.abartiateam.com/dotproject>.
10. **Pressman, Roger S**. "Ingeniería de Software. Un enfoque práctico." 5ta Edición (traducción de la edición original en Inglés "Software Enginering. A practical approach"), Editorial Mac Graw Hill, Madrid, España.
11. **TEAM, A**. Gestión de proyectos con dotProject, 2007. [Marzo, 2010]. Available from world wide web: http://www.abartiateam.com/dotproject.html.

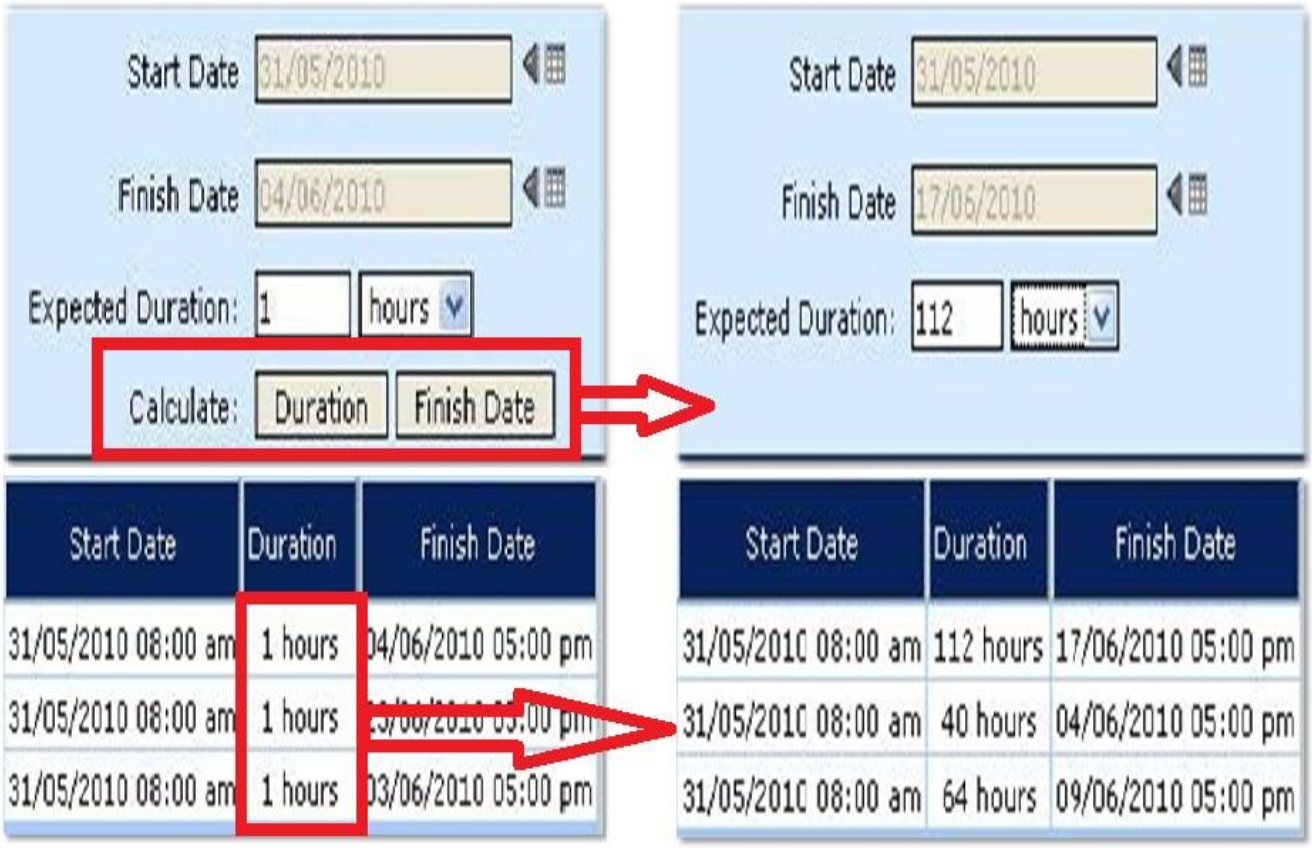
Anexo 1


dotProject 2.1.2
dotProject.net FREE SOFTWARE
Companies | Projects | Tasks | Calendar | Files | Contacts | Forums | Tickets | User Admin | System Admin
- New Item -
Welcome Admin Person
Help | My Info | Todo | Today | Logout
New Project
projects list
cancel
submit
Project Name Proyecto1 *
Project Owner Person, Admin
Company *
Internal Division
Start Date 16/06/2010
Target Finish Date
Target Budget $
Project parent ninguno
La página en http://localhost:3389 dice:
End date is before start date!
Aceptar
change color
Progress
0.0%
Import tasks from: None
Description
Actual Finish Date Dynamically calculated
Actual Budget $
URL
Staging URL
cancel
submit
* indicates required field

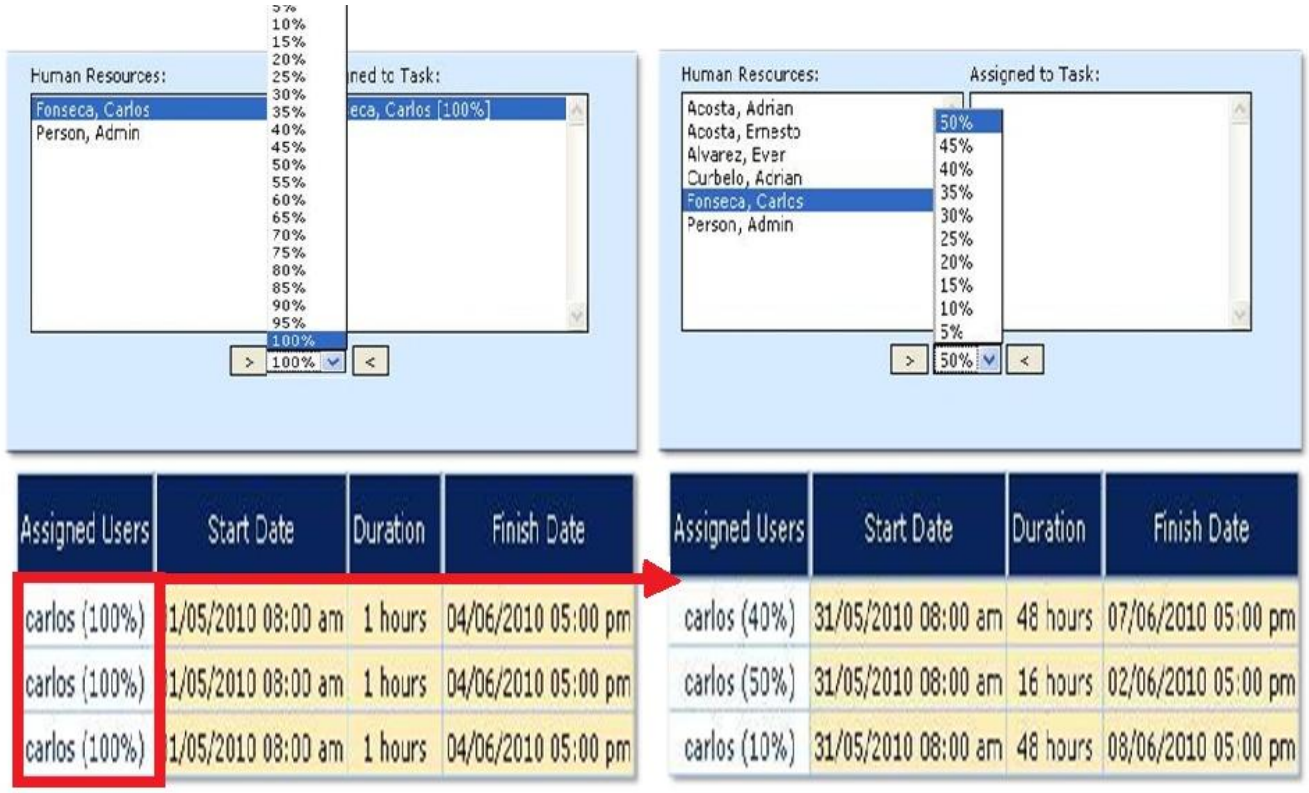
Anexo 2


dotProject 2.1.2
dotProject.net FREE SOFTWARE
Companies | Projects | Tasks | Calendar | Files | Contacts | Forums | Tickets | User Admin | System Admin
- New Item -
Welcome Admin Person
Help | My Info | Todo | Today | Logout
Edit Project
projects list : view this project
cancel
submit
Project Name Proyecto5 *
Project Owner Person, Admin
Company UCI *
Internal Division
Start Date 16/06/2010
Target Finish Date
Target Budget $ 0.00
Project parent ninguno
ninguno
Proyecto2
Proyecto1
Proyecto3
Proyecto4
Actual Finish Date
Actual Budget $
URL
Staging URL
Select contacts...
Departments
Priority normal *
Short Name Proyecto5 *
Color Identifier FFFFFF *
change color
Project Type Unknown *
Status *
Not Defined
Progress
0.0%
Import tasks from: None
Description
cancel
submit
* indicates required field

Anexo 3


Start Date 31/05/2010
Finish Date 04/06/2010
Expected Duration: 1 hours
Calculate: Duration Finish Date
Start Date 31/05/2010
Finish Date 17/06/2010
Expected Duration: 112 hours
Start Date Duration Finish Date
31/05/2010 08:00 am 1 hours 04/06/2010 05:00 pm
31/05/2010 08:00 am 1 hours
31/05/2010 08:00 am 1 hours 03/06/2010 05:00 pm
Start Date Duration Finish Date
31/05/2010 08:00 am 112 hours 17/06/2010 05:00 pm
31/05/2010 08:00 am 40 hours 04/06/2010 05:00 pm
31/05/2010 08:00 am 64 hours 09/06/2010 05:00 pm

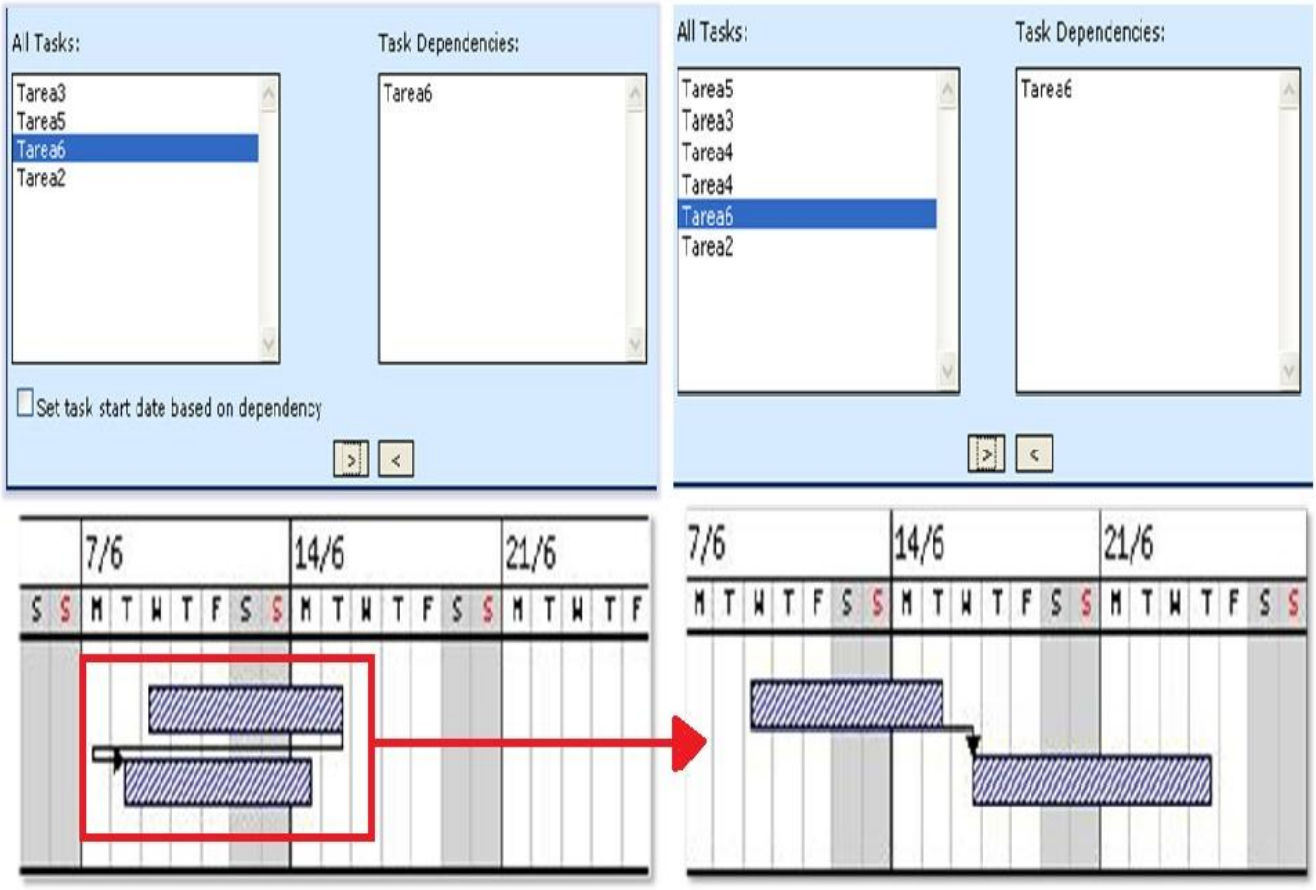
Anexo 4



Assigned Users	Start Date	Duration	Finish Date
carlos (100%)	1/05/2010 08:00 am	1 hours	04/06/2010 05:00 pm
carlos (100%)	1/05/2010 08:00 am	1 hours	04/06/2010 05:00 pm
carlos (100%)	1/05/2010 08:00 am	1 hours	04/06/2010 05:00 pm

Assigned Users	Start Date	Duration	Finish Date
carlos (40%)	31/05/2010 08:00 am	48 hours	07/06/2010 05:00 pm
carlos (50%)	31/05/2010 08:00 am	16 hours	02/06/2010 05:00 pm
carlos (10%)	31/05/2010 08:00 am	48 hours	08/06/2010 05:00 pm

Anexo 5


All Tasks:
Task Dependencies:
Tarea3
Tarea5
Tarea6
Tarea2
Tarea6
Set task start date based on dependency
All Tasks:
Task Dependencies:
Tarea5
Tarea3
Tarea4
Tarea4
Tarea6
Tarea2
Tarea6
7/6
14/6
21/6
7/6
14/6
21/6

[A]

API´s: Son las siglas en inglés de Application Programming Interface. En otras palabras, son los métodos que el desarrollador de cualquier aplicación ofrece a otros desarrolladores para que puedan interactuar con su aplicación**.**

[C]

C: Es un lenguaje de programación orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es uno de los lenguajes de programación más populares para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

[G]

Grafos: Un grafo se representa gráficamente como un conjunto de puntos (llamados vértices o nodos), unidos por líneas (aristas). Los grafos permiten estudiar las interrelaciones entre unidades que se encuentran en interacción.

[L]

Logs: Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación.

[N]

Nodos: En informática, un nodo es un «punto de intersección o unión de varios elementos que confluyen en el mismo lugar». Por ejemplo: en una red de ordenadores, cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.

[P]

PMI: El Project Management Institute (PMI)es una organización internacional sin fines de lucro que asocia a profesionales para la gestión de proyectos. Sus principales objetivos son: 1)Formular estándares profesionales, 2)Generar conocimiento a través de la investigación y 3)Promover la Gestión de Proyectos como profesión a través de sus programas de certificación.

Perl: es un lenguaje de programación, está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.