

**Universidad de las Ciencias Informáticas**  
**“Facultad # 6”**



**Título: “Módulos del Dotproject  
para la gestión de recursos y horarios”**

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

**Autor:** Youssary Hernández Torres

**Tutor:** Ing. Dennys Hernández Peña

Junio 2009

## Declaración de Auditoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Youssary Hernández Torres (Autor)

---

Ing. Dennys J. Hernández Peña (Tutor)

## **GRADECIMIENTOS**

Agradezco a todas aquellas personas que aportaron un poco en mi formación, no quisiera que se me quedara nadie por mencionar, pues fueron 6 largos años en los cuales pasaron varias etapas de mi vida, por ello no quisiera mencionar nombres.

Los que me ayudaron un día, a todos mis más sinceros agradecimientos,

A los que están y a los que ya no se encuentran.

Muchas gracias.

## **DEDICATORIA**

Dedico este trabajo a quien ya no se encuentra, mi abuela; a la cual le prometí un día cuando era muy chico, titularme como universitario.

Pues, para ti abuelita, tu sueño y el mío ya está cumplido.

## RESUMEN

La Universidad de las Ciencias Informáticas (UCI) es una institución educacional en constante evolución. Una de las tareas iniciales de un proyecto es la gestión del propio proyecto que nos permite hacer un balance de los recursos con las tareas a realizar, definiendo el plan de trabajo con la fecha promedio de realización de un proyecto.

Para la gestión de proyecto, la Universidad cuenta con disímiles herramientas, que controlan, y evalúan el proceso. En la Facultad 6 se utiliza para la gestión de proyecto el Dotproject, el cual está estructurado mediante módulos, que nos permiten llevar un control de los miembros, usuarios, proyectos, tareas, documentos y recursos. En el módulo de recursos, se asignan los recursos del proyecto, pero no individualmente a un miembro un recurso en específico. Además no se puede asignar un software para realizar una tarea.

Dentro del Dotproject no se encuentra un módulo capaz de gestionar el horario contra tarea y recurso; por lo que teniendo en cuenta la situación, se realizó la modificación del módulo de recursos y se creó un módulo para la asignación de horario por recursos y tareas a un miembro específico. La aplicación se desarrolló en un ambiente multiplataforma, utilizando para su implementación el lenguaje PHP. Con el incremento de estos módulos la aplicación constará con nuevas funcionalidades capaces de distribuir los recursos PC a un miembro del proyecto, y los software necesarios para las tareas a realizar. También se podrán planificar el horario de trabajo a un miembro del proyecto en una PC con un horario para una tarea.

## PALABRAS CLAVE

DOTPROJECT RECURSOS TAREAS HORARIO

## Índice

GRADECIMIENTOS .....	III
DEDICATORIA .....	IV
RESUMEN.....	V
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA .....</b>	<b>6</b>
<b>Sistemas informáticos para la planificación en proyecto de desarrollo de software.....</b>	<b>6</b>
1.1 Planificación de proyecto .....	7
1.2 Etapas de un proyecto .....	8
1.3 Gestión de proyecto .....	12
1.4 Software internacional para la gestión y control de Proyecto .....	12
1.5 Herramientas propuestas para el desarrollo del sistema .....	19
1.6 Lenguaje de programación: PHP .....	21
1.7 Sistemas gestores de base de datos (SGBD) .....	22
1.8 Entornos de desarrollo integrado (IDE) .....	23
1.9 Lenguaje de Modelación UML (Unified Modeling Language) .....	26
1.10 Rational Unified Process (RUP) .....	29
1.11 Arquitectura del software.....	30
1.12 Arquitectura en Capas .....	31
Conclusiones .....	33
<b>CAPÍTULO 2 Descripción de la propuesta de la solución .....</b>	<b>34</b>
<b>Análisis de la gestión de software del proceso de la asignación de horario y recursos en proyecto.....</b>	<b>34</b>
2.1 Modelo de Dominio .....	34
2.2 Modelo de negocio.....	35
2.3 Descripción del negocio .....	35
2.4 Regla del Negocio.....	36
2.5 Actor del Negocio.....	36
2.6 Trabajadores del Negocio .....	37

# INDICE

2.8 Diagramas de Casos de Usos del Negocio .....	42
2.9 Entidades del Negocio .....	44
2.10 Modelo de Objetos del Negocio .....	44
2.11 Requisitos .....	45
2.12 Actor del sistema .....	48
2.13 Caso de uso del sistema .....	49
2.14 Descripción textual de los casos de uso del sistema.....	50
Conclusiones .....	68
<b>Capítulo 3 Diseño del sistema .....</b>	<b>69</b>
3.1 Diseño .....	69
3.2 Diagrama de clase del diseño .....	70
3.3 Diagrama de secuencia .....	73
3.4 Descripción de las clases.....	80
3.5 Base de datos .....	85
3.6 Modelo de Datos.....	89
3.7 Diagrama de clases persistentes .....	90
3.8 Diagrama de despliegue .....	91
3.9 Definiciones de diseño .....	91
3.10 Tratamiento de errores.....	92
3.11 Seguridad .....	92
Conclusiones .....	92
<b>CAPÍTULO Implementación .....</b>	<b>93</b>
4.1 Diagrama de componentes .....	93
4.2 Modulo de la aplicación.....	98
Conclusiones .....	99
<b>Conclusiones generales .....</b>	<b>100</b>
<b>Recomendaciones.....</b>	<b>101</b>
TRABAJOS CITADOS .....	102
BIBLIOGRAFÍA.....	103
WEBGRAFIA .....	104
GLOSARIO DE TÉRMINOS .....	105

ANEXOS..... 107

**Índice de Ilustraciones**

Ilustración 1 Diferencias entre tipos de proyectos ..... 10

Ilustración 2 Diagrama de Caso de Uso del Negocio ..... 37

Ilustración 3 Definir Usuario..... 42

Ilustración 4 Asignar Recurso ..... 42

Ilustración 5 Planificar Horario ..... 43

Ilustración 6 Definir Recursos ..... 44

Ilustración 7 Modelo de Objeto ..... 45

Ilustración 8 Diagrama de caso de uso del sistema..... 49

Ilustración 9 Prototipo no funcional. Gestionar Software ..... 52

Ilustración 10 Prototipo no funcional. Insertar Software..... 52

Ilustración 11 Prototipo no funcional. Gestionar PC ..... 55

Ilustración 12 Prototipo no funcional. Insertar PC..... 55

Ilustración 13 Prototipo no funcional. Insertar Horario por día ..... 60

Ilustración 14 Prototipo no funcional. Insertar Horario de semana ..... 60

Ilustración 15 Prototipo no funcional. Gestionar asignación de Software ..... 64

Ilustración 16 Prototipo no funcional. Gestionar asignación de PC ..... 67

Ilustración 17 Diagrama de clases, caso de uso gestionar software ..... 70

Ilustración 18 Diagrama de clase. Caso de uso gestionar PC ..... 70

Ilustración 19 Diagrama de clase. Caso de uso Gestionar asignación de software..... 71

Ilustración 20 Diagrama de clase. Caso de uso Gestionar Horario ..... 72

Ilustración 21 Diagrama de clases. Caso de uso Recursos..... 73

Ilustración 22 Diagrama de secuencia. Insertar asignación de software ..... 74

Ilustración 23 Diagrama de secuencia. Buscar asignación de PC..... 75

Ilustración 24 Diagrama de secuencia. Modificar asignación de software ..... 76

Ilustración 25 Diagrama de secuencia. Eliminar asignación de software ..... 77

Ilustración 26 Diagrama de clases. Buscar Horario ..... 77

Ilustración 27 Diagrama de secuencia insertar horario de la semana ..... 78

Ilustración 28 Diagrama de secuencia. Insertar horario por día..... 78

Ilustración 29 Diagrama de secuencia. Eliminar horario ..... 79

Ilustración 30 Modelo de Datos ..... 89

Ilustración 31 Diagrama de Clases persistentes..... 90

Ilustración 32 Diagrama de despliegue..... 91

Ilustración 33 Diagrama de componente. Gestionar PC..... 94

Ilustración 34 Diagrama de componente. Gestionar asignación de PC..... 94

Ilustración 35 Diagrama de componente. Gestionar Horario ..... 95

Ilustración 36 Diagrama de componente. Gestionar Hoarario día ..... 95

Ilustración 37 Diagrama de componente. Gestionar Horario Semana..... 96

Ilustración 38 Diagrama de componente. Gestionar Recurso..... 96

## INDICE

Ilustración 39 Diagrama de componente. Gestionar Software .....	97
Ilustración 40 Diagrama de componente. Gestionar asignación de software .....	97
Ilustración 41 Distribución de las Carpetas.....	98
Ilustración 42 Diagrama de Secuencia. Eliminar PC .....	108
Ilustración 43 Diagrama de Secuencia. Insertar PC .....	109
Ilustración 44 Diagrama de secuencia. Modificar PC .....	110
Ilustración 45 Diagrama de secuencia Mostrar PC.....	111
Ilustración 46 Diagrama de secuencia. Modificar asignación PC .....	111
Ilustración 47 Diagrama de secuencia. Buscar Asignación de PC .....	112
Ilustración 48 Diagrama de secuencia. Eliminar asignación de PC .....	112
Ilustración 49 Diagrama de secuencia. Insertar asignación de PC .....	113
Ilustración 50 Diagrama de secuencia. Buscar Asignación de software .....	114
Ilustración 51 Diagrama de secuencia. Eliminar Asignación de software .....	115
Ilustración 52 Diagrama de secuencia. Modificar Asignación de software .....	115
Ilustración 53 Diagrama de secuencia. Insertar Asignación de software .....	116
Ilustración 54 Diagrama de secuencia. Mostrar Recursos.....	117

## Índice de Tablas

Tabla 1 Etapas en el desarrollo de una aplicación .....	11
Tabla 2 Actor del negocio .....	37
Tabla 3 Trabajadores del Negocio.....	38
Tabla 4 Descripción de casos de uso: Definir Recursos.....	38
Tabla 5 Descripción de casos de uso: Definir Usuario .....	39
Tabla 6 Descripción de casos de uso: Asignar Recurso.....	40
Tabla 7 Descripción de casos de uso: Crear horario de trabajo para proyecto .....	40
Tabla 8 Actor del sistema .....	48
Tabla 9 Gestionar Software .....	50
Tabla 10 Gestionar PC .....	53
Tabla 11 Gestionar Horario.....	56
Tabla 12 Gestionar asignación de Software .....	61
Tabla 13 Gestionar asignación de PC .....	64
Tabla 14 Matriz de trazabilidad entre requerimientos y caso de uso del sistema .....	68
Tabla 15 descripción de clases. Gestionar Software .....	80
Tabla 16 Descripción de clase. Gestionar PC .....	81
Tabla 17 Descripción de clase. Gestionar PC .....	82
Tabla 18 Descripción de clase. Buscar recursos .....	83
Tabla 19 Descripción de clase. Gestionar Software .....	84
Tabla 20 Descripción entidad Resource_Type .....	85
Tabla 21 Descripción entidad Resources .....	85
Tabla 22 Descripción entidad Resource_Tasks.....	85
Tabla 23 Descripción entidad Tasks .....	86
Tabla 24 Descripción entidad Projects .....	87
Tabla 25 Descripción entidad users.....	88
Tabla 26 Descripción entidad Resource_Users.....	88

INDICE

Tabla 27 Descripción entidad Horario..... 88

# INTRODUCCIÓN

## Introducción

Desde la antigüedad, el hombre se ha preocupado por la organización de las tareas y por saber con la frecuencia con que ocurrirán todos los fenómenos del mundo que le rodea.

Sin tener mucho conocimiento de lo que verdaderamente estaba haciendo, iba desarrollando la cultura del uso y manejo del tiempo.

En la didáctica magna de Juan A. Conunius (1592-1670) que fue escrita en 1632, se planteaba: " Hay que hacer una escrupulosa distribución del tiempo para que cada año, día y hora tenga su particular ocupación ".

Más tarde, con el desarrollo de la Humanidad, se fueron realizando estudios acerca de cómo confeccionar un horario de trabajo. El plan de trabajo constituye el instrumento organizativo de las actividades laborales, permite dar una visión general de las distribuciones cuantitativas de las tareas y recursos asignadas para un proyecto, según el tiempo previsto.

La Universidad de las Ciencias Informáticas (UCI) es una institución educacional en constante evolución, que forma profesionales de la informática con una alta capacitación científica, comprometidos con el desarrollo del país. Este centro experimental, surgido en el año 2002, garantiza la preparación académica y el entrenamiento profesional de excelencia a miles de jóvenes informáticos, con segundos perfiles profesionales validados a través del vínculo directo y sistemático con la producción, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación, el empleo de las tecnologías más actuales y diseños curriculares flexibles. Lleva a cabo, importantes proyectos en sectores como la salud, la educación y los servicios al ciudadano y gobierno, a partir del uso de una moderna infraestructura tecnológica y novedosos métodos de organización, que impulsan la informatización de la sociedad cubana y el desarrollo de la Industria Cubana del Software.

## INTRODUCCIÓN

En la UCI se localiza una de las principales redes del país; en ella se forman estudiantes y profesores relacionados con la producción de software en proyectos productivos, cuya principal herramienta de trabajo son las computadoras.

La Producción de Software y Servicios Informáticos se basa en la integración de los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva.

Este espacio de integración temática es denominado Polo Productivo y se promueve la formación de pregrado, postgrado, la colaboración nacional e internacional, el fomento de líneas de investigación, desarrollo y ejecución de proyectos correspondientes a los acuerdos de trabajo.

Todo esto relacionado con grupos de trabajo denominados Dirección de Calidad, que asume los laboratorios de pruebas, grupo de auditoría y revisiones; y otros dos normativos que son Aseguramiento de la Calidad y Métricas.

Para la gestión de proyecto se definen los recursos y tareas, los cuales deben ser distribuidos para lograr la poner fin a esa tarea en un tiempo determinado.

Una mejor distribución de los recursos y organización del tiempo de trabajo hacen que las tareas se cumplan en el tiempo requerido, pues controla el aprovechamiento de los recursos, la utilización del software necesario para la materialización de las tareas; hace que el trabajo sea más rápido.

Al no constar con una herramienta que genere un horario asignado a una tarea, asigne un recurso a tarea u organice los recursos, nos planteamos el siguiente problema científico:

¿Cómo lograr una adecuada planificación de los recursos y la asignación de horarios de trabajo a los integrantes de un proyecto de desarrollo de software?

## INTRODUCCIÓN

Para la siguiente problemática nos trazamos como Objeto de estudio los sistemas informáticos para la planificación en proyecto de desarrollo de software, cuyo campo de acción serían los sistemas informáticos para la planificación de horarios y recursos en proyectos de desarrollo de software.

La planificación del proyecto 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.

Dentro de los sistemas de planificación de proyectos existe el Microsoft Project, herramienta privativa; el OpenProj, que es un software multiplataforma con funcionalidades básicas para la gestión de proyecto; y Redmine y Dotproject, que son aplicaciones WEB, multiplataforma.

El Dotproject es una aplicación web basada en PHP que incluye módulos para compañías, proyectos, tareas (con diagramas de Gantt), foros, ficheros, calendario, contactos, ayuda de escritorio, soporte multilinguaje, módulos con permiso y themes. Corre tanto en Windows como en Linux. Con licencia GPL.

En la facultad 6 para la gestión de proyecto utilizamos esta herramienta, la cual nos permite mantener una adecuada planificación, manteniendo las funcionalidades de la misma nos trazamos el siguiente objetivo general: crear los módulos para la gestión de recursos y horarios en Dotproject.

Para la problemática antes mencionada nos trazamos los siguientes objetivos específicos: analizar el proceso de gestión de recursos y horarios; diseñar los componentes de recursos y horarios; e implementar dichos componentes en los módulos de recursos y horarios.

Las tareas trazadas para la investigación son:

1. Estudio del proceso de planificación de proyectos dentro de la gestión del desarrollo de software en el mundo.

## INTRODUCCIÓN

2. Estudio de la estructura e implementación del Dotproject.
3. Identificación de los requerimientos funcionales y no funcionales.
4. Descripción de los casos de uso del negocio para la gestión de recursos y horarios en los proyectos de desarrollo de software.
5. Diseño de clases.
6. Implementación de las clases diseñadas.

El presente documento cuenta con un Resumen, varios capítulos que constituyen el cuerpo del trabajo, Conclusiones, Recomendaciones, Bibliografía y Referencias bibliográficas. Los capítulos son:

### **CAPÍTULO 1: Fundamentación teórica**

En este capítulo se muestran algunos conceptos fundamentales para la comprensión de la gestión de horario y recursos; junto a ello, realizar un análisis detallado del estado del arte de los distintos sistemas para la gestión de proyecto, comparando sus ventajas y desventajas. Se describen los sistemas integrados y distribuidos, el uso de las arquitecturas existentes y la metodología a utilizar para el análisis y diseño del sistema teniendo en cuenta las facilidades que puede aportar al trabajo. Se hace un estudio del lenguaje de programación y del sistema gestor de base de datos (SGBD) usado; definiéndose las tecnologías más adecuadas para el sistema.

### **CAPÍTULO 2: Descripción de la propuesta de solución**

En este capítulo se realiza un análisis del proceso que se lleva a cabo para realizar la planificación del horario en proyectos, facilitando una mayor comprensión del problema presente. Se presenta el modelo del negocio con el objetivo de entender el plano en que se emplaza el sistema y por tanto contribuir a la comprensión de los requisitos que se desprenden de este contexto. Se determinan los actores y trabajadores del negocio y del sistema; se plantean las principales reglas del negocio a tener en cuenta en el sistema a desarrollar; se muestran los casos de uso del negocio detallados y una representación del

## INTRODUCCIÓN

modelo de objetos; así como una especificación de los requisitos y casos de usos del sistema generados a raíz de la captura de requisitos.

### **Capítulo 3 Diseño del sistema**

En el siguiente capítulo se muestra los modelos visuales del sistema, transformando los requisitos a diseños de sistema, se adapta el diseño para hacerlo correspondiente con el ambiente de implementación para así lograr mejores resultados, se hace un análisis de la base de datos del DotProject, haciendo la ingeniería inversa de la misma, generando el diagrama de clases persistentes.

### **Capítulo 4 Implementación**

El siguiente capítulo muestra las definiciones de los diferentes modelos utilizados en la implementación para la solución del problema y se procede a la construcción de los requisitos.



## **CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA**

### **Sistemas informáticos para la planificación en proyecto de desarrollo de software**

#### **Introducción**

En este capítulo se muestran algunos conceptos fundamentales para la comprensión de la gestión de horario y recursos; junto a ello, realizar un análisis detallado del estado del arte de los distintos sistemas para la gestión de proyecto, comparando sus ventajas y desventajas. Se describen los sistemas integrados y distribuidos, el uso de las arquitecturas existentes y la metodología a utilizar para el análisis y diseño del sistema teniendo en cuenta las facilidades que puede aportar al trabajo. Se hace un estudio del lenguaje de programación y del sistema gestor de base de datos (SGBD) usado; definiéndose las tecnologías más adecuadas para el sistema.

La administración de proyectos es la aplicación de conocimiento, habilidades, herramientas, y técnicas a actividades de proyectos de manera que cumplan o excedan las necesidades y expectativas de partidos interesados de un proyecto. Cumplir o exceder las necesidades o expectativas de los partidos interesados invariablemente involucran balancear demandas que compiten entre sí, tales como:

- Alcance, tiempo, costo y calidad.
- Partidos interesados con diferentes necesidades y expectativas.

### **¿Qué es un sistema?**

Un sistema es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio.

### **¿Qué es un Proyecto?**

Se puede definir proyecto: un proyecto es un esfuerzo caracterizado esencialmente por la exclusividad de sus condiciones dentro del conjunto. Completar con éxito el Proyecto significa cumplir con los objetivos dentro de las especificaciones técnicas, de costo y de plazo de terminación. **(1)**

A un conjunto de proyectos orientados a un objetivo superior se denomina Programa, y un conjunto de programas constituye un plan, como corresponde generalmente a los grandes Planes Nacionales.

### **1.1 Planificación de proyecto**

La planificación de un proyecto debe afrontarse de manera adecuada para que al final del mismo se pueda hablar de éxito. No se trata de una etapa independiente abordable en un momento concreto del ciclo del proyecto. Es decir, no se puede hablar de un antes y un después al proceso de planificación puesto que según avance el proyecto será necesario modificar tareas, reasignar recursos, entre otros.

Se debe tener claro que si bien sí podemos 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.

Encontramos 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 correspondientes estudios técnicos, de mercado, financieros, de rentabilidad... así como una estimación de los recursos necesarios y los costes 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 se ha decidido ejecutar el proyecto. Ahora 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 atención debida 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.

### 1.2 Etapas de un proyecto

Desde un punto de vista muy general puede considerarse que todo proyecto tiene tres grandes etapas:

- **Fase de planificación.** Se trata de establecer cómo el equipo de trabajo deberá satisfacer las restricciones de prestaciones, planificación temporal y coste. Una planificación detallada da consistencia al proyecto y evita sorpresas que nunca son bien recibidas.
- **Fase de ejecución.** Representa el conjunto de tareas y actividades que suponen la realización propiamente dicha del proyecto, la ejecución de la obra de que se trate. Responde, ante todo, a las características técnicas específicas de cada tipo de proyecto y supone poner en juego y gestionar los recursos en la forma adecuada para desarrollar la obra en cuestión. Cada tipo de proyecto responde en este punto a su tecnología propia, que es generalmente bien conocida por los técnicos en la materia.
- **Fase de entrega o puesta en marcha.** Como ya se ha dicho, todo proyecto está destinado a finalizarse en un plazo predeterminado, culminando en la entrega de la obra al cliente o la puesta en marcha del sistema desarrollado, comprobando que funciona adecuadamente y responde a las especificaciones en su momento aprobadas. Esta fase es también muy importante no sólo por representar la

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

culminación de la operación sino por las dificultades que suele presentar en la práctica, alargándose excesivamente y provocando retrasos y costes imprevistos.

A estas tres grandes etapas es conveniente añadir otras dos que, si bien pueden incluirse en las ya mencionadas, es preferible nombrarlas de forma independiente ya que definen un conjunto de actividades que resultan básicas para el desarrollo del proyecto:

- **Fase de iniciación.** Definición de los objetivos del proyecto y de los recursos necesarios para su ejecución. Las características del proyecto implican la necesidad de una fase o etapa previa destinada a la preparación del mismo, fase que tienen una gran trascendencia para la buena marcha del proyecto y que deberá ser especialmente cuidada. Una gran parte del éxito o el fracaso del mismo se fragua principalmente en estas fases preparatorias que, junto con una buena etapa de planificación, algunas personas tienden a menospreciar, deseosas por querer ver resultados excesivamente pronto.
- **Fase de control.** Monitorización del trabajo realizado analizando cómo el progreso difiere de lo planificado e iniciando las acciones correctivas que sean necesarias. Incluye también el liderazgo, proporcionando directrices a los recursos humanos, subordinados (incluso subcontratados) para que hagan su trabajo de forma efectiva y a tiempo.

Estas etapas citadas presentan, sin embargo, características bastante diferentes según se trate de proyectos internos o de proyectos externos. Las principales diferencias aparecen en la etapa de planificación. En el proyecto externo existen un conjunto de acciones que se relacionan con la necesidad de presentar una oferta al cliente y lograr la adjudicación del contrato en competencia con otras empresas o personas. Si, por la razón que fuere, el contrato no se consigue el proyecto queda abortado antes de haberse comenzado y carece de sentido preocuparse de cómo debe ser gestionado. La exigencia comercial tiene, pues, un carácter prioritario para las empresas, siendo la consecución del contrato paso imprescindible para poder acometer un proyecto concreto y, con una perspectiva más amplia, condición

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

esencial para la supervivencia de la empresa. Puedes ver más sobre la importancia del perfil comercial en el apartado de oferta.

Haciendo referencia a las tres grandes etapas nombradas al principio, podemos ver la diferencia entre ambos tipos de proyectos:

<b>PROYECTO INTERNO</b>		<b>PROYECTO EXTERNO</b>	
1. Análisis del proyecto	<b>FASE DE PLANIFICACIÓN</b>	1. Elaboración de la oferta	<b>FASE DE REALIZACIÓN</b>
2. Determinación de las opciones existentes		2. Adjudicación del contrato	
3. Selección de la opción más conveniente: formulación		3. Planificación detallada de la obra	
4. Planificación detallada del trabajo a realizar		4. Desarrollo y realización	
5. Desarrollo y realización	<b>FASE DE ENTREGA O PUESTA EN MARCHA</b>		

### Ilustración 1 Diferencias entre tipos de proyectos

Cuando se abordan proyectos grandes y complejos, la consecución del resultado final depende de la realización armónica del conjunto de las etapas pertinentes con ayuda de los medios materiales y humanos requeridos en cada momento. La concepción de las fases que han de ejecutarse, el orden de encadenamiento lógico de las mismas y la estimación de la naturaleza y cantidad de recursos a emplear en cada momento, precisan de un conocimiento profundo de las tecnologías que concurren en el proyecto y de una experiencia que permita prever y superar las dificultades que en la práctica suelen aparecer.

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

A continuación se presentan las distintas etapas en el desarrollo de una aplicación informática:

**Tabla 1 Etapas en el desarrollo de una aplicación**

ETAPA 1	Nacimiento de la idea del proyecto
	El "cliente o promotor" expone sus necesidades y el deseo de resolver el problema por medios informáticos. Se crea un primer documento breve que recoge el anteproyecto y es aprobado por la dirección o el comité correspondiente.
ETAPA 2	Estudio de oportunidad
	El estudio de oportunidad concreta los objetivos y resultado a aportar por el proyecto, los plazos y costes previstos y los medios a emplear.
ETAPA 3	Estudio detallado
	El jefe de proyecto define, ya en detalle, con el apoyo de los técnicos de su equipo, el contenido del proyecto, su análisis funcional, las cargas de trabajos previstos y la metodología a desarrollar.
ETAPA 4	Cuaderno de cargas para informática
	El estudio de oportunidad concreta los objetivos y resultado a aportar por el proyecto, los plazos y costes previstos y los medios a emplear.
ETAPA 3	Estudio detallado
	El jefe de proyecto define, ya en detalle, con el apoyo de los técnicos de su equipo, el contenido del proyecto, su análisis funcional, las cargas de trabajos previstos y la metodología a desarrollar.
ETAPA 4	Cuaderno de cargas para informática
	A partir del análisis funcional se determinan en forma definitiva los volúmenes, cargas de trabajo, calendario y medios a utilizar, dando lugar al contrato formal entre cliente, usuarios e informáticos, frecuentemente conocido con el nombre de cuaderno de cargas o, más concretamente, "pliego de especificaciones".
ETAPA 5	Análisis orgánico
	Los técnicos realizan el análisis orgánico y las especificaciones para programación.
ETAPA 6	Programación y pruebas
	Se realiza la programación de la aplicación y las pruebas para programación.
ETAPA 7	Recepción provisional
	Al resultar satisfactorias las pruebas se realizan la recepción provisional, dando lugar a los manuales de usuario y de explotación.
ETAPA 8	Puesta en marcha
	La puesta en marcha de la aplicación es una fase delicada que requiere una estricta vigilancia hasta comprobar su correcto funcionamiento. A continuación se realiza un balance de los resultados del proyecto.
ETAPA 9	Balance de funcionamiento
	Después de varios meses de funcionamiento de la aplicación se debe realizar un balance que permita apreciar los beneficios que realmente ha producido a la empresa.

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

ETAPA 10	Auditoría
	Transcurridos uno o dos años, debe efectuarse una auditoría de la aplicación que permita comprobar si sigue siendo adecuada o si es necesario introducir modificaciones.

### 1.3 Gestión de proyecto

La gestión de proyectos se define como el conjunto de capacidades del liderazgo - organizativa, técnica y financiera- utilizadas para alcanzar los objetivos de un proyecto. En otras palabras, la gestión de proyectos implica la coordinación directa y entre departamentos de los procesos **(1)**

Un principio básico de la gestión de proyectos, así como en toda actividad de gestión, es que los objetivos estén definidos a prioridad y con un grado de suficiente de claridad y precisión. Hay proyectos donde la definición de objetivos se hace realmente difícil, pero esa dificultad no significa que no deba hacerse, puesto que cuanto más inmaterial es o más arriesgado sea un proyecto más necesario será contar con un marco de referencia, aunque sus contornos sean menos nítidos que en otras ocasiones.

### 1.4 Software internacional para la gestión y control de Proyecto

Una de las fases críticas e indispensables en cualquier proyecto es la de la gestión en el tiempo del mismo. El proceso de gestión consiste en elaborar un cronograma con las diferentes tareas a realizar y con los agentes implicados en las mismas. De esta manera podemos resolver riesgos críticos como el de evitar que una persona se encuentre realizando simultáneamente dos tareas o que una tarea comience antes de que una tarea previa e indispensable para esta segunda se haya acabado. Para esta gestión del tiempo, los jefes de proyectos o los gerentes se apoyan habitualmente en herramientas diseñadas para tal efecto. A continuación hago una lista con otros programas similares:

#### Microsoft Project

Es un programa de la suite Microsoft Office usado para la gestión de proyectos. Microsoft Project (MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

El software Microsoft Office Project en todas sus versiones es útil para la gestión de proyectos. Aunque este software ha sido etiquetado como miembro de la familia Microsoft Office hasta el momento no ha sido incluido en ninguna de las ediciones de Office. Está disponible en dos versiones, Standard y Professional. Existen versiones para otros sistemas operativos (SO) como Macintosh, Mac. Las versiones fueron lanzadas en 1992 (v3), 1993 (v4), 1995, 1998, 2000, 2002, 2003 y 2007.

### **OpenProj**

OpenProj, es una alternativa nueva y sólida, desarrollada bajo licencia de código abierto. Asegurándonos, libertad y la garantía que nos ofrece la comunidad GNU.

Es un software libre, y se constituye en una alternativa sólida frente a Microsoft Project. Nos permite hacer casi cualquier actividad que realice su símil de Microsoft. Y es muy recomendado a la hora de reemplazar un software comercial de esta categoría, por uno de licencia GNU. Está disponible para Sistemas Operativos: Windows, Linux, Unix, Mac.

Sus funciones principales, se concentran en el diseño de graficas de Gantt, diagramas de redes PERT. Cabe destacar que tiene dos de las herramientas más comerciales en gestión de proyectos. Por un lado las Gráficas de Gantt, que son una técnica sencilla de control que se puede emplear para preparar la producción programada y la real.

Sin embargo, se considera generalmente que estas gráficas de Gantt son inadecuadas para proyectos sofisticados. Los modelos de redes, como el método PERT, muestran las interrelaciones entre las diversas actividades. La red PERT permite determinar el itinerario crítico de forma que se pueda centrar la atención en este programa, impidiendo un retraso en el tiempo de acabado del proyecto.

OpenProj controla todos los aspectos referentes a la gestión de proyectos, como la planificación y programación, la gestión y asignación de recursos, la simulación de alternativas

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

en procesos críticos; asimismo, proporciona la funcionalidad necesaria para trabajar con entornos multiproyecto.

También nos permite hacer un seguimiento de las distintas etapas de un proyecto: presupuesto, definición, estimación. Cuando un proyecto se termina, existe un histórico donde se ven reflejados cada uno de los anteriores puntos.

Para ampliar su funcionalidad, existe una versión adicional de OpenProj, que incluye características adicionales. Esta versión se denomina, Project-ON-Demand, y comparte el mismo entorno de trabajo. Solo incluye módulos adicionales, que en algunos casos pueden resultar muy convenientes para un mejor manejo del proyecto.

A pesar de no ser un producto comercial, OpenProj, ofrece todas las características necesarias y suficientes para trabajar en nuestros proyectos, OpenProj esta en la posibilidad de trabajar con proyectos de gran envergadura.

OpenProj es una aplicación diseñada casi completamente en java. Es por eso necesario contar con el plugin instalado en el ordenador.

También no ofrece la posibilidad de trabajar con calendarios, en los que se incluirá cada actividad del proyecto. Una herramienta útil e indispensable, que incorporan casi todas las aplicaciones de este tipo.

Con todo el potencial de esta herramienta, podremos detallar cualquier actividad en el tiempo que dure el proyecto, indicar costos; la aplicación se encarga de entregarnos resultados automáticamente por cada actividad.

Una de las cualidades de esta aplicación, es que nos permite abrir archivos de Microsoft Project cómo si se trataran de archivos de la aplicación. La compatibilidad que ofrece con el símil de Microsoft, es sorprendente.

Frente a otras aplicaciones que tratan simplemente de copiar a Microsoft Project, OpenProj destaca por su ligereza de instalación, por ser totalmente libre, por permitir la lectura de ficheros y extensiones de la competencia.

### **Redmine**

Es una herramienta de gestión de proyectos software con interface web. Se trata de otra herramienta web, en este caso desarrollada en Ruby on Rails con una gran cantidad de funcionalidades.

Entre sus características cabe destacar la gestión de tareas, incidencias, diagramas de Gantt, calendario de actividades, noticias, foros, wikis, gestión de ficheros y documentos, integración con Subversión, control de tiempos, generación de informes, todo esto por proyecto.

Una vez instalados los requerimientos mínimos para su utilización se puede dar alta a los proyectos y sus jefes, esto pueden definir los hitos del proyecto y las tareas a realizar para cada uno de estos hitos. Los desarrolladores tienen en su página de entrada una lista de las tareas que tienen asignadas. Es una única lista conjunta de las tareas de todos los proyectos. Según van trabajando en las tareas, pueden ir marcando el tiempo que estiman que les llevará la tarea, el tiempo que han trabajado en ella y el porcentaje que creen que tienen realizado.

Una vez que comienzan las pruebas del software, en redmine también se pueden dar de alta los "bugs" o errores que se encuentren, asignándoselos al desarrollador correspondiente y al hito para el que consideremos que debe estar corregido dicho error.

El Redmine tiene diferentes módulos tales como:

- Wiki por proyecto.
- Foro por proyecto.
- Envío automático de e-mail a los desarrolladores cada vez que se les asigna una tarea o ante cualquier evento relacionado con el proyecto.

- Posibilidad de subir ficheros y documentos, bien al proyecto, bien como adjuntos a las tareas y errores.
- Posibilidad de definir nuevos tipos de tareas y errores, con campos personalizados, todo ello fácilmente a través de la interface web. Estas tareas personalizadas y campos personalizados se asignan por proyecto, por lo que unos proyectos pueden tener algunas de esas tareas y campos y otros no.
- Se puede ver a través de redmine los cambios en el repositorio. Entiende CVS, Subversión y algunos de los sistemas de control de versiones más conocidos.
- Gráficos de Gantt, consultas por filtro con posibilidad de salvar dichas consultas, proyectos con sub-proyectos.

### **DotProject**

DotProject fue creado en el año 2000, con el fin de construir una herramienta para la Gestión de Proyectos. DotProject esta construido por aplicaciones de Código abierto y es mantenida por un pequeño pero dedicado grupo de voluntarios. Es una aplicación basada en web, multiusuario, soporta varios lenguajes y es Software libre.

Esta programada en PHP, y utiliza MySQL como 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 de funcionalidad orientada a la Gestión de Proyectos. Construir una herramienta con una interfaz de usuario simple, claro y consistente. Ser de código abierto, libre acceso y utilización.

El software es libre para todo quien quiera descargarlo. Además existe soporte gratuito provisto por voluntarios en los foros de la comunidad y vía la documentación online.

DotProject es una herramienta orientada a la Gestión de Proyectos. Para eso se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente.

La aplicación consta de un grupo de módulos ordenadas jerárquicamente las cuales permiten brindar la funcionalidad del producto, 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 loguearse a DotProject 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 de 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.

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

- **Foros:** Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.
- **Administración del Sistema:** Contiene la actividades relacionadas a la administración de usuarios, roles y configuración del sistema.
- **Recursos:** Permite asignar recursos no humanos (oficinas, equipamiento) a un proyecto.

Sin querer realizar una comparativa exhaustiva entre las distintas opciones contempladas, mencionaremos alguno de los motivos por los que se decidió usar DotProject frente a otros sistemas similares:

- Las ventajas de DotProject frente a productos comerciales como MSProject no se limitan al aspecto económico. En DotProject la programación de addons es muy sencilla de realizar, ya que existe la posibilidad de tomar todo el código del programa y modificarlo.
- En la Facultad 6 la mayoría de los proyectos utilizan el Dotproject como herramienta para la planificación de proyecto.
- El hecho de que el programa sea gratuito es ventajoso no sólo desde el punto de vista económico, sino que también como una reducción del riesgo que supone una eventual adaptación, especialmente si se lleva a cabo como acuerdo de prácticas en la UCI como es el caso.
- A propósito de los riesgos de implantación, el hecho de que toda la base de datos emplee MySQL asegura tanto la portabilidad del contenido registrado en ella en caso de rechazo del sistema una vez implantado, como una garantía de robustez gracias a las facilidades de copias de seguridad y almacenamiento distribuido y replicado que ofrece este sistema.
- El concepto basado en Web se ajusta muy bien al requerimiento presente, ya que es genérico para cualquier proyecto de desarrollo de software.

- La GUI de DotProject es muy sencilla, intuitiva y suficientemente atractiva, en especial en lo relativo a las gráficas de Gantt, que permiten visualizar esquemáticamente y de forma clara la planificación de las tareas de cada proyecto.

### **1.5 Herramientas propuestas para el desarrollo del sistema**

Con el objetivo de desarrollar un sistema óptimo se llevo a cabo un exhaustivo estudio de las diversas metodologías, tendencias y tecnologías que se utilizan en la actualidad para desarrollar aplicaciones WEB en el mundo, definiéndose las siguientes propuestas.

Cuando desarrollamos un programa, lo desarrollamos para que funcione en una plataforma específica. Esto se refiere a que el programa va a funcionar en un determinado sistema operativo.

Normalmente, hay versiones para Windows, MacOs, Linux. Pero cuando nos referimos a una plataforma web, estamos diciendo que nuestro sistema funcionará desde un navegador web, independientemente del SO que se esté utilizando.

Esto tiene la ventaja de que no es necesario crear el mismo software para cada tipo de SO. Pero tiene desventajas en el sentido de que no puede ser tan potente como un sistema que funciona usando los recursos del SO.

Existe multitud de plataformas o conjunto de subsistemas para crear y poner en funcionamiento sitios web. La elección de unos u otros dependerá de varios factores como precios, velocidad, flexibilidad, soporte, entre otros.

Las plataformas presentadas a continuación son las más comunes, pero muchas más combinaciones para crea un sistema que sirva como páginas web.

#### **LAMP(Linux, Apache, MySQL)**

Esta plataforma web promocionado por el editor de la editorial O'Reilly Dale Dougherty y algunos autores de MySQL es el más popular y el que podrás encontrar más fácilmente en la mayoría de los proveedores de Hosting.

### **Ventajas:**

- Disponibilidad y bajos costos
- Velocidad y mejor tiempo de actividad.
- Es muy fácil colocar instrucciones tanto en nuestro código para activar nuevas funcionalidades del servidor.
- Puede utilizar un sistema gestor de contenido (CMS), los mejores como Drupal, Joomla y Wordpress
- Servidor de bases de datos relacional MySQL muy rápido para consultas de lectura y sitios no demasiado grandes.

### **Desventajas:**

- No muy buen rendimiento de MySQL para sitios web demasiado grandes y con un gran volumen de inserción de datos.
- Es muy distinto de Windows, lo que dificulta el trabajo a quienes estén acostumbrados a él.

### **WAMP(Windows, Apache, MySQL)**

Es una herramienta de desarrollo por la mayoría de los programadores que tienen Windows como SO principal y tenemos que desarrollar aplicaciones web en PHP.

Es decir, probamos nuestra aplicación PHP en el servidor de desarrollo WAMP y luego subimos la aplicación al servidor LAMP

Aunque a nivel de programación no existen diferencias con un sistema LAMP, si que existen diferencias a nivel de administración.

Existe un software llamado WAMP5 que instala de forma compacta y configura fácilmente toda la plataforma WAMP.

### **XAMP (X=Cualquier S.O, Apache, MySQL, PHP)**

XAMPP podría decirse que más que una plataforma web es un software que instala de forma compacta y configura fácilmente PHP, MySQL y Apache para Windows, Linux, Sun Solaris y Mac OS X.

También se puede decir entonces que WAMP es una variante de XAMPP.

### **1.6 Lenguaje de programación: PHP**

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt (biblioteca multiplataforma para desarrollar interfaces gráficas de usuario) o GTK+( GTK+ o The GIMP Toolkit es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario (GUI)).

PHP significa **PHP Hypertext Pre-processor** (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la licencia PHP, la Free Software Foundation (fundación de software libre) considera esta licencia como software libre. **(2) (3)**

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, La versión más reciente de PHP es la 5.2.9-1 (Windows) del 10 de Marzo de 2009.

### Ventajas

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados exts o extensiones).
- Posee una amplia documentación en su página oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

### Desventajas

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes

### 1.7 Sistemas gestores de base de datos (SGBD)

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad.

Por tanto debe permitir:

- **Definir una base de datos:** especificar tipos, estructuras y restricciones de datos.
- **Construir la base de datos:** guardar los datos en algún medio controlado por el mismo SGBD
- **Manipular la base de datos:** realizar consultas, actualizarla, generar informes.

### **Gestor de base de datos: MySQL**

El principal objetivo que se tuvo en cuenta para su diseño fue la velocidad., por lo que sacrificó algunas características que eran esenciales para sistemas más completos. Otra funcionalidad que lo vuelve útil es que consume muy pocos recursos, tanto de memoria como de CPU.

#### **Ventajas:**

- Mayor velocidad tanto al conectar con el servidor como al servir diversas consultas
- Mejores utilidades de administración (backup, recuperación de errores).
- Aunque en ocasiones suele colgarse con grandes volúmenes de información, no suele perderla ni corromper los datos.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- MySQL se comporta mejor que otros gestores de Base de datos, como por ejemplo, Postgres, a la hora de modificar o añadir campos a una tabla.

### **1.8 Entornos de desarrollo integrado (IDE)**

Un entorno de desarrollo integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

### **Zend Studio**

Zend Studio o Zend Development Environment es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre. (4)

Está disponible también una versión de Zend Studio para Eclipse.

### **Características**

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
- phpDoc integrado.
- Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- Inserción automática de paréntesis y corchetes de cierre.
- Ayudas de formato de código.
- Emparejamiento de paréntesis y corchetes (si se sitúa el cursor sobre un paréntesis (corchete) de apertura (cierre), Zend Studio localiza el correspondiente paréntesis (corchete) de cierre (apertura)).
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada, seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos.
- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox.

- Soporte para gestión de grandes proyectos de desarrollo.
- Manual de PHP integrado.
- Soporte para control de versiones usando CVS o Subversion
- Cliente FTP integrado.
- Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.

### **Eclipse**

Eclipse se basa fuertemente en el uso de plug-ins. Hoy en día, existen plug-ins para hacer casi cualquier cosa, desde crear tests automáticos y generar código, hasta usar un CVS. Si bien se puede programar en distintos lenguajes, el que se utiliza más es, sin dudas, Java. Este sistema se distribuye bajo una licencia Open Source. Sin embargo, hay que considerar que está íntegramente desarrollado en Java, que no es libre, de modo que Eclipse tampoco lo es. Esta plataforma de desarrollo es una de las más utilizadas en la actualidad, ya que no sólo tiene una gran calidad, sino que también es inherentemente útil para programar en Java, que es un lenguaje multiplataforma. Esto nos permite integrar distintos tipos de ambientes (GNU/Linux, Windows, Mac entre otros) de manera casi transparente. El punto más fuerte de Eclipse es, sin duda, su capacidad para extenderse mediante plugins. La idea es, entonces, contar con una herramienta que pueda ser empleada por todos los usuarios, de modo tal que el pasaje de información entre distintas aplicaciones sea transparente. Si todos corren sobre la misma herramienta, es sencillo que dos aplicaciones compartan datos sin mucha dificultad. Dado que este entorno tiene una gran aceptación y una gran cantidad de usuarios, existen plugins para numerosas necesidades.

### **Zend Studio para Eclipse**

Zend Studio para Eclipse es el Entorno de Desarrollo Integrado (IDE) para PHP, ofreciendo al desarrollador profesional de PHP la potencia de Zend Studio y el soporte multilenguaje de Eclipse y su enorme conjunto de extensiones (plugins).

Zend Studio para Eclipse para mejorar la calidad de los proyectos en PHP, agilizar los ciclos de desarrollos y simplificar la complejidad de los proyectos. El plugin incluye herramientas para edición, debugging, análisis, optimización y bases de datos, e incluso soportando los procesos del desarrollo por programación ágil. **(5)**

En concreto, Zend Studio para Eclipse aporta:

### **Las ventajas de Eclipse/PDT:**

- Una comunidad de millones de usuarios y miles de desarrolladores
- Cientos de plugins
- Soporte multi-lenguaje en una única herramienta (Eclipse)
- Coloreado de sintaxis PHP, autocompletado de código e inspección de métodos y atributos
- Soporte básico de depuración de scripts PHP

### **Valor añadido de Zend Studio:**

- Soporte de depuración más avanzada (permitiendo debugging local como remoto)
- Soporte a la refactorización de código
- Análisis de rendimiento y conversión de código
- Soporte de baterías de test PHP Unit
- Integración mejorada con Zend Framework
- Posibilidad de edición HTML WYSIWYG
- Integración con Zend Platform
- Herramientas de migración para los usuarios de Zend Studio
- Instalación simplificada

### **1.9 Lenguaje de Modelación UML (Unified Modeling Language)**

El Lenguaje Unificado de Modelación (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema. El UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. UML no es método.

Es importante recalcar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. **(6)**

Con UML se realizan los diseños previos a la codificación del sistema ya que este se encarga de especificar, visualizar y documentar los componentes de un sistema siempre y cuando posea desarrollo orientado a objetos. Pese a que UML es un lenguaje este posee más características visuales que características de programación, las cuales facilitan a los integrantes del equipo mantener una constante comunicación. Las herramientas Case, tales como el Visual Paradigm y el Rational Rose lo han adoptado como lenguaje de modelado.

En este trabajo se utilizara el Visual Paradigm por ser una herramienta desarrollada en software libre. RUP es una metodología que se utiliza para el análisis diseño, implementación y documentación de los sistemas realizados cuya característica fundamental es que son orientados a objetos .Es una metodología que se adapta al entorno, es decir sus utilidades se moldean de acuerdo al contexto y la necesidad de cada organización en específico.UML y RUP están estrechamente relacionados, ya que el primero define los diagramas y el segundo los pasos.

### **Visual Paradigm para UML**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

### **Lista de características:**

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio, proceso, decisión, actor de negocio

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

- Modelado colaborativo con CVS y Subversion
- Interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse) a través de XMI
- Ingeniería de ida y vuelta
- Ingeniería inversa
- Código a modelo, código a diagrama
- Ingeniería inversa Java, C++, Esquemas XML, XML,.NET
- Generación de código
- Modelo a código, diagrama a código
- Editor de Detalles de Casos de Uso
- Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos
- Soporte ORM - Generación de objetos Java desde la base de datos
- Generación de bases de datos
- Transformación de diagramas de Entidad-Relación en tablas de base de datos
- Ingeniería inversa de bases de datos. Desde sistemas gestores de bases de datos existentes a diagramas de Entidad-Relación
- Generador de informes para generación de documentación
- Distribución automática de diagramas
- Reorganización de las figuras y conectores de los diagramas UML
- Importación y exportación de ficheros XMI
- Integración con Visio
- Dibujo de diagramas UML con plantillas (stencils) de MS Visio
- Editor de figuras

### ¿Qué metodología de desarrollo debo utilizar para el desarrollo del software?

Se debe tener en cuenta además utilizar una metodología adecuada, es decir que sea la más apropiada para el caso que vaya a desarrollar. Para dar una idea de qué metodología podemos utilizar y cuál se adapta más a nuestro medio, mencionaré tres de ellas de las que se consideran las más importantes, tal como: RUP, XP y MSF.

#### 1.10 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. **(6)**

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 óptima.

**Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios reléase del producto que han pasado las pruebas. Se ponen estos reléase a consideración de un subconjunto de usuarios.

**Transición:** El reléase 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.

### **1.11 Arquitectura del software.**

La Arquitectura del Software es un conjunto de decisiones y especificidades sobre la forma a conformar un sistema, también de como estructurarlo y la determinación de como es que se va a concebir mediante las interfaces que lo conformaran y su comportamiento. La arquitectura hace un estudio de las bases fundamentales del sistema que se convierten en los pilares iniciales para poder desarrollarlo, comprenderlo además de producirlo económicamente.

La arquitectura del software establece los fundamentos para que los analistas, diseñadores, programadores, entre otros, trabajen en una línea que permita alcanzar los objetivos del sistema de información. Define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación ente ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad y disponibilidad.

Una de los aspectos fundamentales dentro de la arquitectura de software son los estilos arquitectónicos. Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software, mientras que las formas complejas se articulan mediante composición de los estilos fundamentales.

Un estilo arquitectónico o variante arquitectónica define a una familia de sistemas informáticos en términos de su organización estructural. Un estilo arquitectónico describe componentes y las relaciones entre ellos con las restricciones de su aplicación, la composición asociada y el diseño para su construcción.

### **1.12 Arquitectura en Capas**

El estilo de Arquitectura de Capas, define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo que significa que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores.

Existen distintas variantes del estilo capas como son arquitecturas de dos capas, arquitecturas de tres capas y arquitecturas de n capas. La variante utilizada en este trabajo es la de

arquitectura de tres capas, la misma fue escogida por las condiciones y estructura que presenta dicho sistema, esta arquitectura permite implementar componentes de una manera más flexible sin dejar de señalar que es una de las arquitecturas más complejas, todas las peticiones del cliente se controlan en la capa correspondiente a la lógica del negocio.

Pues el cliente y el gestor de reglas de negocio tienen que hablar el mismo lenguaje, el gestor de reglas de negocio y el servidor de datos tienen que hablar el mismo lenguaje, esto conlleva consigo que se logren beneficios como ya sea separar las reglas de negocio de los interfaces especialmente en los entornos multiplataforma permite que las reglas se cambien con un mínimo impacto sobre los usuarios de las aplicaciones, posibilita integrar aplicaciones que accedan a las mismas base de datos de una forma sencilla, en fin el uso de modelos tres capas aumenta la flexibilidad a la hora de aplicar las posibilidades de la informática para aspectos específicos de la problemática del cliente.

### **Descripción de las distintas capas**

**La Capa de Presentación:** esta capa reúne todos los aspectos del software que tiene que ver con las interfaces y la interacción con los diferentes tipos de usuarios, estos aspectos típicamente incluyen el manejo y aspecto de las ventanas, menús, gráficos entre otros.

**La Capa del Negocio:** Esta capa reúne todos los aspectos del software que automatizan o apoyan los procesos de negocio que llevan a cabo los usuarios. Estos aspectos típicamente incluyen las tareas que forman parte de los procesos, las reglas y restricciones que aplican. Esta capa también recibe el nombre de la capa de la Lógica de la Aplicación.

**La Capa de acceso a datos:** Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes, por lo que también se le denomina la capa de las Bases de Datos.

### **Conclusiones**

En este capítulo se pudo constatar que ninguno de los sistemas para la gestión de software garantiza la distribución de los recursos según las tareas, además de no crear un horario de trabajo con respecto a las tareas de un proyecto. Además, se ve comprueba la necesidad en desarrollar un software que permita mantener un control del aprovechamiento del tiempo de un estudiante en un proyecto productivo. Se estudio Dotproject como herramienta utilizada actualmente en la facultad 6 para la gestión de software. Se realizo un estudio sobre las principales herramientas que se utilizaran para desarrollar este software, se determino la más apropiada a utilizar para el desarrollo de los módulos.



## **CAPÍTULO 2** **Descripción de la propuesta de la solución**

### **Análisis de la gestión de software del proceso de la asignación de horario y recursos en proyecto.**

#### **Introducción**

En este capítulo se realiza un análisis del proceso que se lleva a cabo para realizar la planificación del horario en proyectos, facilitando una mayor comprensión del problema presente. Se presenta el modelo del negocio con el objetivo de entender el plano en que se emplaza el sistema y por tanto contribuir a la comprensión de los requisitos que se desprenden de este contexto. Se determinan los actores y trabajadores del negocio y del sistema; se plantean las principales reglas del negocio a tener en cuenta en el sistema a desarrollar; se muestran los casos de uso del negocio detallados y una representación del modelo de objetos; así como una especificación de los requisitos y casos de usos del sistema generados a raíz de la captura de requisitos.

#### **2.1 Modelo de Dominio**

Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema.

El modelo del dominio se considera en RUP un subconjunto del llamado modelo de objetos del negocio. Se puede desarrollar un modelo de objetos del negocio enfocado a la explicación de los productos, entregables o eventos que son importantes en el negocio. Esos modelos,

que no incluyen las responsabilidades de las personas que ejecutan las actividades, se refieren a veces como modelo del dominio. **(7)**

### **2.2 Modelo de negocio**

Si los procesos están claramente definidos y no se van a introducir cambios entonces sólo es necesario modelar el negocio propuesto. En este escenario basta con conocer el mapa de la organización y los procesos para comprender mejor los requerimientos de la aplicación a construir. En este caso no se pretende cambiar la organización; en realidad la implantación del sistema siempre incluye algún nivel de mejora del negocio. **(7)**

#### **¿Por qué realizar el Modelo de negocio?**

Después de haber definido el concepto de modelo de dominio y negocio llevamos a la conclusión de que con el trabajo se debe modelar el negocio, pues están claramente definidos los procesos.

Se puede determinar con claridad los pasos a seguir para conocer el proceso que se lleva en la realización del horario y la distribución de los recursos, ya sea el personal a trabajar en un proyecto así como los software que se necesitan para la realización de las tareas del mismo.

### **2.3 Descripción del negocio**

En la facultad 6 para el trabajo en los proyectos se necesita tener un horario elaborado donde el jefe de proyecto pueda organizar el trabajo según las tareas, para el cumplimiento del tiempo de trabajo en el proyecto.

El jefe de proyecto le pide al planificador del proyecto que organice las tareas y asigne los recursos según la disponibilidad de los mismos. Para ello, el planificador del proyecto cuenta con un horario previamente realizado por la planificadora de la facultad. El planificador del proyecto tiene un listado de los recursos asignados a su proyecto tanto humano como material y una lista de tareas para la organización del plan de trabajo de los miembros del proyecto.

El planificador asigna los recursos materiales a los miembros del proyecto, y asigna tareas. También asigna software a las tareas para el mejor cumplimiento de las mismas y para

mantener una uniformidad, ya que los proyectos constan de un grupo de trabajo que laboran en horarios distintos.

Después de asignados los recursos, el planificador del proyecto asigna un horario de trabajo según las tareas asignadas a cada miembro del proyecto, dándole una mejor organización y rapidez al trabajo.

### **2.4 Regla del Negocio**

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

En el sistema se han definido las reglas del negocio que se mencionan a continuación:

- El jefe de proyecto es el encargado de entregarle al planificador del proyecto el horario.
- En el Dotproject deben estar definida los siguiente aspectos:
  - Para poder hacer uso de los módulos usted debe estar conectado en el Dotproject
  - Debe tener definidas las Tareas, usuarios y proyectos en el Dotproject
- No se le asignara una PC a un miembro del proyecto en el horario de clases del mismo
- No se le asignara tiempo de maquina a dos estudiante a la vez en la misma PC
- No se asignara tarea sino existe PC asignada
- No se asignara un horario si no existe tarea
- No se asignara horario sino existe miembros del proyecto

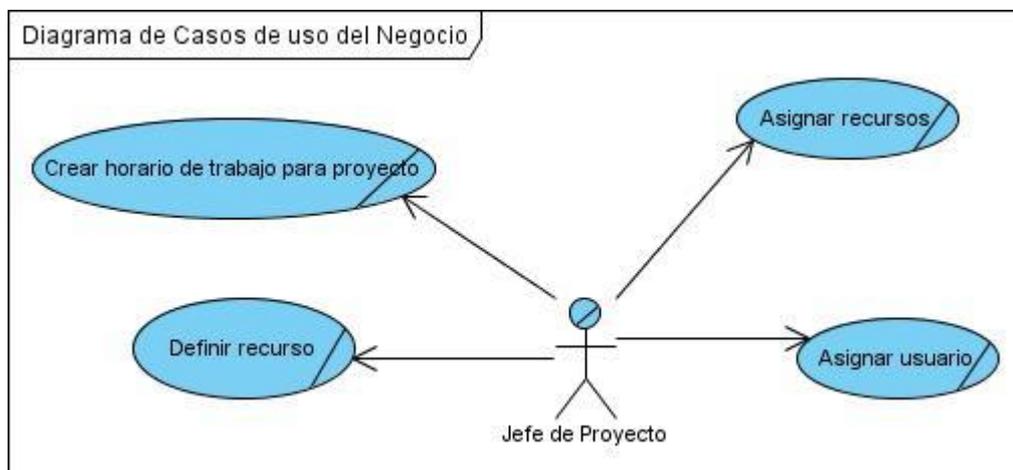
### **2.5 Actor del Negocio**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados. **(9)**

**Tabla 2 Actor del negocio**

Actores del Negocio	Justificación
Jefe de proyecto	El Jefe del Proyecto es el encargado de lograr que se obtenga la optimización del trabajo realizado por el planificador del proyecto, logrando así que se entregue en tiempo el proyecto. Por lo que se beneficia del proceso del negocio.

**Diagrama de Caso de Uso del Negocio**



**Ilustración 2 Diagrama de Caso de Uso del Negocio**

## 2.6 Trabajadores del Negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades. Representa un rol.

**Tabla 3 Trabajadores del Negocio**

Trabajadores del Negocio	Justificación
Jefe de Polo	El jefe de polo es el encargado de asignar el laboratorio y la cantidad de recursos (PC), no se beneficia directamente del proceso de negocio.
Planificadora de la Facultad	La Planificadora de la Facultad es la encargada de facilitar el horario de clase de los miembros del proyecto, no se beneficia directamente del proceso de negocio.
Planificador del Proyecto	El planificador del proyecto es el encargado de distribuir los recursos a los miembros del proyecto. No se beneficia del proceso del negocio.

## 2.7 Casos de uso del Negocio

La Realización de un Caso de Uso de negocio muestra como colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Se considera que con una descripción textual y los diagramas de actividades, es suficiente para describir completamente el proceso de negocio

Se definieron los siguientes casos de uso del negocio:

- Definir recursos
- Definir usuarios
- Asignar recursos
- Crear horario de trabajo para proyecto

**Tabla 4 Descripción de casos de uso: Definir Recursos**

<b>Nombre del Caso de Uso</b>	Definir Recursos
<b>Actores</b>	Jefe de proyecto (Inicia)
<b>Propósito</b>	Permitir al Jefe de proyecto conocer los recursos asignados.
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe de proyecto tiene en sus manos la cantidad de PC asignadas en un laboratorio.
<b>Curso Normal de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

1. El Jefe de proyecto solicita que enumeren las PC del laboratorio asignado para su proyecto.	1.1 El planificador del proyecto enumera las PC por orden de ubicación.
2. El Jefe de proyecto solicita que le asignen un IP fijo según la posición de las PC.	2.1 El planificador del proyecto asigna un IP fijo por ubicación.
3. El Jefe de proyecto solicita el listado de las tareas del proyecto.	3.1 El planificador del proyecto entrega el listado de tareas del proyecto.
4. El Jefe de proyecto depura los Software a utilizar por tareas.	4.1 El planificador del proyecto asigna el software a las tareas del proyecto.
<b>Curso Alternativo de los eventos</b>	
Acción 2.1	En caso de que exista un IP duplicado en el laboratorio se le asigna otro IP a la PC.
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	
<b>Otros</b>	

**Tabla 5 Descripción de casos de uso: Definir Usuario**

<b>Nombre del Caso de Uso</b>	Definir Usuarios.
<b>Actores</b>	Jefe de proyecto (inicia)
<b>Propósito</b>	Permitir al Jefe de proyecto tener un control de los miembros del proyecto.
<b>Resumen</b>	El caso de uso se inicia cuando al Jefe de proyecto necesita definir los usuarios por definir los recursos y tareas.
<b>Curso Normal de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>
1. El Jefe de proyecto solicita listado de miembros del proyecto para trabajar en el proyecto.	1.1 El jefe del polo de la facultad le entrega un listado de los miembros del proyecto.
2. El jefe de proyecto recoge el listado de miembros del proyecto.	2.1 El planificador del proyecto recibe el listado de los miembros del proyecto.
<b>Curso Alternativo de los eventos</b>	
Acción 2.1	En caso de que no exista algún miembro del proyecto en el listado lo incorporamos.
Acción 3.1	En caso de que algún miembro del proyecto no tenga usuario, solicitar la creación del mismo.

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Prioridad</b>	Crítico
<b>Mejoras</b>	
<b>Otros</b>	

**Tabla 6 Descripción de casos de uso: Asignar Recurso**

<b>Nombre del Caso de Uso</b>	Asignar recursos PC
<b>Actores</b>	Jefe de proyecto (inicia)
<b>Propósito</b>	Tiene como propósito asignar un Recurso.
<b>Resumen</b>	El caso de uso se inicia cuando al Jefe de proyecto asigna una cantidad de recursos determinada a los miembros del proyecto y a las tareas.
<b>Curso Normal de los eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del proceso de negocio</b>
1. El Jefe de proyecto solicita al planificador del proyecto que se distribuyan las PC a los miembros del proyecto.	1.1 El planificador del proyecto asigna las PC a miembros del proyecto.
2. El jefe del proyecto solicita al planificador de proyecto que asigne los recursos a las tareas asignadas.	2.1 El planificador el proyecto asigna el software correspondiente a las tareas.
<b>Curso Alternativo de los eventos</b>	
Acción 2.1	En caso de que no haya tarea asignada, el jefe del proyecto solicitara al planificador del proyecto asignar una tarea.
<b>Prioridad</b>	Crítico
<b>Mejoras</b>	
<b>Otros</b>	

**Tabla 7 Descripción de casos de uso: Crear horario de trabajo para proyecto**

<b>Nombre del Caso de Uso</b>	Crear horario de trabajo para proyecto
<b>Actores</b>	Jefe de proyecto (inicia)
<b>Propósito</b>	Permitir al Jefe de proyecto distribuir las tareas y PC por horario.
<b>Resumen</b>	El caso de uso se inicia cuando el Jefe de proyecto necesita distribuir las tareas y PC según el horario de clases.
<b>Curso Normal de los eventos</b>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Acciones del Actor</b>		<b>Respuesta del proceso de negocio</b>
1. El Jefe de proyecto solicita a la planificadora de la facultad el horario de clases de los miembros del proyecto.		1.1 La planificadora de la facultad informa el horario de clases de los miembros del proyecto.
2. El Jefe de proyecto recibe el horario de clases.		
3. El Jefe de proyecto le entrega a el planificador del proyecto el horario de clases		3.1 El planificador del proyecto recibe el horario de clases
4. El Jefe del proyecto solicita al planificador del proyecto que se le asignen las tareas a los miembros del proyecto.		4.1 El planificador del proyecto asigna las tareas.
6. El Jefe del proyecto solicita al planificador del proyecto que se ubiquen a los miembros del proyecto en una PC.		6.1 El planificador del proyecto ubica a los miembros del proyecto en la PC según el horario disponible.
<b>Curso Alternativo de los eventos</b>		
Acción 3.1		En caso de que no se informe del horario de clases de los miembros del proyecto no se podrá conocer el horario de proyecto de los estudiantes por lo que no se pudra planificar el tiempo de trabajo en el proyecto.
Acción 6.1		En caso de no tener PC libre no se asignara el recurso.
<b>Prioridad</b>	Crítico	
<b>Mejoras</b>		
<b>Otros</b>		

2.8 Diagramas de Casos de Usos del Negocio

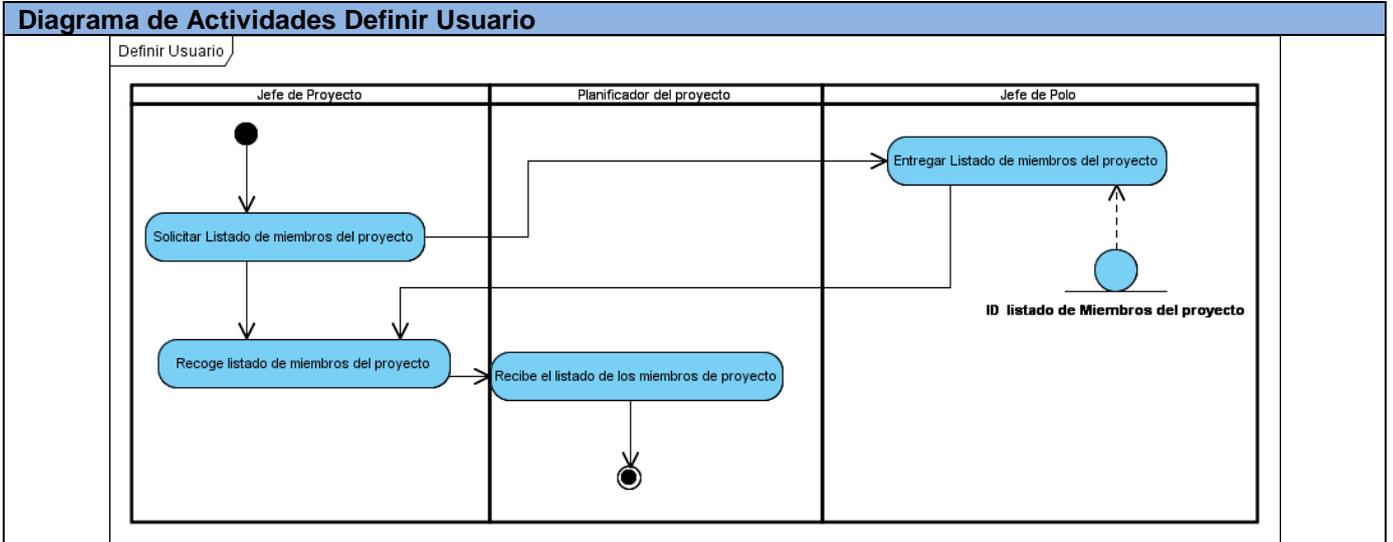


Ilustración 3 Definir Usuario

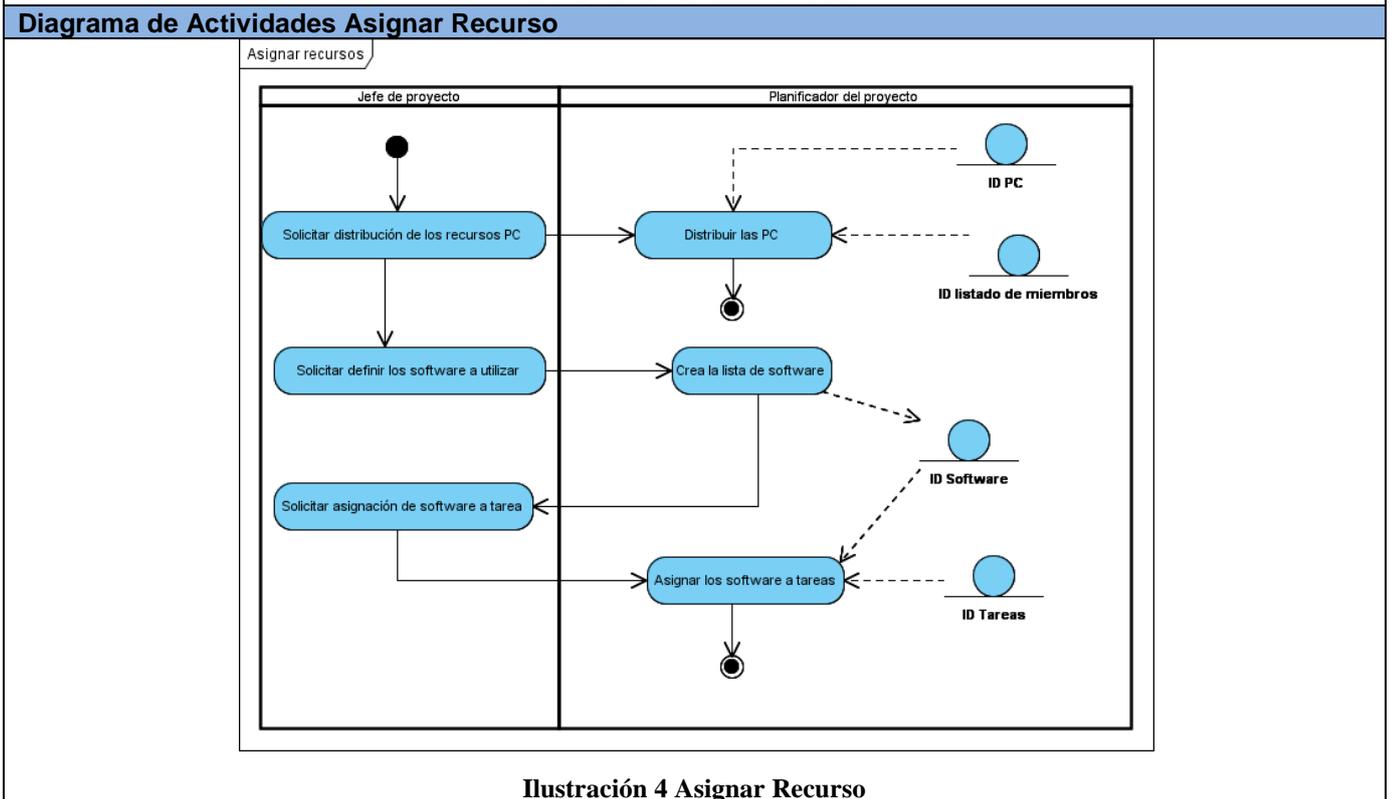


Ilustración 4 Asignar Recurso

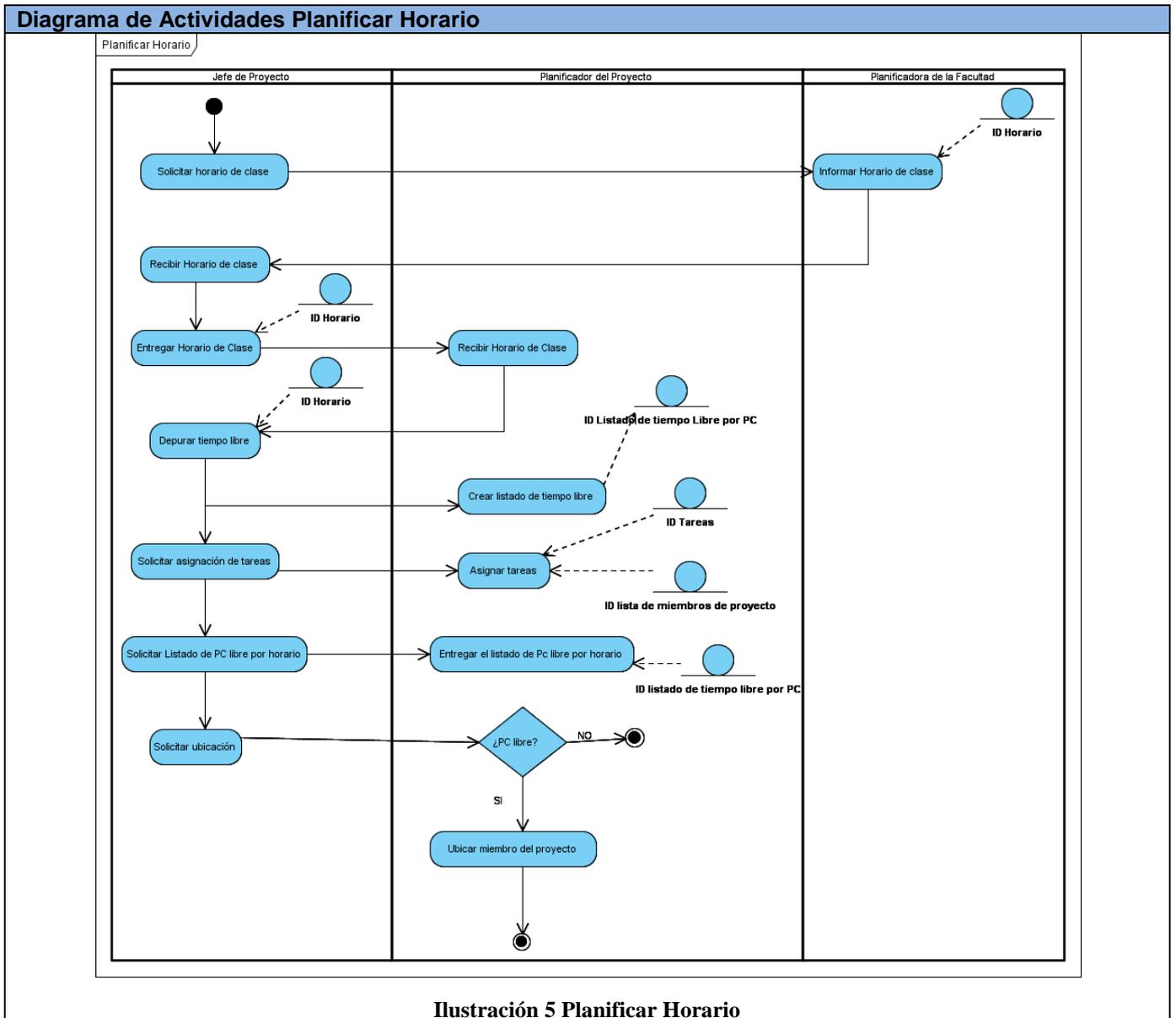


Ilustración 5 Planificar Horario

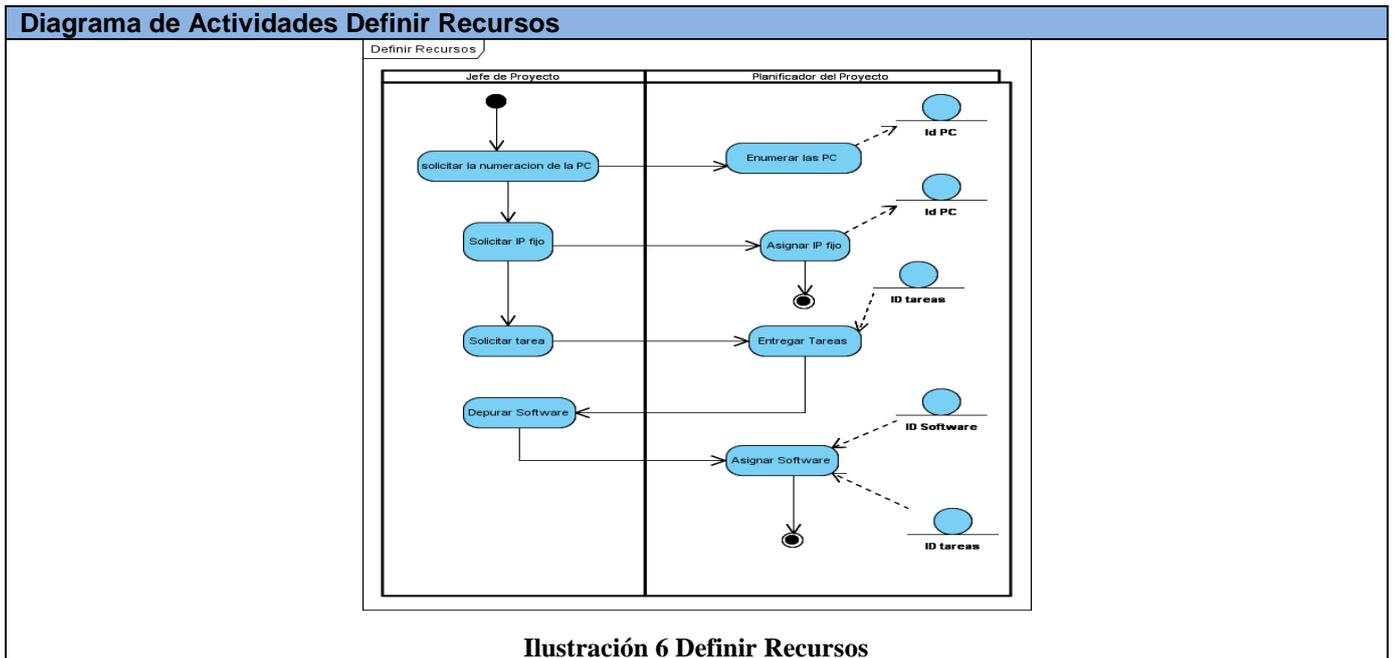


Ilustración 6 Definir Recursos

## 2.9 Entidades del Negocio

Las entidades de negocio representan a los objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio. Comúnmente representan un documento o una parte esencial de un producto.

- PC
- Software
- Horario
- Tareas
- Listado de tiempo libres por PC
- Listado de Miembros del proyecto
- Listado de Asistencia
- Listado de PC Asignada.
- Listado de horario asignado por PC

## 2.10 Modelo de Objetos del Negocio

El modelo de objetos de negocio describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

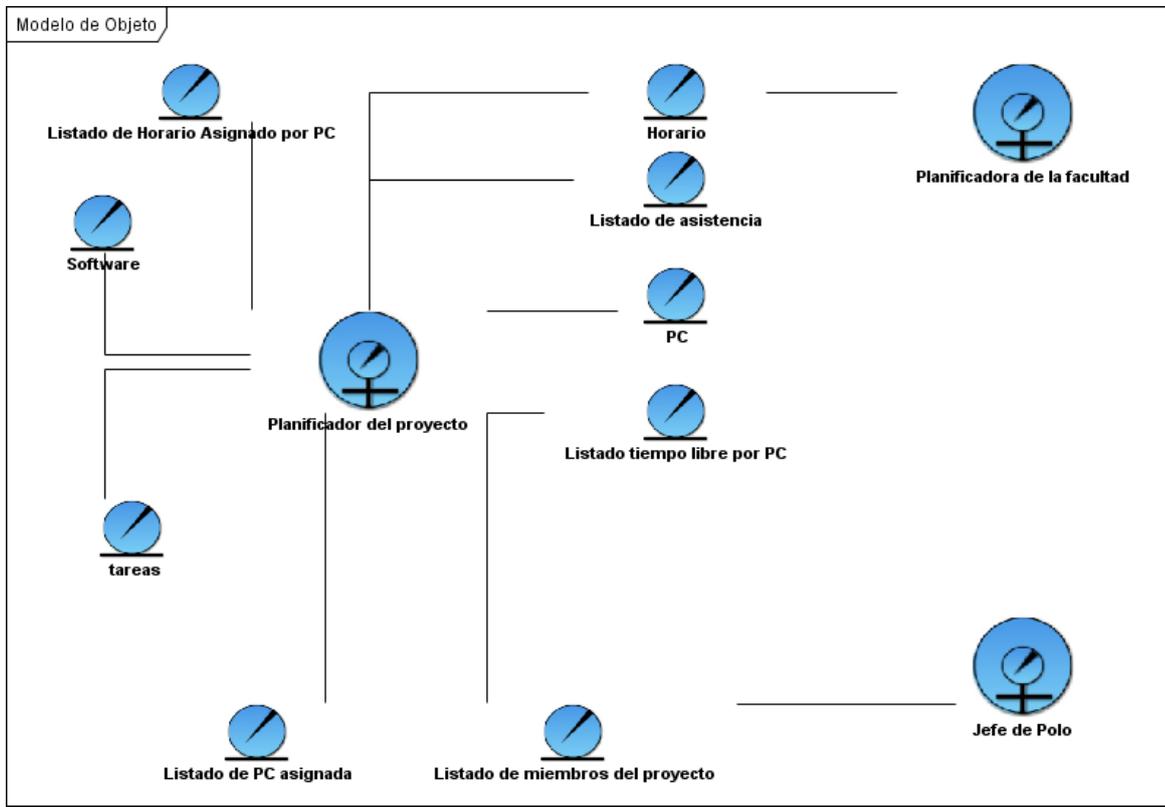


Ilustración 7 Modelo de Objeto

## 2.11 Requisitos

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 pueden clasificar 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 que propiedades o cualidades se relacionen. **(8)**

### 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. **(8)**

### Requisitos Funcionales

RF 1. Agregar Software.

RF 1.1 Generar código del Software.

RF 2. Buscar Software.

RF 2.1 Lista software

RF 3. Eliminar software

RF 4. Modifica software

RF 5. Agregar PC

RF 5.1 Generar código de la PC.

RF 6. Eliminar PC

RF 7. Modifica PC

RF 8. Buscar proyecto

RF 8.1 Listar proyecto

RF 9. Buscar Tareas de un proyecto

FR 9.1 Listar tareas del proyecto

RF 10. Agregar asignación de software

RF 10.1 Generar código de asignación de software

RF 11. Buscar usuario.

RF 11.1 Mostrar la lista de usuario

RF 12. Buscar PC

RF 12.1 Listar PC

RF 13. Agregar PC a usuario

RF 13.1 Generar código de asignación de PC a usuario

RF 14. Buscar PC asignada en el proyecto.

RF 14.1 Listar PC asignada en el proyecto

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

RF 15. Buscar tarea por usuario.

RF 15.1 Listar Tareas de usuario

RF 16. Buscar PC asignada a usuario.

RF 16.1 Listar PC asignada a usuario

RF 17. Eliminar horario.

RF 18. Buscar horario.

RF 18.1 listar horario

RF 19. Modificar horario

RF 20. Agregar horario

RF 20.1 Generar código de asignación de software

RF 21 Mostrar recursos

### **Requisitos no Funcionales**

#### **Usabilidad**

El sistema debe ser amigable al usuario permitiendo al mismo un fácil manejo, debe mantener la misma estructura del Dotproject para su mejor comprensión.

#### **Fiabilidad**

Disponible el 100% del tiempo de utilización El sistema debe estar disponible para su utilización cada vez que el usuario lo requiera. Solamente debe cesar la disponibilidad cuando sea necesario realizar mantenimiento.

El tiempo medio de reparación no debe exceder las 4 horas Después de una falla del sistema esta se debe corregir en menos de 4 horas.

Debe garantizarse el resguardo de la información, de modo que estén duplicados, así como la grabación periódica de la Base de Datos.

#### **Eficiencia**

El tiempo de respuesta por transacción no debe exceder 1 seg.

El sistema debe realizar una transacción sin exceder el tiempo máximo de 1 seg.

### **Soporte**

El sistema debe ser compatible con diferentes versiones del Dotproject.

Interfaz de usuario

La interfaz de usuario debe ser amigable.

La aplicación será diseñada con una interfaz amigable, fácil de usar por el usuario, de manera que agilice y facilite el trabajo con el software.

El sistema debe tener el mismo diseño del Dotproject.

### **Portabilidad**

El sistema debe ser multiplataforma, debido a que será capaz de ejecutarse sobre diferentes sistemas operativos sin importar sus versiones, y sin necesidad de modificar su código fuente.

Al sistema se podrá acceder a través de cualquier navegador.

### **Seguridad**

El sistema debe restringir la autenticación de usuarios referente al Dotproject.

El sistema debe seguir las políticas de seguridad del Dotproject.

### **2.12 Actor del sistema**

Un actor puede representar personas externas al sistema que de alguna forma interactúan con él.

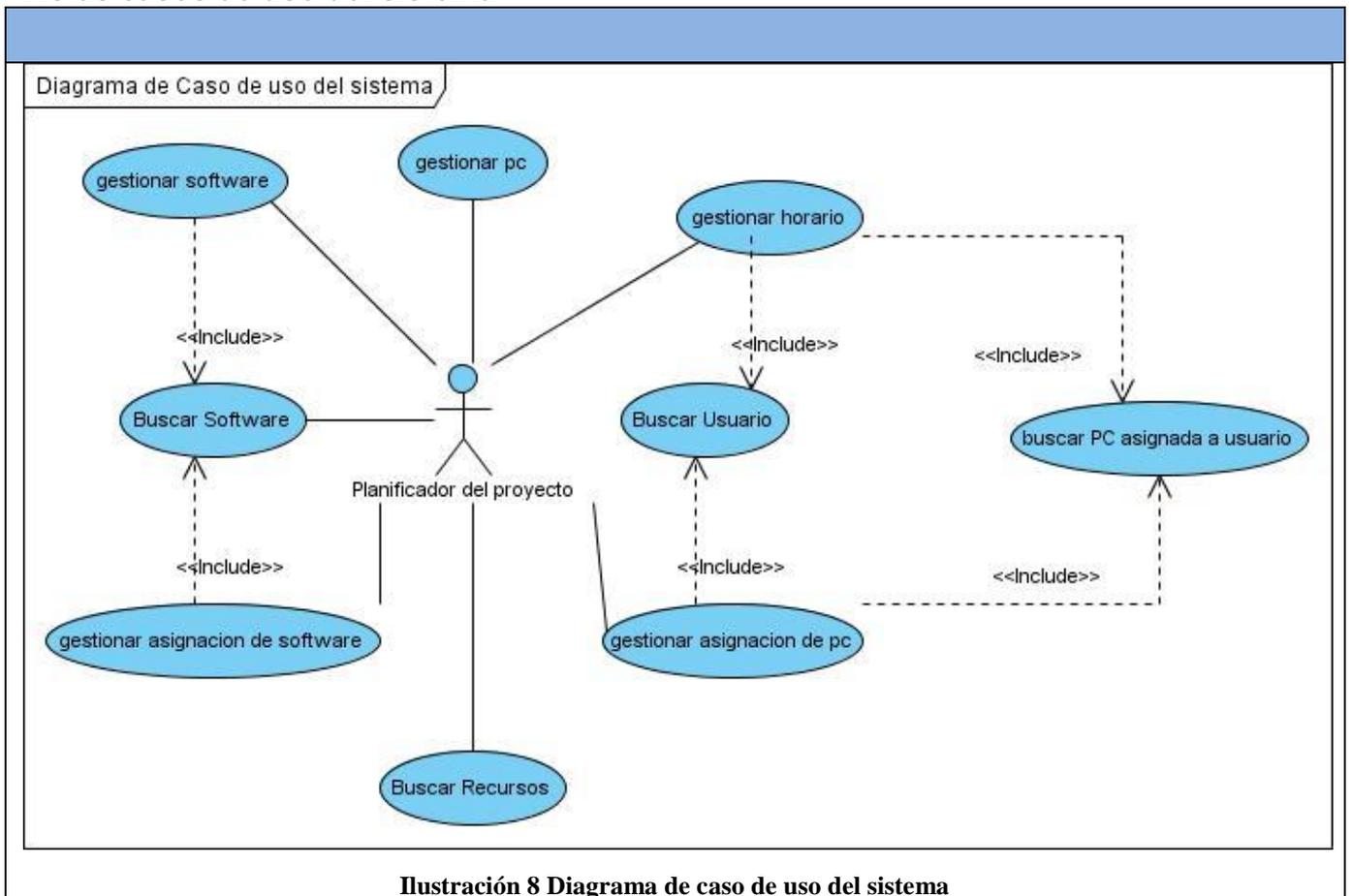
**Tabla 8 Actor del sistema**

<b>Actores del Sistema</b>	<b>Justificación</b>
Planificador del proyecto	Es el responsable de asignar los recursos de los miembros del proyecto, PC y Software contra tareas, así como asignar un horario de trabajo según el tiempo libre.

### 2.13 Caso de uso del sistema

- Gestionar Software.
- Gestionar PC.
- Gestionar Horario.
- Gestionar asignación de Software.
- Gestionar asignación de PC.
- Buscar Software
- Buscar Usuario
- Recursos

#### Vistas casos de uso del sistema



**2.14 Descripción textual de los casos de uso del sistema**

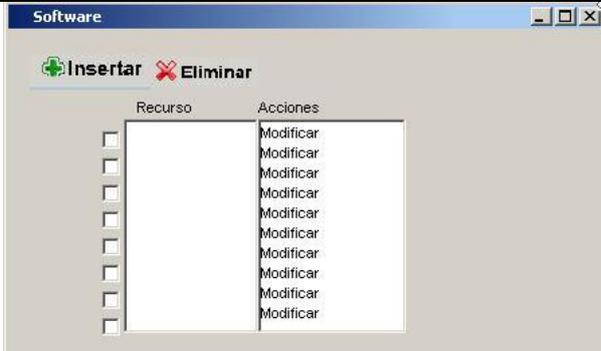
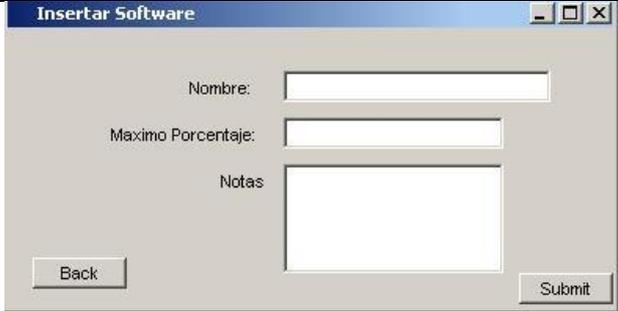
**Tabla 9 Gestionar Software**

<b>Nombre del CU</b>	Gestionar Software.	
<b>Actores</b>	Planificador del proyecto.(Inicia)	
<b>Propósito</b>	El caso de uso tiene como propósito la incorporación del software necesarios a utilizar en el proyecto.	
<b>Resumen</b>	<p>El caso de uso se inicia cuando el planificador del proyecto conoce la lista de software a utilizar por tareas donde se desarrollan las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar: El planificador del proyecto incorpora en el sistema el nuevo software a utilizar, finalizando así el caso de uso.</li> <li>• Modificar: El planificador del proyecto selecciona de la lista el software a modificar, mostrándose los datos del mismo para modificarlo, finalizando así el caso de uso.</li> <li>• Eliminar: El planificador del proyecto selecciona el nombre del software, o el software a eliminar donde puede eliminar el mismo, finalizando así el caso de uso.</li> </ul>	
<b>Referencias</b>	RF: 1, RF 2,RF 2.1,RF 3,RF 4	
<b>Precondiciones</b>	<p>El jefe de proyecto debe estar autenticado en el Dotproject.</p> <p>El jefe de proyecto debe estar con el rol de administrador en el sistema Dotproject.</p>	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El planificador de la facultad selecciona cuál acción desea realizar:</p> <p>a. Insertar.</p> <p>b. Modificar.</p> <p>c. Eliminar.</p>	<p>2. El sistema, en dependencia de la acción solicitada por el planificador del proyecto, muestra la interfaz correspondiente:</p> <p>a. Insertar: ir al modulo recursos y selecciona la pestaña Software y dentro de la misma la sección "Insertar".</p> <p>b. Modificar: ir al modulo recursos y selecciona la pestaña Software y dentro de la misma la sección "Modificar".</p> <p>c. Eliminar: ir al modulo recursos y selecciona la pestaña Software y dentro de la misma la sección "Eliminar".</p>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Sección “Insertar”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. El planificador del proyecto se autentica en el Dotproject	2. El sistema compraba el rol que desempeña en el Dotproject.
3. El planificador del proyecto selecciona el modulo de Recurso.	4. El sistema crea la interfaz para agregar software
5. El planificador del proyecto incorpora los datos necesarios para la introducción del software	6. El sistema verifica que ningún campo haya quedado vacío.
	7. El sistema verifica si el software existe
	8. El sistema registra el nuevo software, terminando así el caso de uso.
<b>Sección “Insertar”: Flujo Alternativo</b>	
	1. Si el usuario no está registrado en la base de datos del Dotproject no podrá incorporar nuevos software.
	2. Si el planificador del proyecto no tiene el rol indicado para utilizar este recurso no podrá agregar software.
	6. El sistema muestra un mensaje de error “No puede estar vacío”. 7. En el caso de que ya exista el software, el sistema muestra un mensaje “Ya existe el software”
<b>Prioridad</b>	<b>Critico</b>
<b>Sección “Modificar”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección “Software” y muestra sus datos.
2. El planificador selecciona la opción de modificar.	3. El sistema carga los datos a modificar
4. El planificador del proyecto introduce los datos nuevos del software.	5. El sistema verifica que el campo no haya quedado vacío. 6. El sistema muestra el dato existente para su modificación. 7. El sistema actualiza los datos.

CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Sección “Modificar”: Flujo Alternativo de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	4. En caso del campo estar vacio muestra un mensaje “No puede estar vacio” 5. El sistema muestra un mensaje ”Ya existe el software”
<b>Sección “Eliminar Software” : Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección “Software” y muestra sus datos.
2. El planificador del proyecto selecciona los software que desea eliminar.	3. El sistema elimina los Software.
<b>Sección “Eliminar Software”: Flujo Alternativo de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	3. En caso de que no exista el software mostrara un mensaje “El software no existe”.
 <p><b>Ilustración 9 Prototipo no funcional. Gestionar Software</b></p>	 <p><b>Ilustración 10 Prototipo no funcional. Insertar Software</b></p>
<b>Prioridad</b>	<b>alta</b>

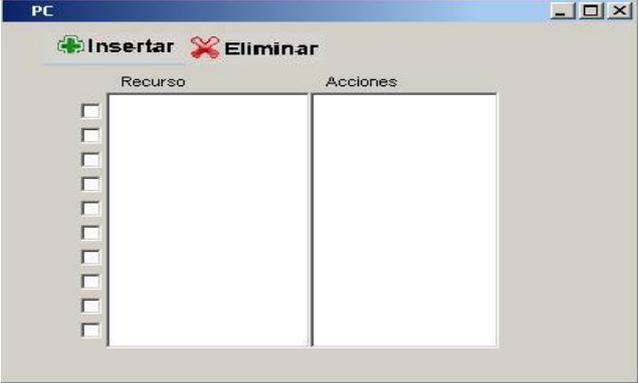
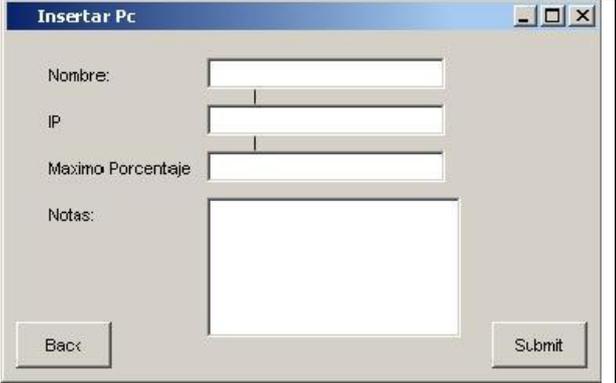
**Tabla 10 Gestionar PC**

<b>Nombre del CU</b>	Gestionar PC.	
<b>Actores</b>	Planificador del proyecto.(Inicia)	
<b>Propósito</b>	El caso de uso tiene como propósito la incorporación del PC a utilizar en el proyecto.	
<b>Resumen</b>	<p>El caso de uso se inicia cuando el planificador del proyecto conoce la lista de software a utilizar por tareas donde se desarrollan las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar: El planificador del proyecto incorpora un nombre a los IP, finalizando así el caso de uso.</li> <li>• Modificar: El planificador del proyecto selecciona de la lista el nombre de la PC que desea modificar, mostrándose la información, modificándose así los datos correspondientes, finalizando el caso de uso.</li> <li>• Eliminar: El planificador selecciona de la lista uno o mas nombres de PC que desea eliminar, eliminar el mismo, finalizando así el caso de uso.</li> </ul>	
<b>Referencias</b>	RF 14, 14.1, 12	
<b>Precondiciones</b>	<p>El jefe de proyecto debe estar autenticado en el Dotproject.</p> <p>El jefe de proyecto debe estar con el rol de administrador en el sistema Dotproject.</p>	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El planificador de la facultad selecciona cuál acción desea realizar:</p> <ul style="list-style-type: none"> <li>a. Insertar.</li> <li>b. Modificar.</li> <li>c. Eliminar.</li> </ul>	<p>2. El sistema, en dependencia de la acción solicitada por el planificador del proyecto, muestra la interfaz correspondiente:</p> <ul style="list-style-type: none"> <li>a. Insertar: ir al modulo recursos y selecciona la pestaña PC y dentro de la misma la sección "Insertar".</li> <li>b. Modificar: ir al modulo recursos y selecciona la pestaña PC y dentro de la misma la sección "Insertar".</li> <li>c. Eliminar: ir al modulo recursos y selecciona la pestaña PC y dentro de la misma la sección</li> </ul>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

		"Insertar".
<b>Sección "Insertar": Flujo Normal de Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
		1. El planificador del sistema selecciona la opción de insertar creándose la interfaz para agregar PC.
2. El planificador del proyecto introduce el número de IP y el nombre de la PC y los datos correspondiente a la misma.		3. El sistema verifica que no haya quedado ningún campo vacio. 4. El sistema verifica que haya introducido un número de IP valido.
		5. El sistema registra la nueva PC, terminando así el caso de uso.
<b>Sección "Insertar": Flujo Alternativo</b>		
		3. En caso de que quede algún campo vacio se mostrara un mensaje "Campo Vacio".
		4. En caso de que en número de IP no sea valido se mostrara un mensaje "numero de IP no valido".
		5. En caso de que el nombre de la PC sea duplicado el sistema emitirá un mensaje "Ya existe una PC con el mismo nombre".
<b>Prioridad</b>	<b>Critico</b>	
<b>Sección "Modificar": Flujo Normal de Eventos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
		1. El sistema ejecuta la sección "Modificar" y muestra sus datos.
2 El planificador del proyecto modifica los datos.		3 El sistema muestra el dato existente para su modificación. 4 El sistema verifica que no haya quedado ningún campo vacio. 5 El sistema verifica que haya introducido un número de IP valido.
<b>Sección "Modificar": Flujo Alternativo de Eventos</b>		

CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Acciones del Actor	Respuesta del Sistema
	<p>4. En caso de que en número de IP no sea valido se mostrara un mensaje “numero de IP no valido”.</p> <p>5. En caso de que el nombre de la PC sea duplicado el sistema emitirá un mensaje “Ya existe una PC con el mismo nombre”.</p>
<b>Sección “Eliminar” : Flujo Normal de Eventos</b>	
Acciones del Actor	Respuesta del Sistema
	<p>1. El sistema ejecuta la sección “Software” y muestra sus datos.</p>
<p>2. El planificador del proyecto selección el nombre de las PC que desea eliminar.</p>	<p>3. El sistema elimina las PC seleccionadas.</p>
<b>Sección “Eliminar”: Flujo Alternativo de Eventos</b>	
Acciones del Actor	Respuesta del Sistema
	<p>3. En caso de que no exista el nombre de la PC mostrara un mensaje “El nombre de la PC no existe”.</p>
<b>Interfaz no Funcional</b>	
	
<p><b>Ilustración 11</b> Prototipo no funcional. Gestionar PC</p>	<p><b>Ilustración 12</b> Prototipo no funcional. Insertar PC</p>
Prioridad	

**Tabla 11 Gestionar Horario**

<b>Nombre del CU</b>	Gestionar Horario.
<b>Actores</b>	Planificador del Proyecto (Inicia)
<b>Propósito</b>	El caso de uso tiene como propósito la distribución equitativa de las PC en un horario determinado.
<b>Resumen</b>	<p>El CU se inicia cuando el sistema tiene el listado de los miembros del proyecto así como el listado del horario libre por PC:</p> <ul style="list-style-type: none"> <li>• <u>Buscar Usuario:</u> En cual el planificador del proyecto introduce la Fecha de asignación del horario y selecciona de la lista de los miembros del proyecto existentes el Nombre del Usuario Ver caso de uso buscar usuario.</li> <li>• <u>Buscar PC asignada:</u> En el cual el planificador del proyecto selecciona de la lista de PC asignada el IP.</li> <li>• <u>Buscar Tareas:</u> El planificador del proyecto introduce la fecha de asignación del horario y la compara con la fecha de cumplimiento de la tarea y selecciona de la lista de tareas el código correspondiente a la misma obteniendo un listado.</li> <li>• <u>Eliminar horario:</u> El planificador del proyecto introduce la fecha de asignación del horario y selecciona de la lista de horarios asignados a tareas el usuario de uno de los miembros del proyecto, eliminando su horario en ese tiempo de aprovechamiento laboral.</li> <li>• <u>Modificar Horario:</u> El planificador del proyecto introduce la fecha de asignación del horario y selecciona de la lista de los miembros del proyecto el Nombre de Usuario de uno de los miembros del proyecto visualizando sus datos de asignación y modificándolos.</li> <li>• <u>Agregar Horario:</u> El planificador del proyecto introduce la fecha de asignación del horario y selecciona de la lista de los miembros del proyecto el nombre de usuario, selecciona la tarea correspondiente a dicha fecha agregando el horario correspondiente.</li> </ul>
<b>Referencias</b>	RF 11,11.1,15,15.1,16,16.1,17,18,18.1,19,20,20.1

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Precondiciones</b>	<p>Que el planificador debe estar autenticado en el Dotproject.</p> <p>Deben estar incorporadas las tareas al Dotproject.</p> <p>Deben tener las PC asignadas a los miembros del proyecto.</p>
<b>Curso Normal de los Eventos</b>	
<p><b>Acciones del Actor</b></p> <p>1.- El planificador del proyecto selecciona la acción a realizar:</p> <ul style="list-style-type: none"> <li>a. <u>Buscar Usuario</u></li> <li>b. <u>Buscar PC asignada</u></li> <li>c. <u>Buscar Tareas</u></li> <li>d. <u>Eliminar horario</u></li> <li>e. <u>Modificar Horario</u></li> </ul>	<p><b>Respuesta del Sistema</b></p> <p>2.- El sistema, en dependencia de la acción solicitada por el planificador del proyecto, muestra la interfaz correspondiente:</p> <ul style="list-style-type: none"> <li>a. <u>Buscar Usuario</u>: Ir a sección “Buscar usuario”</li> <li>b. <u>Buscar PC asignada</u>: Ir a sección “Buscar PC asignada”</li> <li>c. <u>Buscar Tareas</u> Ir a sección “Buscar Tareas”</li> <li>d. <u>Eliminar horario</u> Ir a sección “Eliminar Horario”</li> <li>e. <u>Modificar Horario</u> Ir a sección “Modificar Horario”</li> </ul>
<b>Sección “Buscar Usuario”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El planificador del proyecto crea una Interfaz para buscar los datos del sistema.
6 El planificador del proyecto introduce la fecha de planificación de horario.	7 El sistema muestra el listado de los miembros de los usuarios del sistema.
8 El planificador del proyecto selecciona el nombre de usuario del miembro del proyecto al cual será asignado un horario.	9 El sistema muestra el listado de usuario planificado según la selección, finalizando el caso de uso.
<b>Sección “Buscar PC asignada”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El planificador del proyecto crea una Interfaz para buscar las PC asignadas.
2. El planificador del proyecto selecciona el Nombre de Usuario de uno de los miembros	3. El sistema muestra una lista de las IP asignada al usuario seleccionado.

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

del proyecto.	
4. El planificador del proyecto selecciona el IP que corresponde a la asignación.	5. El sistema muestra una lista de los horarios asignados según el IP, finalizando así el caso de uso.
<b>Sección “Buscar PC asignada”: Flujo Alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. Si el Nombre de usuario seleccionado no tiene IP asignado se emitirá un mensaje de error “ <u>El Usuario no tiene PC asignada</u> ”, donde podrá seleccionar dos opciones una de asignar PC o cancelar la operación.
<b>Sección “Buscar Tareas”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El planificador del proyecto crea una Interfaz para buscar Tareas.
2. El planificador del proyecto introduce la fecha de planificación de horario.	3. El sistema compara la fecha de planificación del tiempo de trabajo con la fecha de planificación de las tareas asignadas en el proyecto.
	4. El sistema muestra una lista de las tareas a realizar en dicha fecha planificada.
5. El planificador del proyecto selecciona ID correspondiente a la tarea.	6. El sistema muestra una lista de los horarios asignados según el ID de la tarea, finalizando así el caso de uso.
<b>Sección “Buscar Tareas”: Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. En caso de dejar la fecha en blanco emitir un mensaje “Debe introducir una fecha para la planificación”. En caso de teclear la fecha en un formato diferente al indicado, se emitirá un mensaje de

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	error "Fecha incorrecta".
	3. Si no existe tareas asignada en la fecha introducida se emitirá un mensaje "No existe tarea para la fecha introducida", dando la opción de agregar tarea o cancelar.
<b>Sección "Eliminar horario": Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El planificador del proyecto crea una Interfaz para buscar eliminar horario.
2. El planificador del proyecto introduce la fecha de planificación de horario.	
3. El planificador del proyecto selecciona el nombre de usuario de uno de los miembros del proyecto.	4. El sistema muestra una lista de los usuarios que tienen asignado tarea e IP según la fecha escrita.
5. El planificador del proyecto selecciona el horario a eliminar, finalizando así el caso de uso.	
<b>Sección "Eliminar horario": Flujo alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	2. En caso de dejar la fecha en blanco emitir un mensaje "Debe introducir una fecha para la planificación" En caso de teclear la fecha en un formato diferente al indicado, se emitirá un mensaje de error "Fecha incorrecta".
	5. Si no existe horario planificado en esa fecha no podrá eliminarlo.
<b>Sección "Modificar horario": Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	1. El planificador del proyecto crea una Interfaz para buscar eliminar horario.
1. El planificador del proyecto introduce la fecha de planificación de horario.	
2. El planificador del proyecto selecciona el nombre de usuario de uno de los miembros del proyecto.	3. El sistema muestra una lista de los usuarios que tienen asignado tarea e IP según la fecha escrita.
4. El planificador del proyecto introduce el nuevo horario, finalizando así el caso de uso.	
Sección “Eliminar horario”: Flujo alternativo	
Acciones del Actor	Respuesta del Sistema
	5. En caso de dejar la fecha en blanco emitir un mensaje “Debe introducir una fecha para la planificación” En caso de teclear la fecha en un formato diferente al indicado, se emitirá un mensaje de error “Fecha incorrecta”.
	5. Si no existe horario planificado en esa fecha no podrá eliminarlo.

The screenshot shows a web form titled "Dia" with the following fields:

- Fecha:
- Usuario:
- PC:
- Tarea:
- Hora Inicio (00:00):
- Hora Fin(00:00):

**Ilustración 13** Prototipo no funcional. Insertar Horario por dia

The screenshot shows a web form titled "Semana" with the following fields and a button:

- Semana:
- Usuario:
- Submit:

**Ilustración 14** Prototipo no funcional. Insertar Horario de semana

**Tabla 12 Gestionar asignación de Software**

<b>Nombre del CU</b>	Gestionar asignación de Software.	
<b>Actores</b>	Planificador del proyecto.(Inicia)	
<b>Propósito</b>	El caso de uso tiene como propósito la asignación de un software a una tarea especifica.	
<b>Resumen</b>	<p>El caso de uso se inicia cuando el planificador del proyecto conoce la lista de software a utilizar por tareas donde se desarrollan las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• Insertar: El planificador del proyecto selecciona el proyecto y la tarea de dicho proyecto a la cual desea agregar el software, seleccionado dicho software, finalizando así el caso de uso.</li> <li>• Buscar: El planificador del proyecto selecciona el proyecto y las tareas correspondiente al mismo, buscando la los software, finalizando así el caso de uso.</li> <li>• Eliminar: El planificador del proyecto selecciona de la lista de software el cual desea eliminar, finalizando así el caso de uso.</li> <li>• Modificar: El planificador del proyecto selecciona de la lista la asignación de software que desea modificar, finalizando asi el caso de uso.</li> </ul>	
<b>Referencias</b>	RF 2,8,8.1,9,9.1,10,10.1	
<b>Precondiciones</b>	Debe estar las tareas del proyecto introducida en el Dotproject Debe estar la lista de los Software introducida en el Dotproject	
<b>Curso Normal de los Eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1. El planificador de la facultad selecciona cuál acción desea realizar: <ul style="list-style-type: none"> <li>a. Buscar</li> <li>b. Insertar</li> <li>c. Eliminar</li> <li>d. Modificar</li> </ul>	2. El sistema, en dependencia de la acción solicitada por el planificador del proyecto, muestra la interfaz correspondiente: <ul style="list-style-type: none"> <li>a. Buscar: Ir al modulo de recurso y seleccionar la pestaña de asignar software, donde debe seleccionar el proyecto y la tarea correspondiente al mismo, ejecuta la sección submit.</li> </ul>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	<p>b. Insertar: El usuario debe ir al modulo recursos y buscar el proyecto y la tarea en la pestaña de asignar software después selecciona selecciona el proyecto y la tarea.</p> <p>c. Eliminar: El usuario debe ir a modulo Recursos y seleccionar la lección “Eliminar”.</p>
<b>Sección “Agregar asignación de software”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El planificador del sistema crea la interfaz para Agregar asignación de software.
2. El planificador del sistema selecciona de la lista el nombre de la tarea asignar.	3. El sistema muestra la lista de los ID de las tarea.
4. El planificador del sistema selecciona de la lista el software asignar.	<p>5. El sistema muestra el nombre de los software asignar.</p> <p>6. El sistema verifica que haya seleccionado la tarea y el software.</p> <p>7. El sistema guarda la tarea y software asignados.</p>
<b>Sección “Agregar asignación de software”: Flujo Alternativo</b>	
	6. Si el usuario no selecciona la tarea se mostrara el siguiente mensaje “Seleccione una tarea”, si el usuario no selecciona un software se mostrara un mensaje “Seleccione un Software”, cancelando en ambos casos la asignación de software a tarea.
	7. Si existe en el sistema la tarea seleccionada y el software seleccionada se mostrara un mensaje “Tarea ya asignada a el software seleccionado, seleccione otro software”.
<b>Prioridad</b>	<b>Critico</b>
<b>Sección “Buscar asignación de software”: Flujo Normal de Eventos</b>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección "Buscar asignación de software".
2. El planificador del proyecto selecciona la tarea a buscar.	3. El sistema verifica el software asignado a dicha tarea y muestra los datos existentes.
<b>Sección "Buscar asignación de software" : Flujo Alternativo de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	3. En el caso de que no exista software asignado a dicha tarea se mostrara un mensaje "No hay software asignado".
<b>Sección "Eliminar asignación de software": Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección "Eliminar asignación de software" y muestra sus datos..
2. El planificador del sistema selecciona de la lista el nombre de la tarea a eliminar	3. El sistema muestra la lista de los ID de las tarea.
4. El planificador del sistema selecciona de la lista el software a eliminar.	5. El sistema muestra el nombre del software a eliminar. 6. El sistema verifica que haya seleccionado la tarea y el software. 7. El sistema elimina la tarea y software asignados.
<b>Sección "Eliminar asignación de software": Flujo Alternativo de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	6. Si el usuario no selecciona la tarea se mostrara el siguiente mensaje "Seleccione una tarea", si el usuario no selecciona un software se mostrara un mensaje "Seleccione un Software", cancelando en ambos casos la asignación de software a tarea.
	7. Si existe en el sistema la tarea seleccionada y el software seleccionada se elimina la tarea asignada.

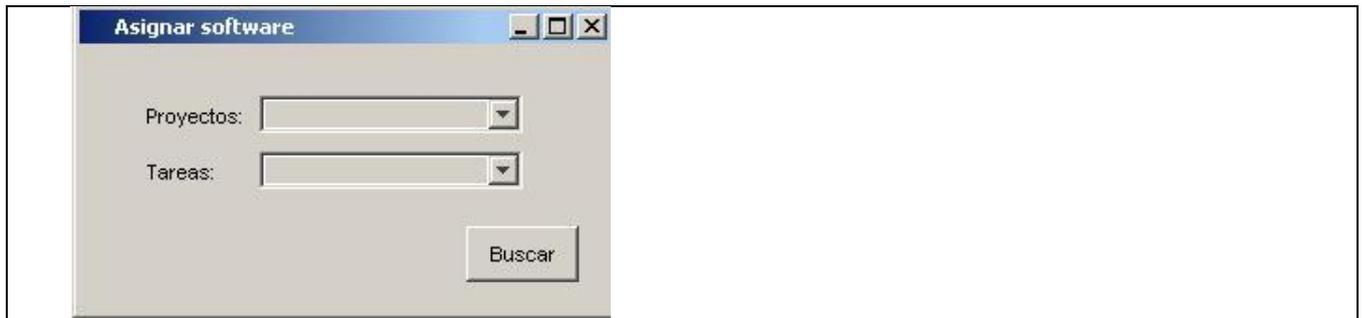


Ilustración 15 Prototipo no funcional. Gestionar asignación de Software

<b>Prioridad</b>	
------------------	--

Tabla 13 Gestionar asignación de PC

<b>Nombre del CU</b>	Gestionar asignación de PC.
<b>Actores</b>	Planificador del proyecto.(Inicia)
<b>Propósito</b>	El caso de uso tiene como propósito la asignación de una o varias PC a miembros del proyecto.
<b>Resumen</b>	<p>El caso de uso se inicia cuando el planificador del proyecto conoce la lista de los miembros del proyecto y la lista de las PC asignadas para la utilización del proyecto donde se desarrollan las siguientes operaciones:</p> <ul style="list-style-type: none"> <li>• <b>Agregar asignación de PC:</b> El planificador del proyecto selecciona de la lista de usuario el miembro del proyecto a asignar PC, ver caso de uso Gestionar horario, y después selecciona de la lista el nombre de la PC asignar, ver caso de uso Gestionar PC, finalizando así el caso de uso.</li> <li>• <b>Buscar asignación de PC:</b> El planificador del proyecto selecciona de la lista de usuario el miembro del proyecto a buscar asignación de PC, ver caso de uso Gestionar horario, mostrando la lista de los nombres de las PC asignada, finalizando así el caso de uso.</li> <li>• <b>Eliminar asignación de PC:</b> El planificador del proyecto selecciona de la lista de usuario el miembro del proyecto a eliminar la asignación de la PC, ver caso de uso Gestionar horario, y después se muestra la lista de las PC que ya están asignadas al miembro del proyecto, ver</li> </ul>

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	Buscar asignación de PC
<b>Referencias</b>	RF 11,11.1,12,12.1,13,13.1
<b>Precondiciones</b>	Debe estar los miembros del proyecto introducido en el Dotproject.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
<p>1. El planificador de la facultad selecciona cuál acción desea realizar:</p> <ul style="list-style-type: none"> <li>a. Agregar asignación de PC.</li> <li>b. Buscar asignación de PC.</li> <li>c. Eliminar asignación de PC.</li> </ul>	<p>2. El sistema, en dependencia de la acción solicitada por el planificador del proyecto, muestra la interfaz correspondiente:</p> <ul style="list-style-type: none"> <li>a. Agregar asignación de PC: ir al modulo recursos y selecciona la sección "Agregar asignación de PC".</li> <li>b. Buscar asignación de PC: Ir al modulo Recursos y selecciona la sección "Buscar asignación de PC"</li> <li>c. Eliminar asignación de PC: ir a modulo Recursos y seleccionar la lección "Eliminar asignación de PC".</li> </ul>
<b>Sección "Agregar asignación de PC": Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	<p>1. El planificador del sistema crea la interfaz para Agregar asignación de software.</p>
<p>2. El planificador del sistema selecciona de la lista el nombre del miembro del proyecto.</p>	<p>3. El sistema muestra la lista de los ID de los miembros del proyecto.</p>
<p>4. El planificador del sistema selecciona de la lista el nombre de la PC asignar.</p>	<p>5. El sistema verifica que haya seleccionado la el nombre el miembro y la PC.</p> <p>6. El sistema guarda el ID de las tareas y el IP de las PC.</p>
<b>Sección "Agregar asignación de PC": Flujo Alternativo</b>	
	<p>5. Si el usuario no selecciona la tarea se mostrara el nombre del miembro del proyecto se muestra el mensaje "Seleccione miembro del proyecto", si el</p>

CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

	usuario no selecciona un IP se mostrara un mensaje “Seleccione una PC”, cancelando en ambos casos la asignación de PC.
	6. Si existe en el sistema el miembro de proyecto y la pc seleccionada se mostrara un mensaje “Ya existe el miembro del proyecto y la PC asígnele otra PC”. Cancelando la asignación de la PC.
<b>Prioridad</b>	<b>Critico</b>
<b>Sección “Buscar asignación de PC”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección “Buscar asignación de PC”.
2. El planificador del proyecto selecciona el miembro del proyecto.	3. El sistema verifica los IP asignados a dicho miembro del proyecto y muestra los IP asignado.
<b>Sección “Buscar asignación de PC” : Flujo Alternativo de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	3. En el caso de que no exista ningun IP asignado al usuario seleccionado se mostrara un mensaje “No hay PC asignada al miembro del proyecto seleccionado”.
<b>Sección “Eliminar asignación de PC”: Flujo Normal de Eventos</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema ejecuta la sección “Eliminar asignación de PC” y muestra sus datos..
2. El planificador del sistema selecciona de la lista el nombre del miembro del proyecto a eliminar la asignación de PC	3. El sistema muestra la lista de los IP asignada a dicho miembro del proyecto.
4. El planificador del sistema selecciona de la lista la PC a eliminar.	5. El sistema muestra los datos a eliminar. 6. El sistema elimina el IP asignado al miembro del proyecto.
<b>Sección “Eliminar asignación de software”: Flujo Alternativo de Eventos</b>	

## CAPITULO 2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Acciones del Actor	Respuesta del Sistema
	5. Si el usuario seleccionado no tiene IP asignado se mostrara el siguiente mensaje “No existe PC asignada”
	
<b>Ilustración 16 Prototipo no funcional. Gestionar asignación de PC</b>	
<b>Prioridad</b>	<b>Alta</b>

**Tabla 14 Matriz de trazabilidad entre requerimientos y caso de uso del sistema**

Requerimientos	Gestionar Software	Gestionar PC	Buscar Software	Gestionar Horario	Buscar usuario	Gestionar asignación software	Gestionar asignación PC	Buscar recursos
RF 1	x							
RF 1.1	x							
RF 2	x		x			x		
RF 2.1	x		x					
RF 3	x							
RF 4	x							
RF 5		x						
RF 5.1		x						
RF 6		x						
RF 7		x						
RF 8						x		
RF 8.1						x		
RF 9						x		
RF 9.1						x		
RF 10						x		
RF 10.1						x		
RF 11				x	x		x	
RF 11.1				x	x		x	
RF 12							x	
RF 12.1							x	
RF 13							x	
RF 13.1							x	
RF 14								
RF 14.1								
RF 15				x				
RF 15.1				x				
RF 16				x				
RF 16.1				x				
RF 17				x				
RF 18				x				
RF 18.1				x				
RF 19				x				
RF 20				x				
RF 20.1				x				
RF 21								x

### Conclusiones

En el presente capítulo se describieron detalladamente los procesos de gestión desde la perspectiva del negocio, así como las personas que intervienen en dichos procesos. Se definieron las actividades que realizan estas personas así como las que deben ser informatizadas, utilizando los Diagramas de Actividades. Se definieron los requisitos que debe cumplir el sistema para su correcto funcionamiento. Determinado así las diferentes funcionalidades del mismo resumidas como casos de uso, de los que se realizó una descripción detallada para su mejor comprensión.



## Capítulo 3 Diseño del sistema

### Diseño

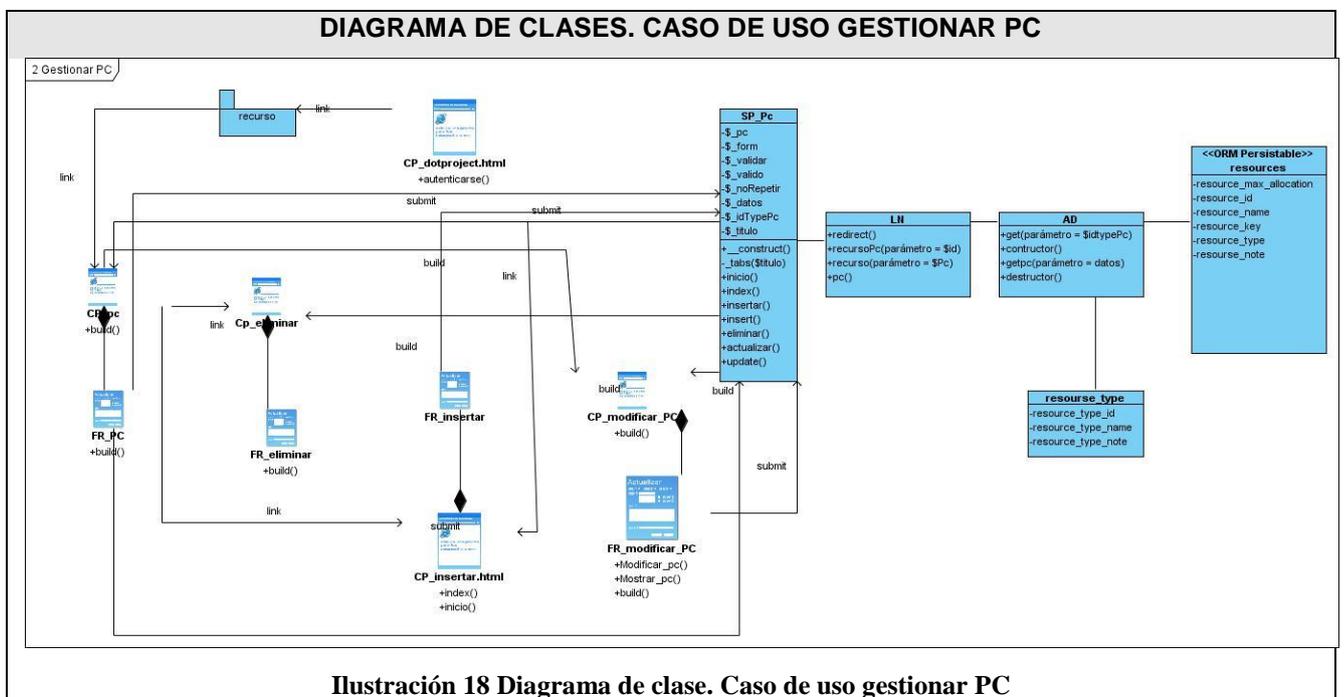
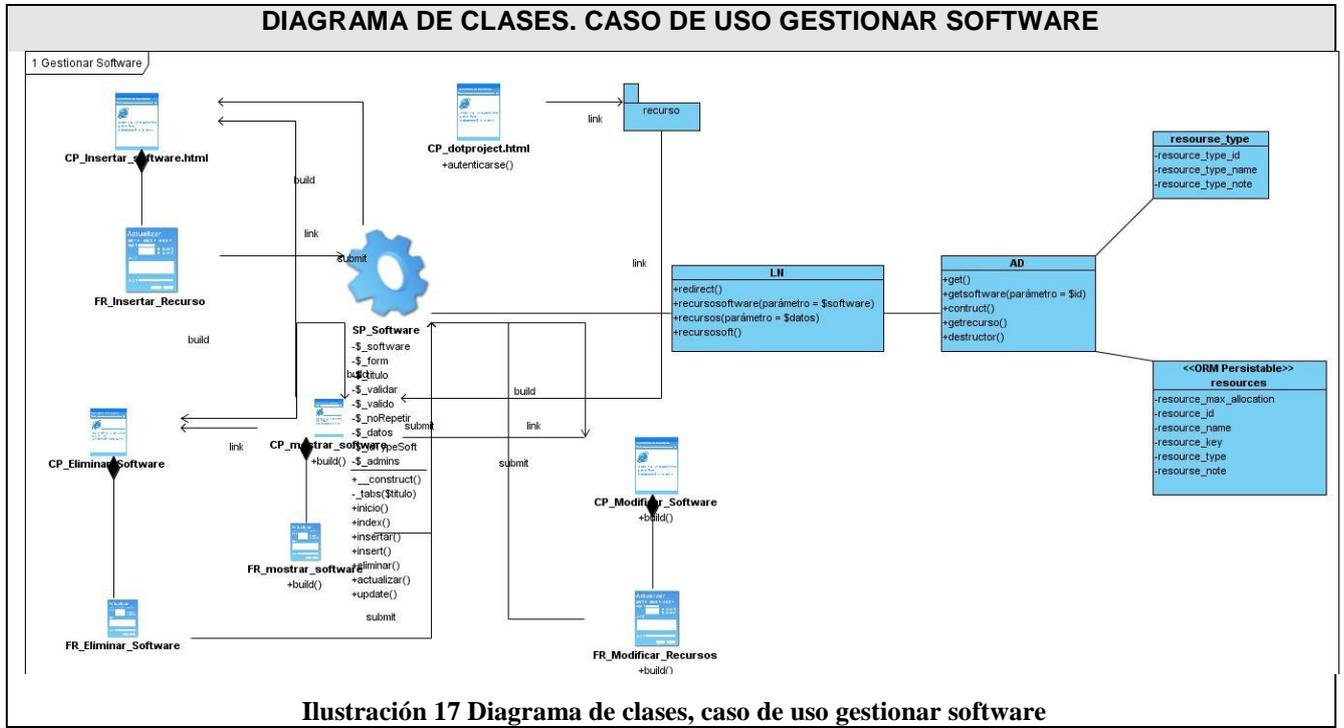
#### Introducción.

En el siguiente capítulo se muestra los modelos visuales del sistema, transformando los requisitos a diseños de sistema, se adapta el diseño para hacerlo correspondiente con el ambiente de implementación para así lograr mejores resultados, se hace un análisis de la base de datos del DotProject, haciendo la ingeniería inversa de la misma, generando el diagrama de clases persistentes.

#### 3.1 Diseño

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.

3.2 Diagrama de clase del diseño



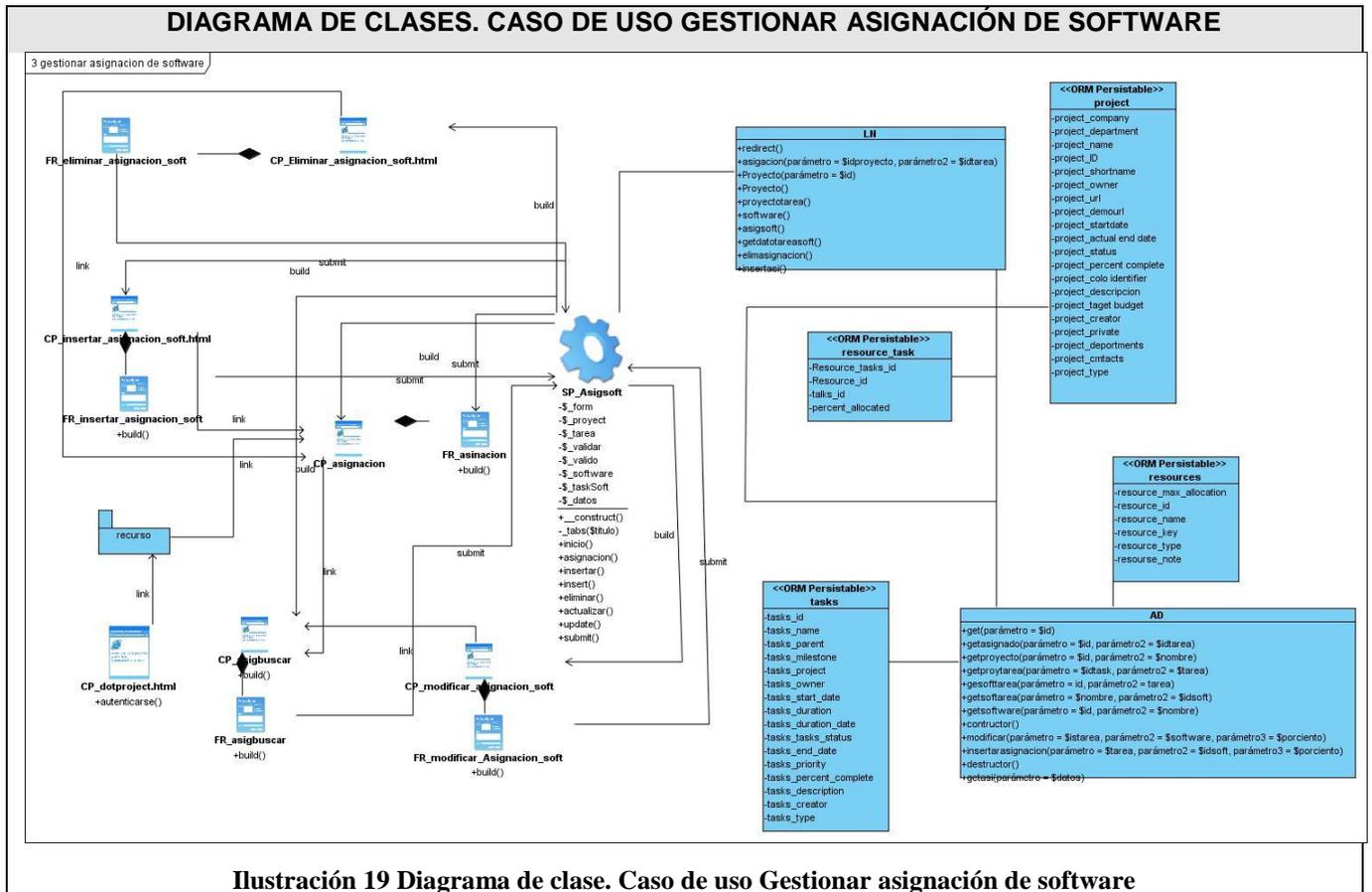
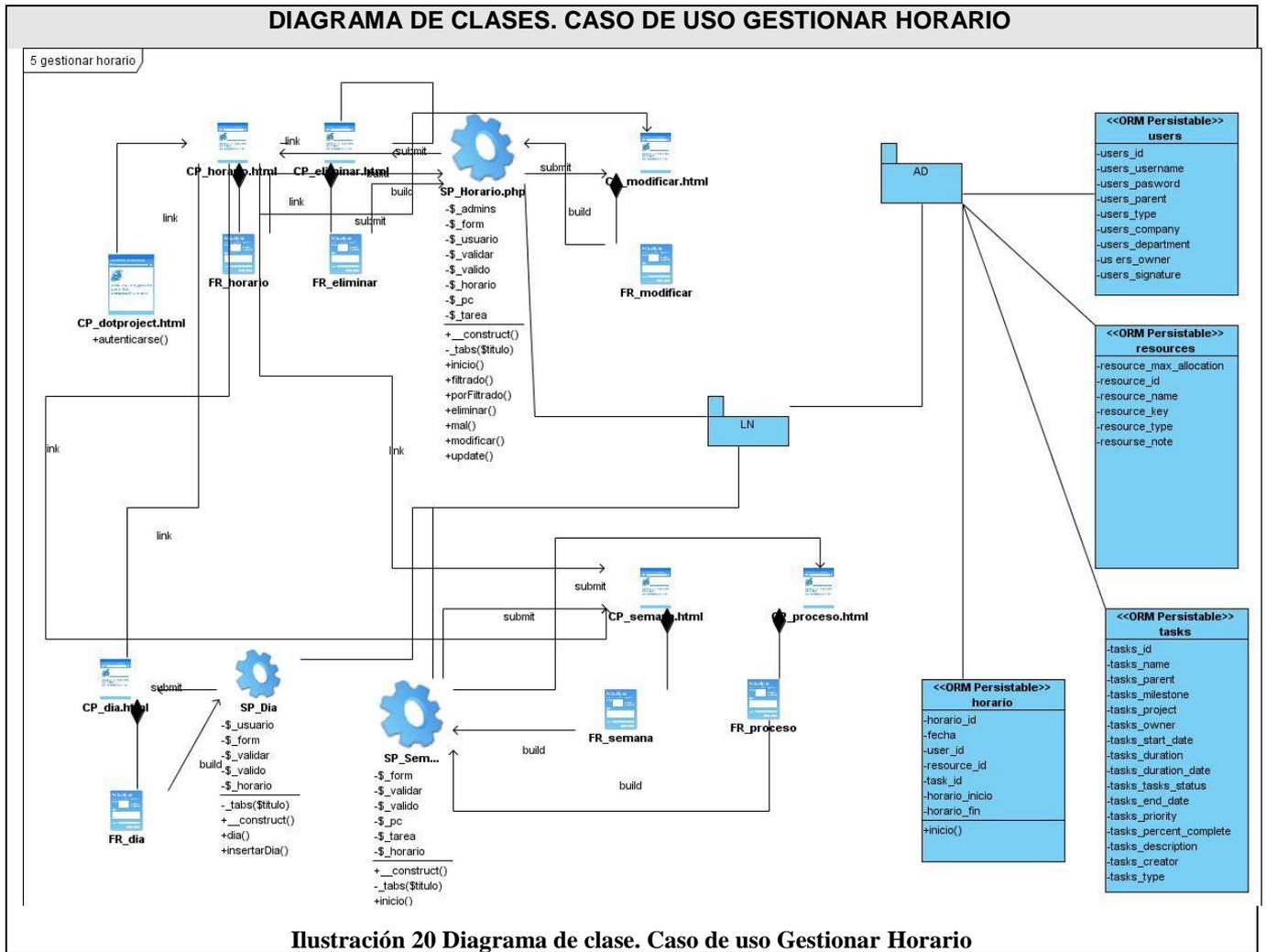
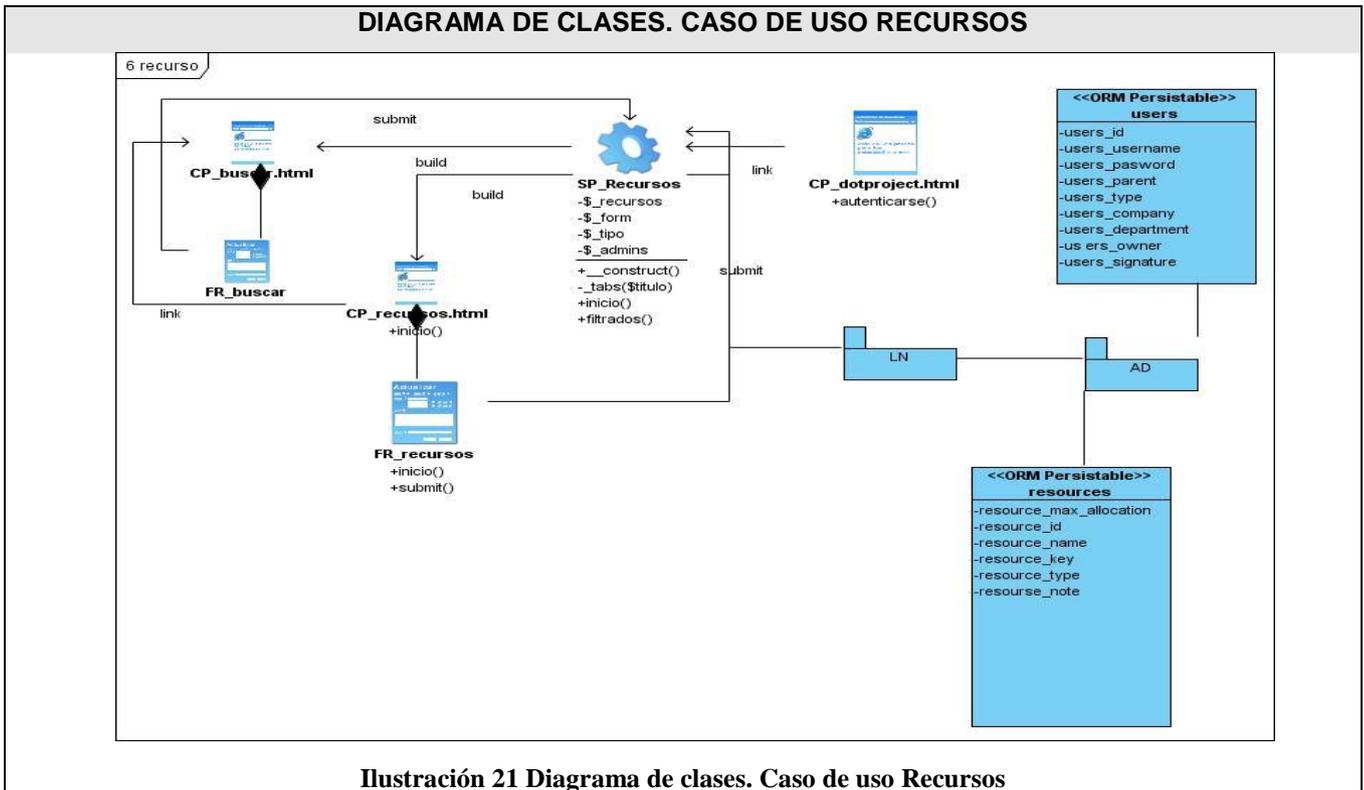


Ilustración 19 Diagrama de clase. Caso de uso Gestionar asignación de software



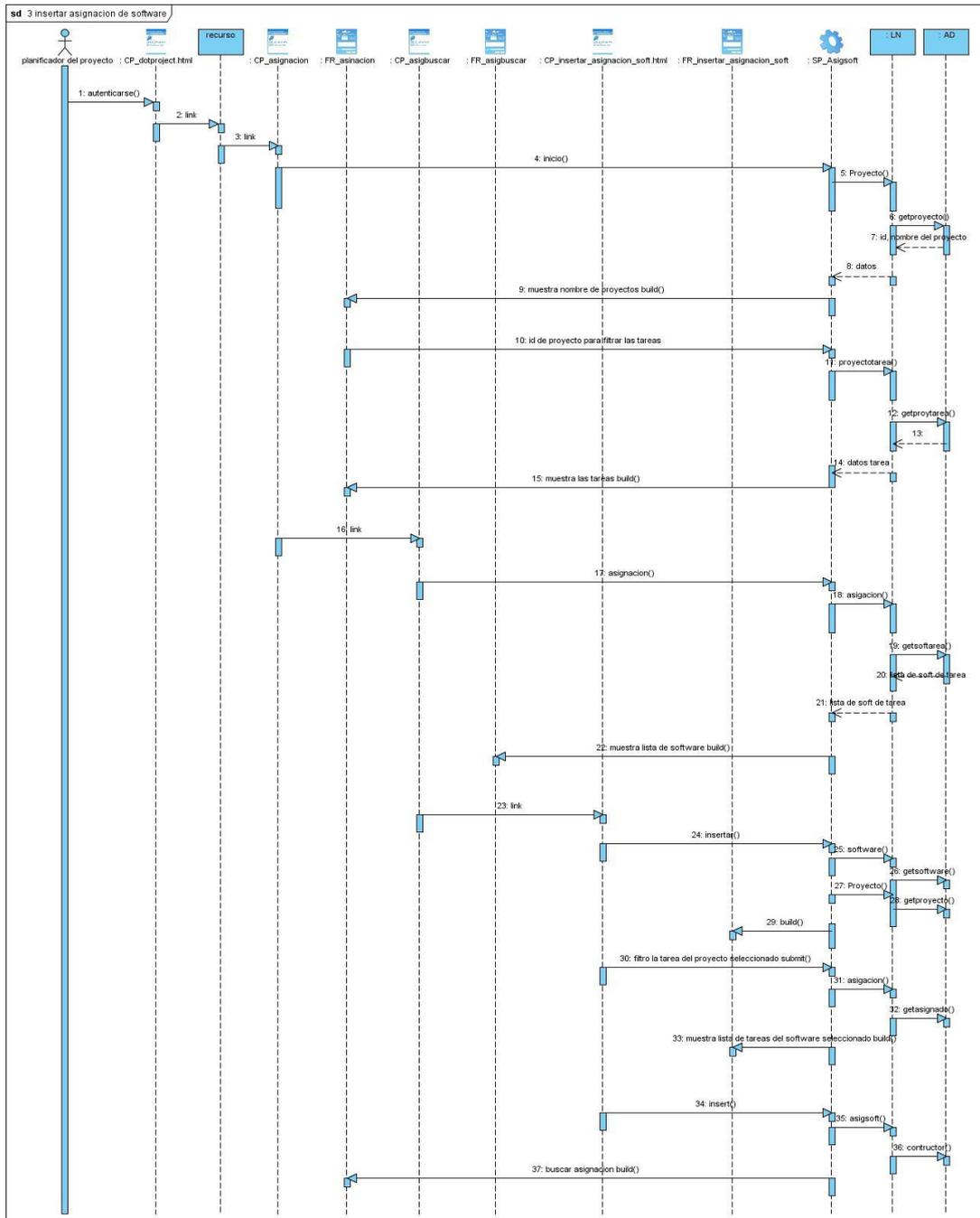
**Ilustración 20 Diagrama de clase. Caso de uso Gestionar Horario**



### 3.3 Diagrama de secuencia

El siguiente diagrama de secuencia corresponde al caso de uso de gestionar asignación de software, cuando el usuario necesita asignar un software, la página cliente, carga automáticamente de la BD la información comprobando el rol que desempeña siendo administrado o usuario. El primer mensaje sería el correspondiente a la selección de los proyectos ya existente, luego dado el identificador del proyecto seleccionado, se muestra las tareas asignadas a ese proyecto, este proceso se realiza a través de la clase controladora.

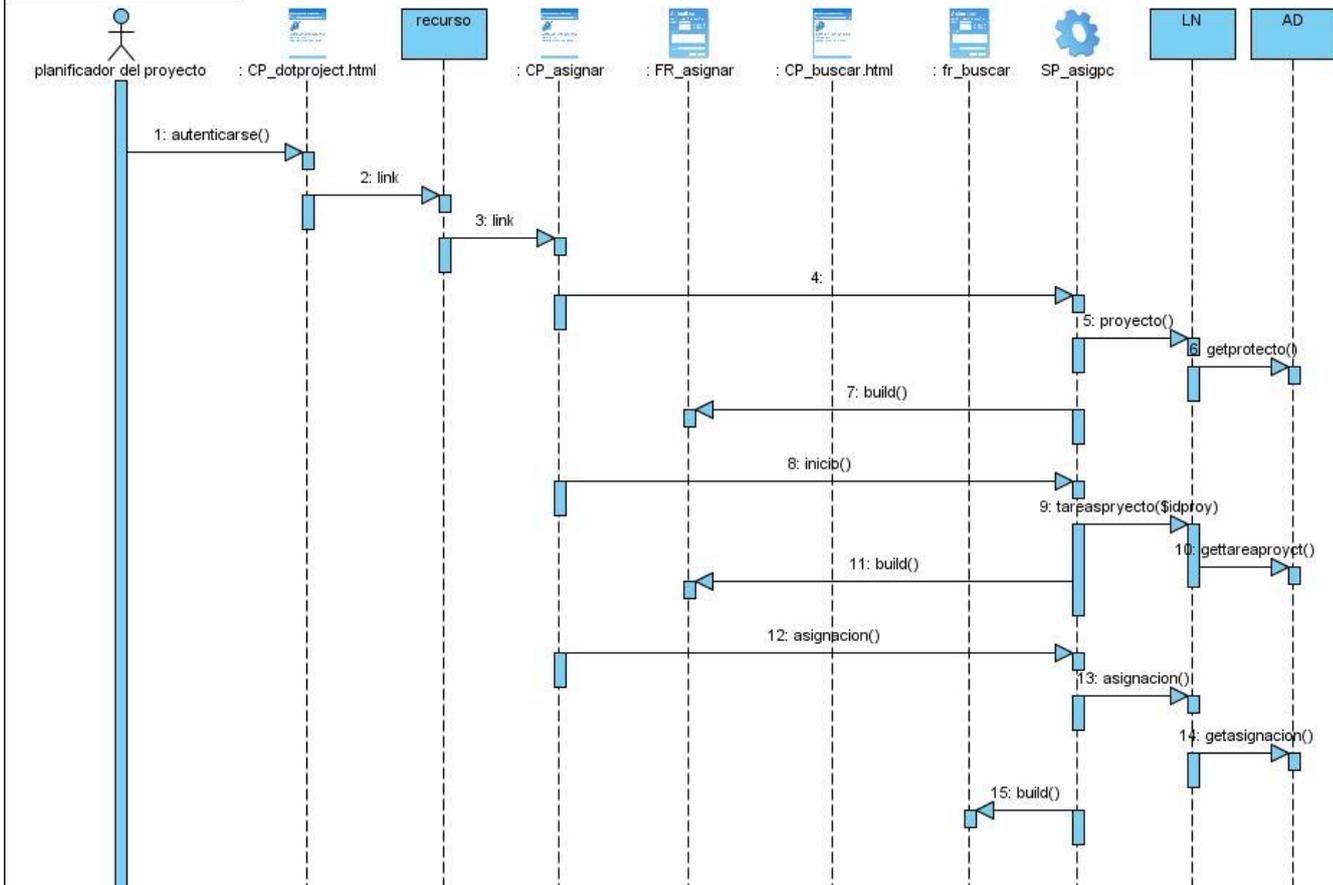
**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR ASIGNACIÓN DE SOFTWARE  
INSERTAR ASIGNACIÓN DE SOFTWARE**



**Ilustración 22 Diagrama de secuencia. Insertar asignación de software**

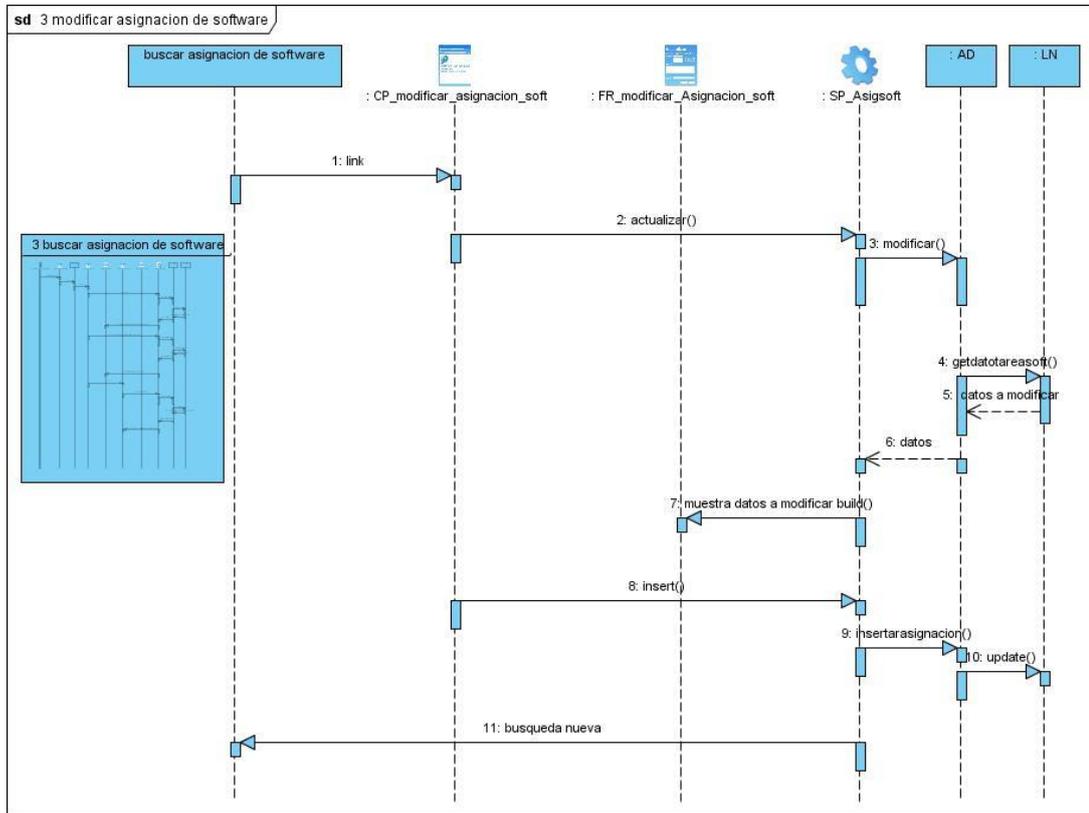
**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR ASIGNACIÓN DE SOFTWARE  
BUSCAR ASIGNACIÓN DE SOFTWARE**

sd 4 buscar asignacion PC



**Ilustración 23 Diagrama de secuencia. Buscar asignación de PC**

**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR ASIGNACIÓN DE SOFTWARE  
MODIFICAR ASIGNACIÓN DE SOFTWARE**



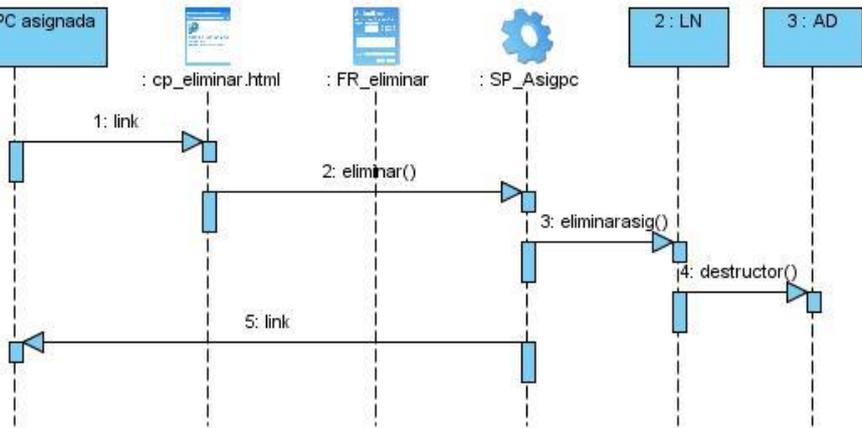
**Ilustración 24 Diagrama de secuencia. Modificar asignación de software**

**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR ASIGNACIÓN DE SOFTWARE  
ELIMINAR ASIGNACIÓN DE SOFTWARE**

**sd 4** eliminar asignacion de PC



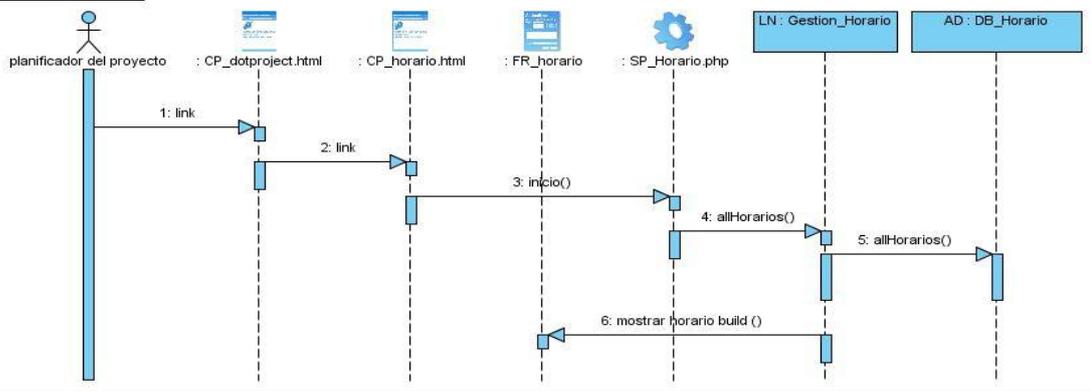
buscar PC asignada



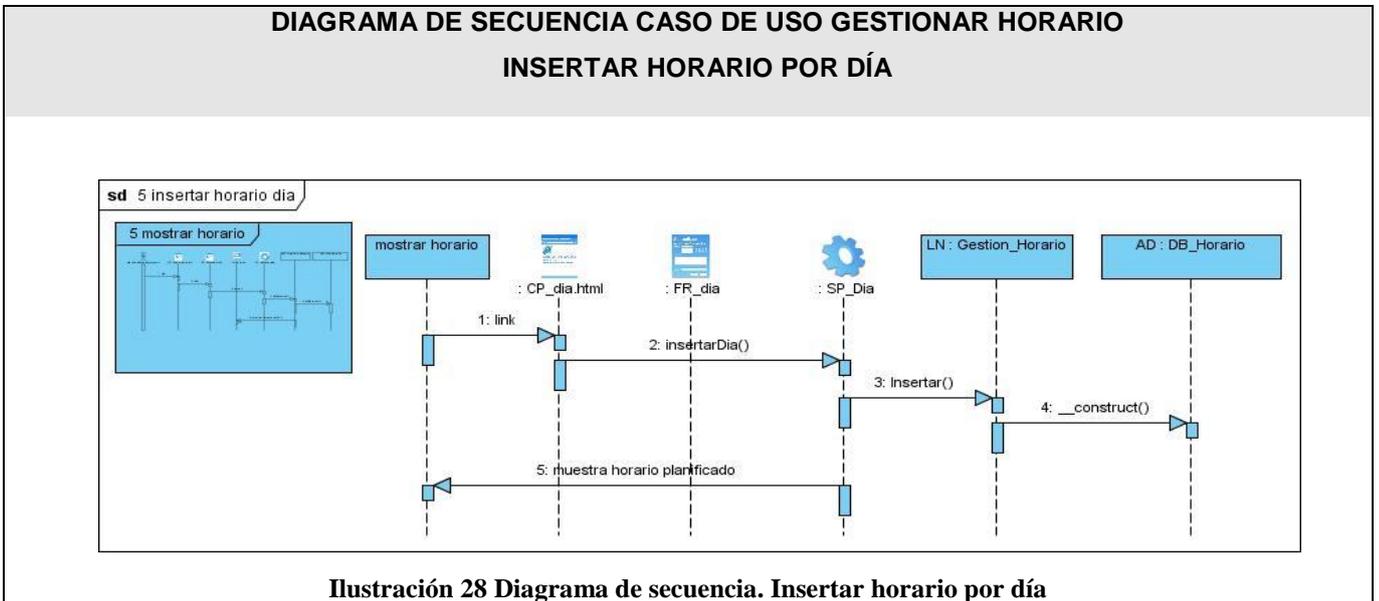
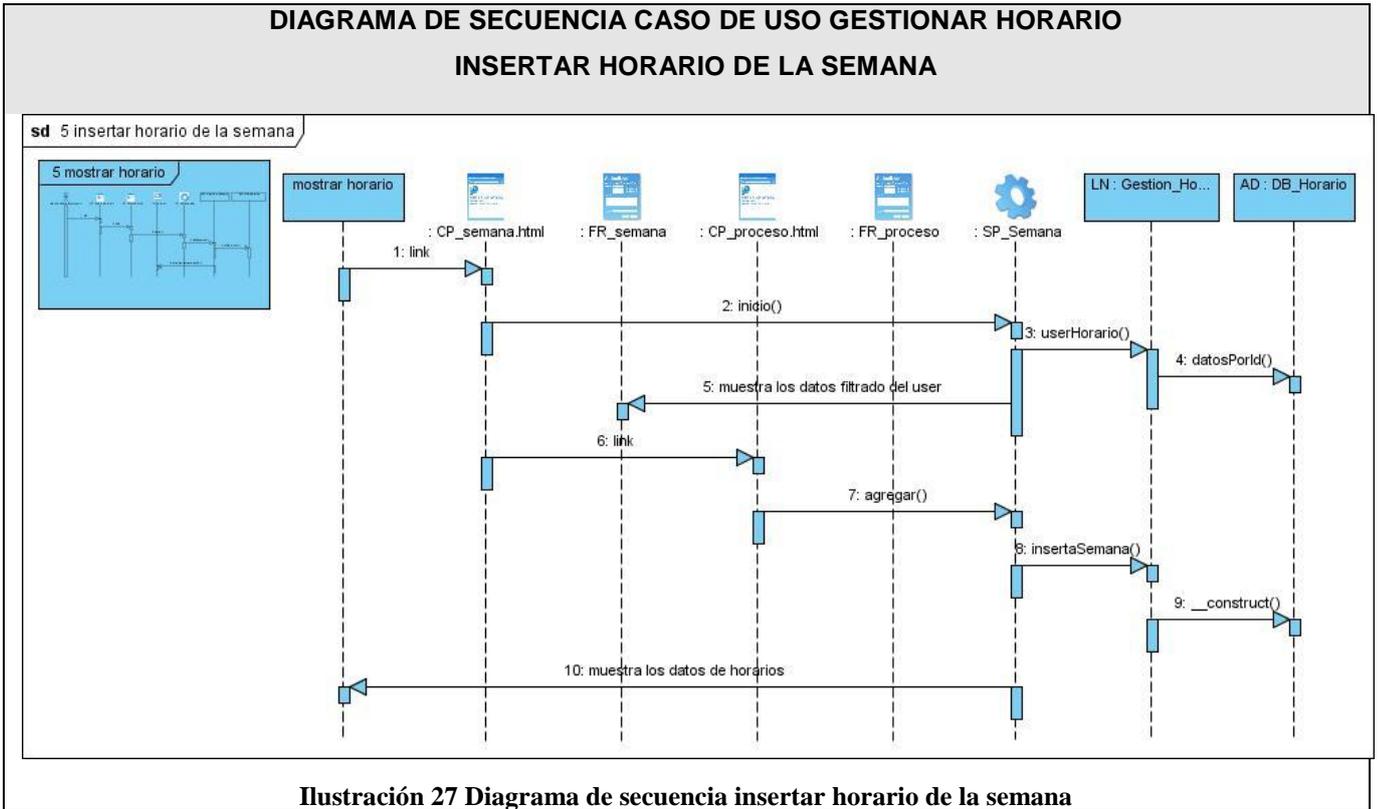
**Ilustración 25 Diagrama de secuencia. Eliminar asignación de software**

**DIAGRAMA DE INTERACCION CASO DE USO GESTIONAR HORARIO  
BUSCAR HORARIO**

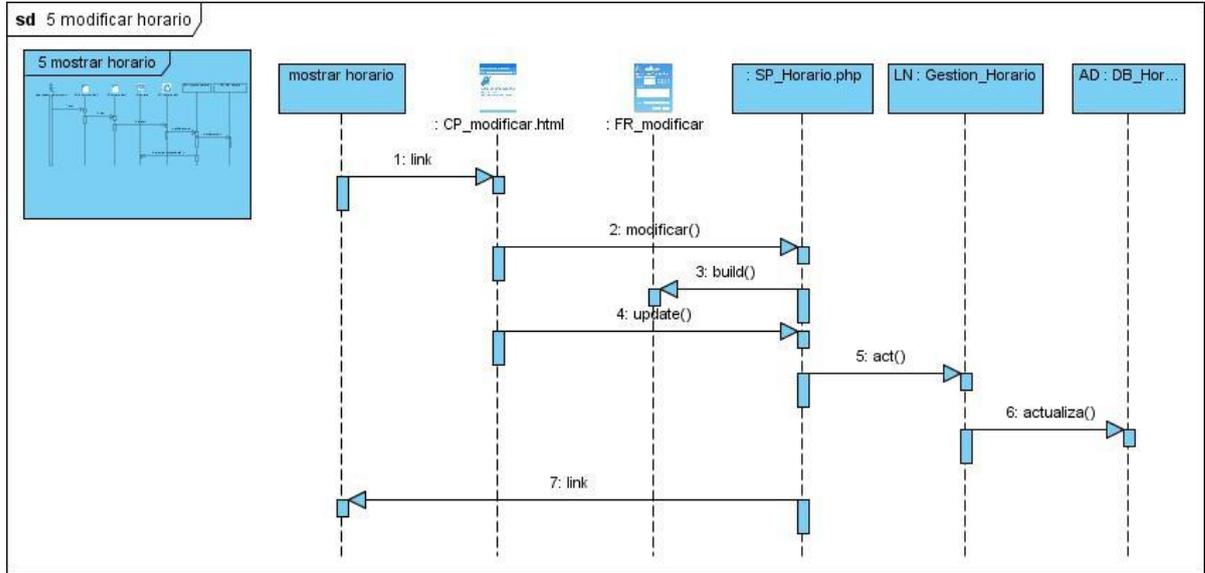
**sd 5** mostrar horario



**Ilustración 26 Diagrama de clases. Buscar Horario**

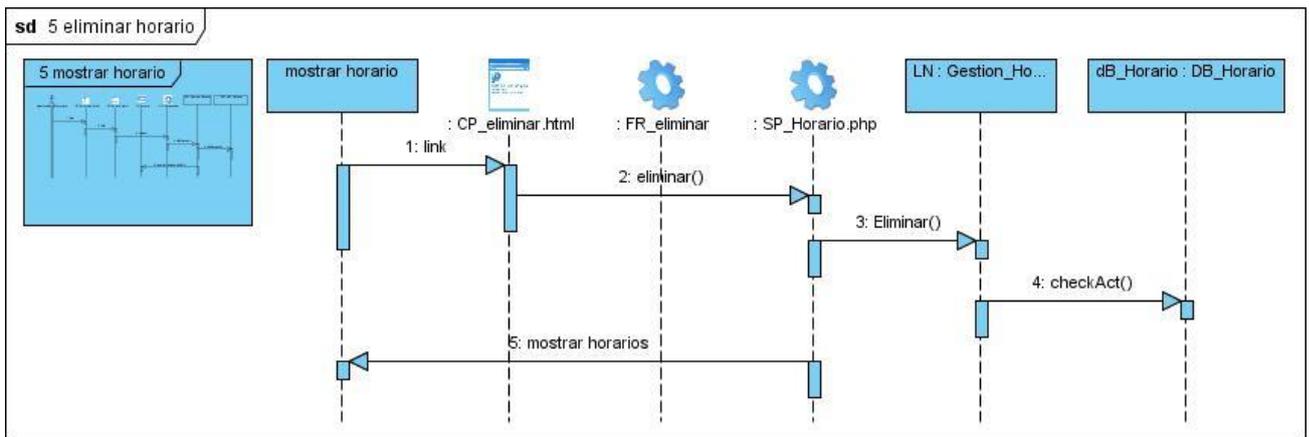


**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR HORARIO  
MODIFICAR HORARIO**



**Ilustración 21 Diagrama de secuencia. Modificar horario**

**DIAGRAMA DE SECUENCIA CASO DE USO GESTIONAR HORARIO  
ELIMINAR HORARIO**



**Ilustración 292 Diagrama de secuencia. Eliminar horario**

### 3.4 Descripción de las clases

A continuación se muestra la descripción de algunas de las principales clases.

#### Clases Asignar Software

Tabla 15 descripción de clases. Gestionar Software

	Nombre	Documentación
<b>Asigsoft</b> -_form -_proyect -_tarea -_validar -_valido -_software -_taskSoft -_datos +__construct() -_tabs() +inicio() +asignacion() +insertar() +insert() +eliminar() +actualizar() +update()	 Asigsoft	Clase para asignar los software
	<b>Atributos</b>	
	private _form	objeto de tipo formulario
	private _proyect	objeto de tipo tabla proyecto
	private _tarea	objeto de tipo tabla Task
	private _validar	objeto de tipo validar
	private _valido	resultado de la validación
	private _software	objeto de tipo tabla resources en especifico software
	private _taskSoft	objeto de tipo tabla resources_task
	private _datos	Datos para el procesamiento
	<b>Operaciones</b>	
	public __construct()	método para la seguridad del modulo
	private _tabs(titulo)	método que muestra las pestañas
	public inicio()	método que muestra el formulario para buscar los datos de proyectos y tareas en dependencia del proyecto
	public asignacion()	método que muestra el resultado de la búsqueda
	public insertar()	método para mostrar el formulario de insertar
	public insert()	método que valida los datos e inserta en la Base de datos
	public eliminar()	método para eliminar los datos
public actualizar()	método que muestra el formulario para actualizar los datos	
public update()	método que verifica los datos y actualiza en la Base de datos	

**Clase Gestionar PC**

Tabla 16 Descripción de clase. Gestionar PC

	Nombre	Documentación
<b>Asigpc</b> -\$_form -\$_usuario -\$_pc -\$_resource -\$_validar -\$_valido -\$_resourceUser +__construct() -_tabs() +inicio() +asignado() +eliminar() +insertar() +insert() +actualizar() +update()	 Asigpc	Clase para la asignación de PC
	<b>Atributos</b>	
	private _form	Objeto de tipo formulario
	private \$_usuario	Objeto de tipo table users
	private \$_pc	Objeto de tipo table resource específicamente las PC
	private _validar	Objeto de tipo validar
	private _valido	resultado de la validación
	private \$_resource	Objeto de tipo recurso
	private \$_resourceUser	Objeto de tipo tabla resource_user
	<b>Operaciones</b>	
	public __construct()	Método para la seguridad del modulo
	private _tabs(titulo)	Método que muestra las pestañas
	public inicio()	Método que muestra el formulario para buscar el usuario y la PC.
	public asignado()	Método que muestra el resultado de la búsqueda en dependencia si el usuario es administrador o no método para mostrar el formulario de insertar
public insertar()	Método que valida los datos e inserta en la base de datos	
public insert()	Método que valida los datos e inserta en la base de datos	
public eliminar()	Método para eliminar los datos	
public actualizar()	Método que muestra el formulario para actualizar los datos	
public update()	Método que verifica los datos y actualiza en la base de datos	

**Clase Gestionar PC**

Tabla 17 Descripción de clase. Gestionar PC

	Nombre	Documentación
<b>Pc</b> -\$_pc -\$_form -\$_validar -\$_valido -\$_noRepetir -\$_datos -\$_idTypePc -\$_titulo +__construct() -_tabs() +inicio() +index() +insertar() +insert() +eliminar() +actualizar() +update()	 Pc	clase para la gestión de los recursos de tipo PC
	<b>Atributos</b>	
	\$_pc	objeto de tipo tabla resource en especifico PC
	\$_form	objeto de tipo formulario
	\$_validar	objeto de tipo validación
	\$_valido	resultado de la validación
	\$_noRepetir	datos que no se deben repetir
	\$_datos	datos para ser procesados
	\$_idTypePc	identificador del resorce tipo PC
	\$_titulo	título del modulo
	<b>Operaciones</b>	
	function __construct()	método para la seguridad del modulo
	private function _tabs(\$titulo)	método para mostrar las pestañas
	function inicio()	método que muestra todos los recursos de tipo PC para los administradores
	function index()	método que muestra todos los recursos de tipo pc para los usuarios
	function insertar()	método con el formulario de insertar datos
	function insert()	método que verifica los datos y los inserta en la BD
function eliminar()	método para eliminar de la BD	
function actualizar()	método con el formulario de actualizar los datos	
function update()	método que verifica y actualiza los datos en la BD	

**Clase Buscar Recursos**

Tabla 18 Descripción de clase. Buscar recursos

	Nombre	Documentación
<div style="border: 1px solid black; padding: 5px;"> <p><b>Recursos</b></p> <p>-\$_recursos</p> <p>-\$_form</p> <p>-\$_tipo</p> <p>-\$_admins</p> <hr/> <p>+_construct()</p> <p>+_tabs()</p> <p>+inicio()</p> <p>+filtrados()</p> </div>	 Recursos	clase para mostrar los recursos y su tipo
	<b>Atributos</b>	
	private \$_recursos	objeto de tipo tabla resources
	private \$_form	objeto de tipo formulario
	private \$_tipo	objeto de tipo resource_type
	private \$_admins	objeto de tipo tabla users
	<b>Operaciones</b>	
	function __construct()	método que crea la sección de administrador
	private function _tabs(\$titulo)	método que muestra las pestañas
	function inicio()	método que muestra todos los recursos y su tipo
	function filtrados()	método que muestra los recursos filtrados por el tipo

**Clase Gestionar Software**

Tabla 19 Descripción de clase. Gestionar Software

	Nombre	Documentación
<b>Software</b> -\$_software -\$_form -\$_titulo -\$_validar -\$_valido -\$_noRepetir -\$_datos -\$_idTypeSoft -\$_admins +__construct() -_tabs() +inicio() +index() +insertar() +insert() +eliminar() +actualizar() +update()	 Pc	clase para la gestión de los recursos de tipo PC
	<b>Atributos</b>	
	private \$_software	objeto de tipo resources en especifico los software
	private \$_form	objeto de tipo formulario
	private \$_titulo	titulo del modulo
	private \$_validar	objeto de tipo validar
	private \$_valido	resultado de la validación
	private \$_noRepetir	datos que no se deben de repetir
	private \$_datos	datos para el procesamiento
	private \$_idTypeSoft	identificador de tipo software
	private \$_admins	objeto de tipo tabla users
	<b>Operaciones</b>	
	Function __construct()	método para la seguridad del modulo
	private function _tabs(\$titulo)	método para mostrar las pestañas
	function inicio()	método que me muestra todos los recursos de tipo software para los administradores
	function index()	método que me muestra todos los recursos de tipo software para los usuarios
	function insertar()	método que muestra el formulario de insertar
	function insert()	método que verifica los datos y los inserta en la bd
	function eliminar()	método para eliminar los datos de la bd
	function actualizar()	método que muestra el formulario para actualizar los datos
function update()	método que verifica los datos y los actualiza en la bd	

### 3.5 Base de datos

Para la realización del trabajo se realizo un estudio sobre la base de dato del Dotproject, pues esta cuenta para si funcionamiento con 72 tablas básicas, a las cuales se le añade las entidades correspondientes a los módulos agregados, lo cual garantiza el soporte de información necesaria para la la gestión de horarios y recursos.

Se hizo uso de las siguientes entidades:

**Tabla 20 Descripción entidad Resource\_Type**

<b>Resource-Type</b> Esta entidad almacena los tipos de recursos a utilizar.		
Campo	Tipo	
Resource-Type-id	int(11)	Identificador del tipo de recurso
Resource-Type-name	varchar(255)	Nombre del recurso
Resource-Type-note	text	Descripción del mismo

**Tabla 21 Descripción entidad Resources**

<b>Resources</b> Esta entidad almacena los recursos		
Campo	Tipo	
Resource-id	int(11)	Identificador del recurso
Resource-name	varchar(255)	Nombre
Resource-key	varchar(64)	Clave
Resource-type	int(11)	Tipo de recurso
Resource-note	text	descripción
Resource-may-allocation	int(11)	Porcentaje de utilización

**Tabla 22 Descripción entidad Resource\_Tasks**

<b>Resource-tasks</b> Entidad que almacena los recursos por tarea		
Campo	Tipo	
Resource-tasks-id	int(10)	Identificador de tareas por recurso
Resource-id	int(11)	Identificador de recursos
tasks-id	int(11)	Identificador de tarea
percent-allocated	int(11)	Porcentaje asignado para la tarea en el recurso

**Tabla 23 Descripción entidad Tasks**

Tasks		Entidad que almacena las tareas
Campo	Tipo	
task-id	int(11)	Identificador de tarea
task-name	varchar(255)	Nombre de la tarea
task-parent	int(11)	
task-milestone	tinyint(1)	
task-project	int(11)	Tarea por proyecto
task-owner	int(11)	
task-start-date	datetime	Tiempo de inicio de la tarea
task-duration	float	Tiempo de duración
task-duration-type	int(11)	Tiempo de tipo de duración
task-hours-worked	float	Horas de trabajo necesarios para la tarea
task-end-date	datetime	Fecha final de la tarea
task-status	int(11)	Estado de la tarea
task-priority	tinyint(4)	Prioridad de la tarea
task-percent-complete	tinyint(4)	Porciento completado de la tarea
task-description	text	Descripción de la tarea
task-target-budget	decimal(10,2)	
task-related-url	varchar(255)	
task-creator	int(11)	
task-order	int(11)	
task-client-publish	tinyint(1)	
task-dynamic	tinyint(1)	
task-access	int(11)	Usuario de acceso a la tarea
task-natify	int(11)	
task-departments	char(100)	Departamento el cual solicita la tarea
task-contacts	char(100)	Contactos de la tarea
task-custom	longtext	
task-type	smallint(6)	

**Tabla 24 Descripción entidad Projects**

Projects		Entidad que almacena los proyectos
Campo	Tipo	
Project-id	int(11)	Identificador de proyecto
Project-compny	int(11)	Compañía a la cual pertenece el proyecto
Project-department	int(11)	Departamento correspondiente al proyecto
Project-name	varchar(255)	Nombre del proyecto
Project-short-name	varchar(10)	Nombre abreviado del proyecto
Project-owner	int(11)	
Project-url	varchar(255)	Dirección electrónica del proyecto
Project-demo-url	varchar(255)	Otra página
Project-start-date	datetime	Tiempo inicial del proyecto
Project-end-date	datetime	Tiempo final del proyecto
Project-actual-end-date	datetime	Fecha del proyecto terminado
Project-status	int(11)	Estado del proyecto
Project-percent-complete	tinyint(4)	Porcentaje completado
Project-color-identifier	varchar(6)	Color que identifica el proyecto
Project-description	text	Descripción del proyecto
Project-target-budget	decimal(10,2)	
Project-actual-budget	decimal(10,2)	
Project-creator	int(11)	
Project-private	tinyint(3)	
Project-departments	char(100)	
Project-contacts	char(100)	Contactos del proyecto
Project-priority	tinyint(4)	Prioridad del proyecto
Project-type	smallint(6)	Tipo de proyecto

**Tabla 25 Descripción entidad users**

users		Entidad que almacena los usuarios
Campo	Tipo	
users-id	int(11)	Identificador de usuarios
users-contact	int(11)	Identificador de contactos de usuarios
users-username	varchar(255)	Usuario de acceso
users-pasword	varchar(32)	Contraseña de usuario
users-parent	int(11)	
users-type	tinyint(3)	Tipo de usuario
users-company	int(11)	Compañía a la cual pertenece
users-department	int(11)	Departamento al cual pertenece
users-owner	int(11)	
users-signature	text	

## Tablas creadas para la realización de los módulos

**Tabla 26 Descripción entidad Resource\_Users**

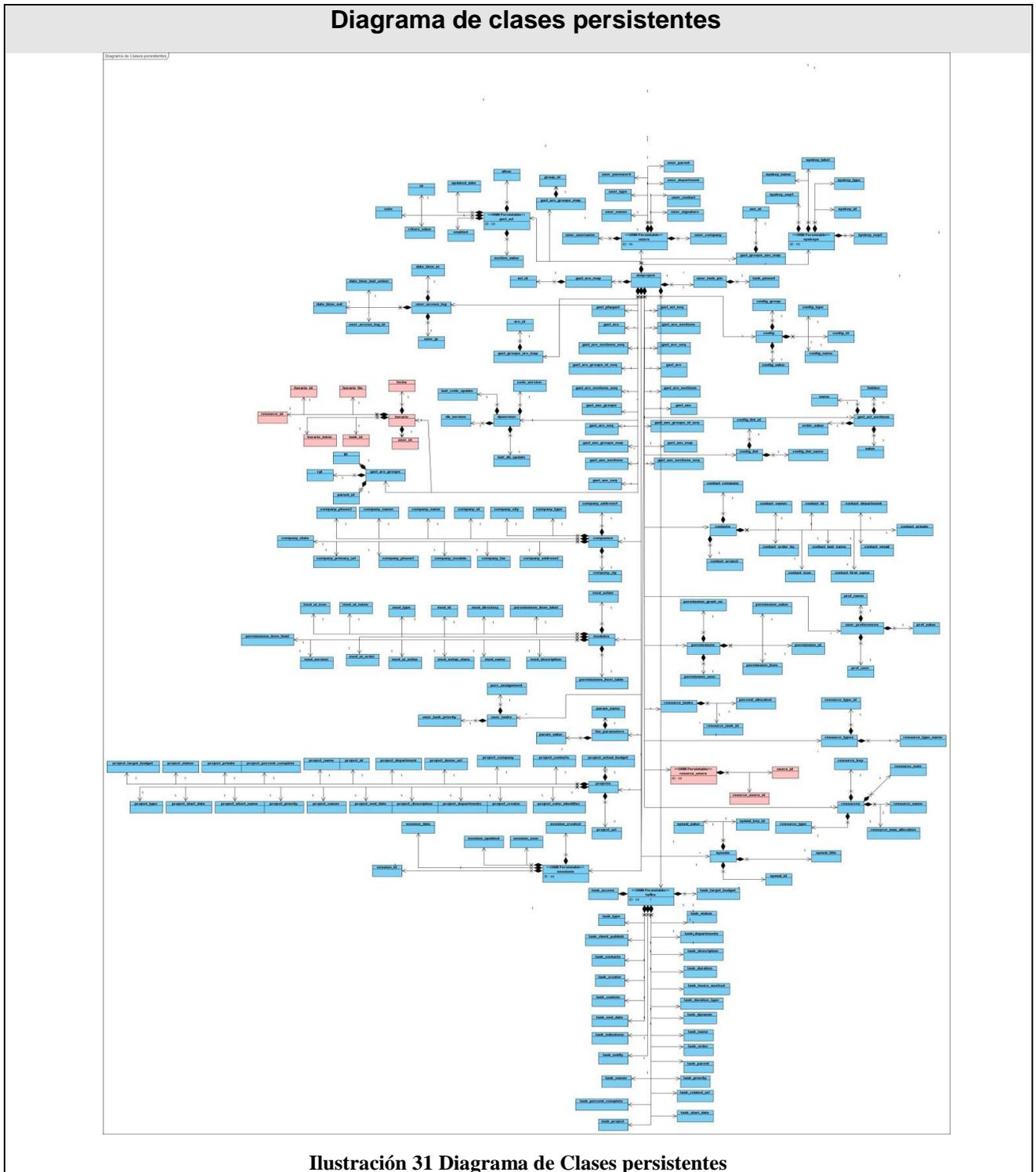
Resource-users		Entidad que almacena los Recurso asignado por usuario
Campo	Tipo	
Resource-users-id	int(10)	Identificador de recurso asignado a usuario
Resource-id	int(10)	Identificador de recurso
users-id	int(10)	Identificador de usuarios

**Tabla 27 Descripción entidad Horario**

Horario		Entidad que almacena el horario por usuarios y recursos
Campo	Tipo	
Horario-id	int(10)	Identificador de horario
fecha	date	Fecha asignada para el horario
user-id	int(11)	identificador de usuario
resource-id	int(11)	Identificador de recurso
task-id	int(11)	Identificador de tarea
horario-inicio	time	Horario de inicio
horario-fin	time	Horario final



3.7 Diagrama de clases persistentes



### 3.8 Diagrama de despliegue

La aplicación estará dispuesta para todo aquellos que necesiten una planificación del horario contra tareas, la cual es genérica y la podemos utilizar para cualquier proyecto, definiendo los recursos a usar y las tareas de las mismas.

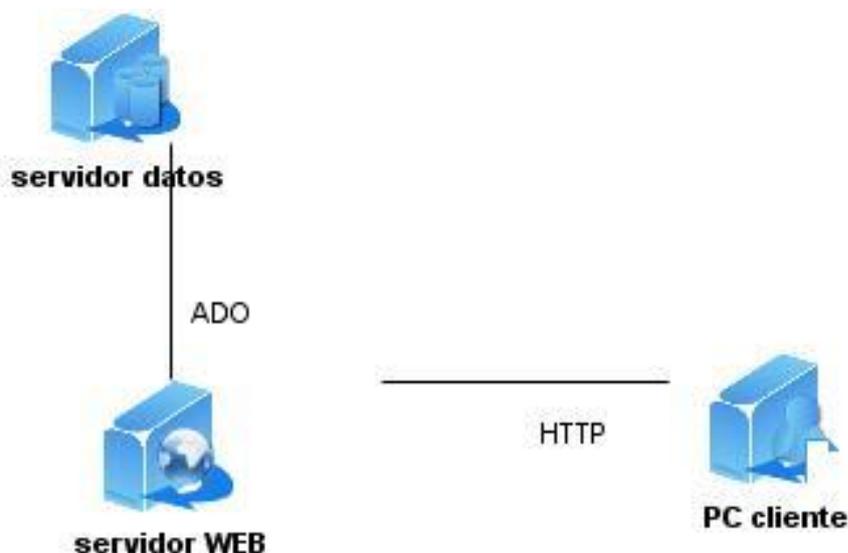


Ilustración 32 Diagrama de despliegue

### 3.9 Definiciones de diseño

Para el desarrollo de una aplicación se llevan de la mano una serie de responsabilidades que dan un funcionamiento correcto y acorde a los requerimientos y diseño del Dotproject. Para dar cumplimiento a dichos requerimientos se trabajó con formularios web organizados, utilizando un mismo color, predominando el azul en todas las páginas y un mismo tipo y tamaño de letra con textos claros. Se utilizó Ajax para recibir y enviar información a la base de datos, evitando un continuo refrescamiento de las páginas a la hora de seleccionar algunos campos que necesitan información de la base de datos.

La aplicación mostrará siempre una pestaña para garantizar una mejor navegabilidad al usuario.

### **3.10 Tratamiento de errores**

La aplicación constará con un chequeo de errores, inmediatos, para cada uno de los campos existentes en cada una de las interfaces. Con estas validaciones se garantiza que los datos que sean enviados no estén en blanco y tengan el formato correcto. Con el uso de las expresiones regulares en las funciones Javascript se garantiza el formato correcto para campos que sean números, letras, fechas y demás. También las validaciones se hacen tanto del lado del cliente como del servidor, pues con mayor seguridad si esta desactivado el Javascript podemos tener seguridad que los datos guardados sean válidos.

La aplicación garantizará que el usuario teclee la menor cantidad de datos posibles, para esto se contará con cuadros de opción, menú de selección lo cual facilitará la entrada de datos.

### **3.11 Seguridad**

La seguridad de la aplicación será responsabilidad del Dotproject, el cual implementará una jerarquía de accesos para los diferentes usuarios del sistema ya que todos no tendrán los mismos privilegios sobre las posibles acciones a realizar, no obstante los módulos tiene su propia seguridad implementada.

### **Conclusiones**

En el capítulo que concluye para poder detallar descriptivamente la solución propuesta, se ha representado la información referente a la etapa de diseño, se mostraron los diagramas de clases del diseño, la descripción detallada de los casos de uso más importantes del sistema así como los diagramas de secuencia por cada escenario donde se obtuvo una visión cercana a la etapa de implementación de los módulos a desarrollar. Los resultados serán utilizados como entrada en la siguiente etapa de trabajo.



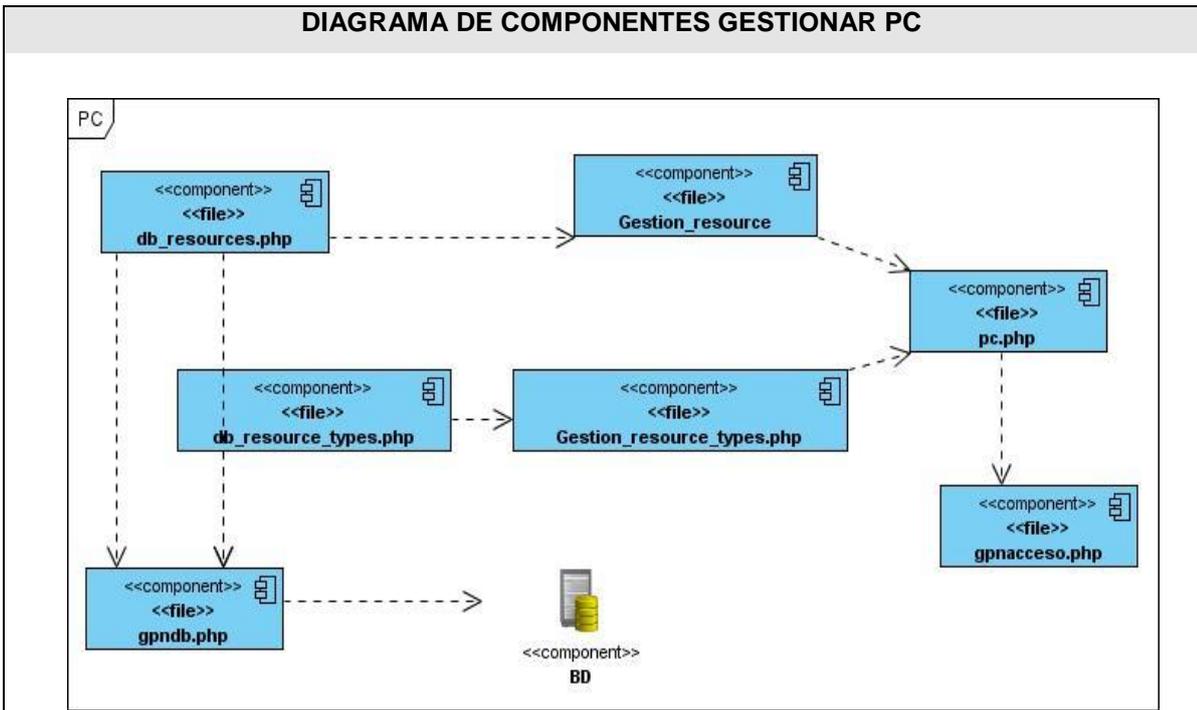
### **CAPÍTULO Implementación**

#### **Introducción**

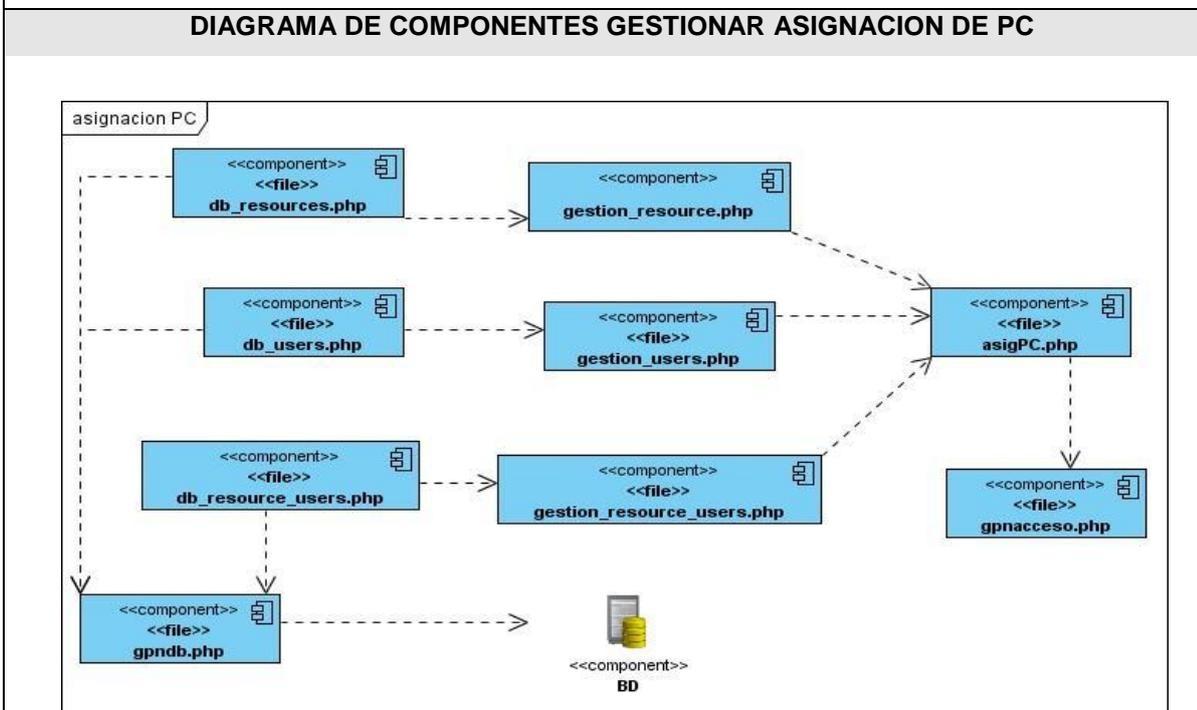
El siguiente capítulo muestra las definiciones de los diferentes modelos utilizados en la implementación para la solución del problema y se procede a la construcción de los requisitos.

#### **4.1 Diagrama de componentes**

El diagrama de componentes es un diagrama que muestra un conjunto de elementos del modelo, tales como componentes, subsistemas de implementación y sus relaciones. Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software, sean estos componentes de código fuente, librerías, binarios o ejecutables.



**Ilustración 33** Diagrama de componente. Gestionar PC



**Ilustración 34** Diagrama de componente. Gestionar asignación de PC

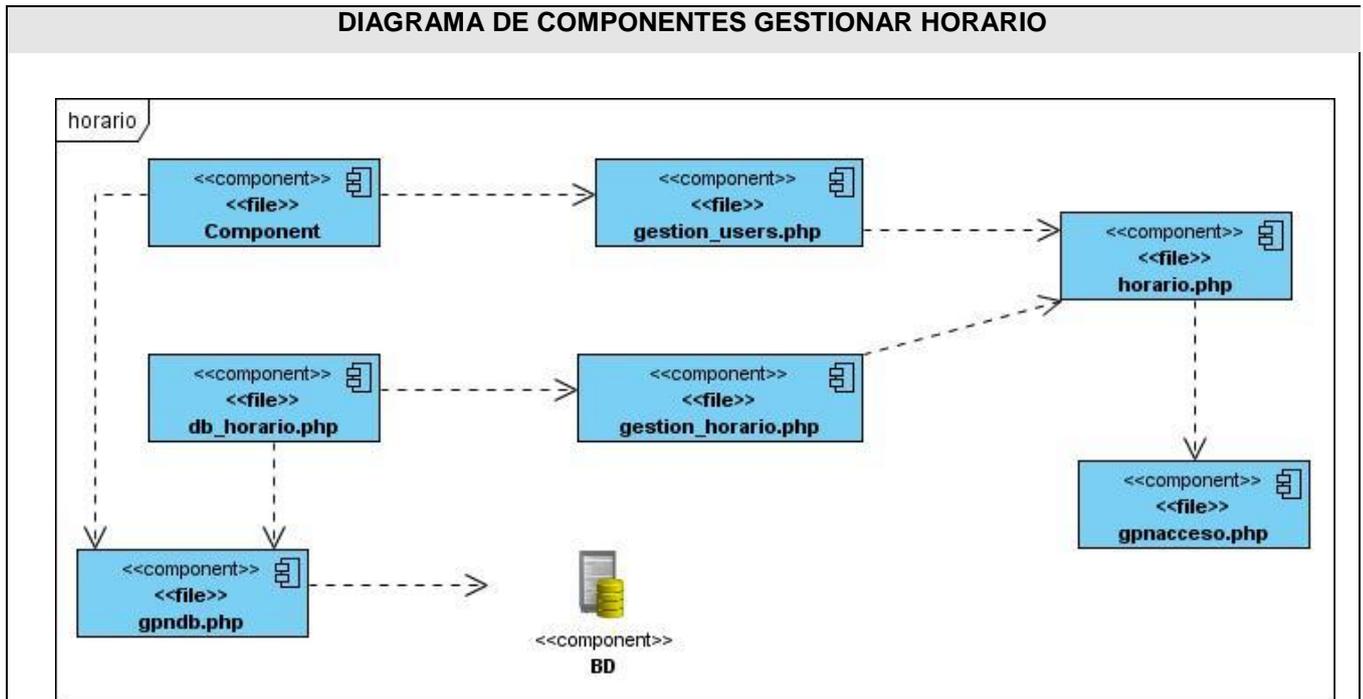


Ilustración 35 Diagrama de componente. Gestionar Horario

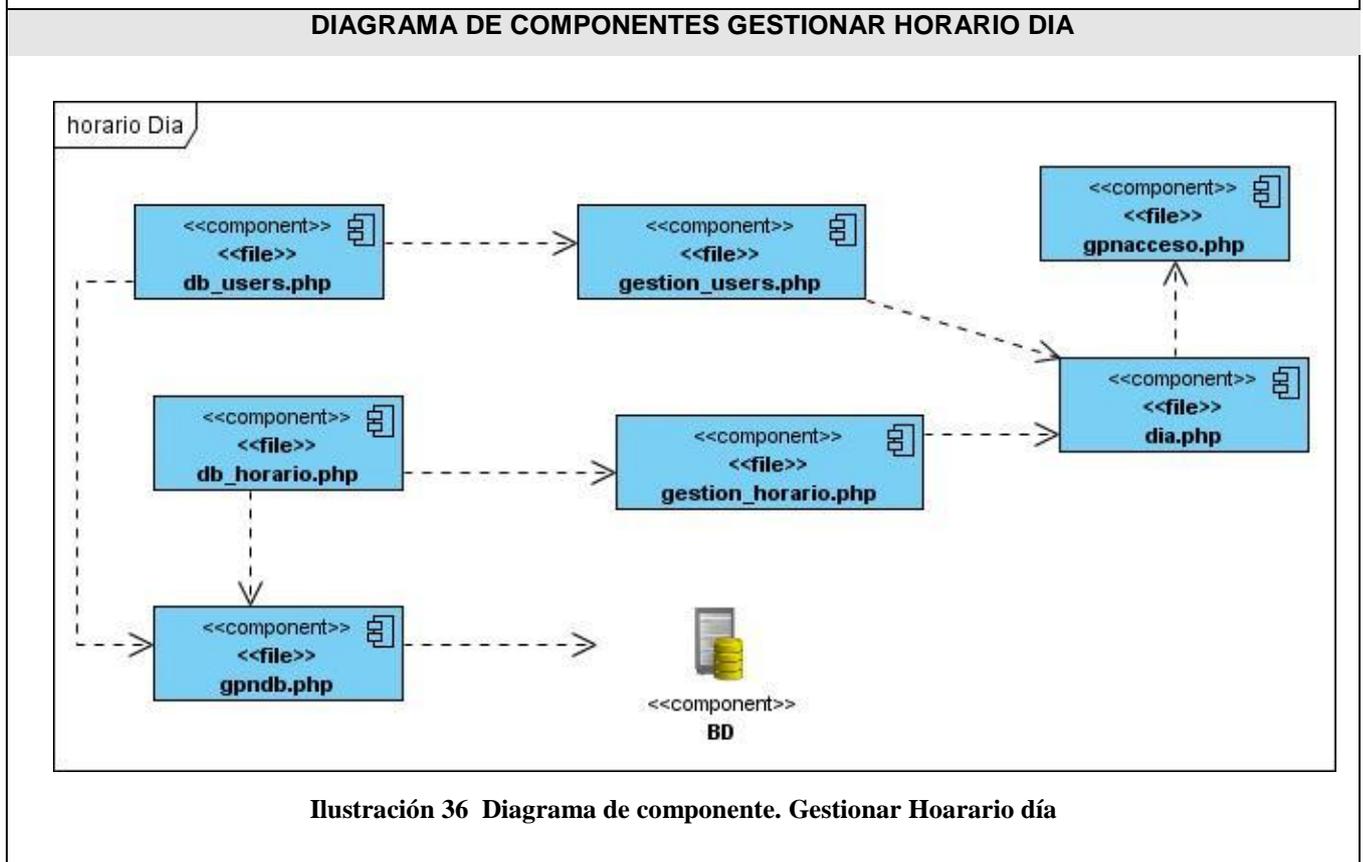
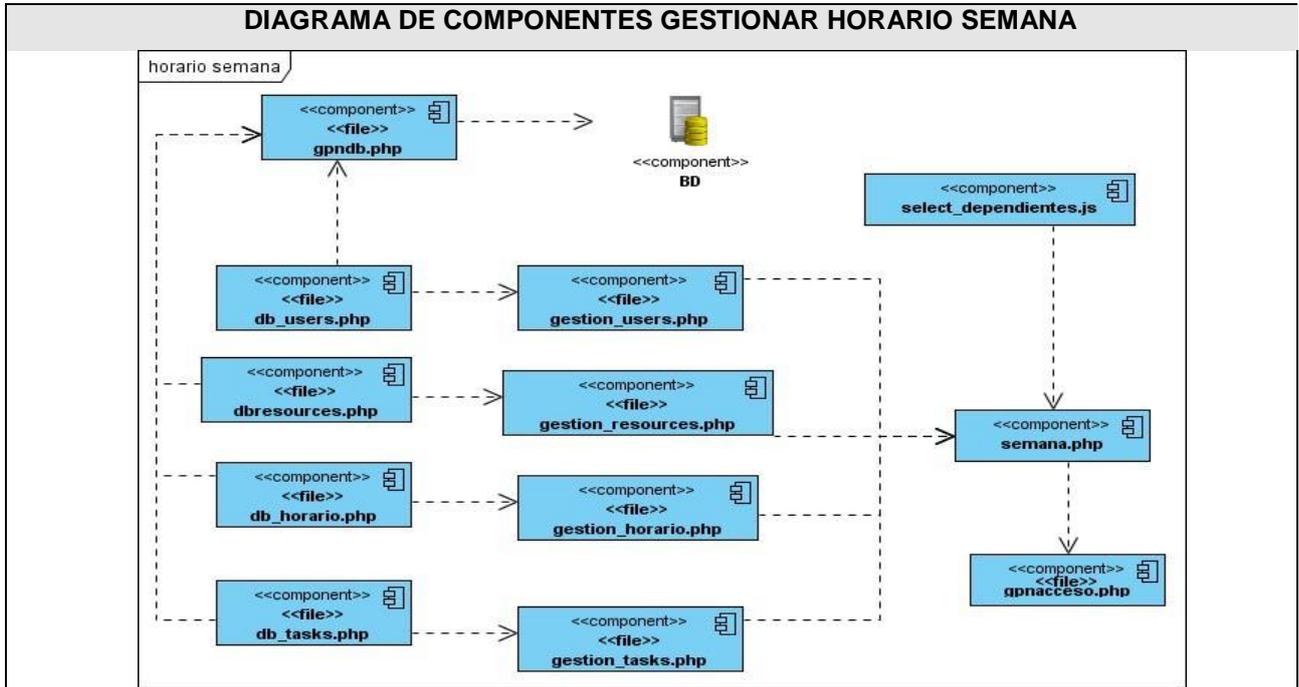
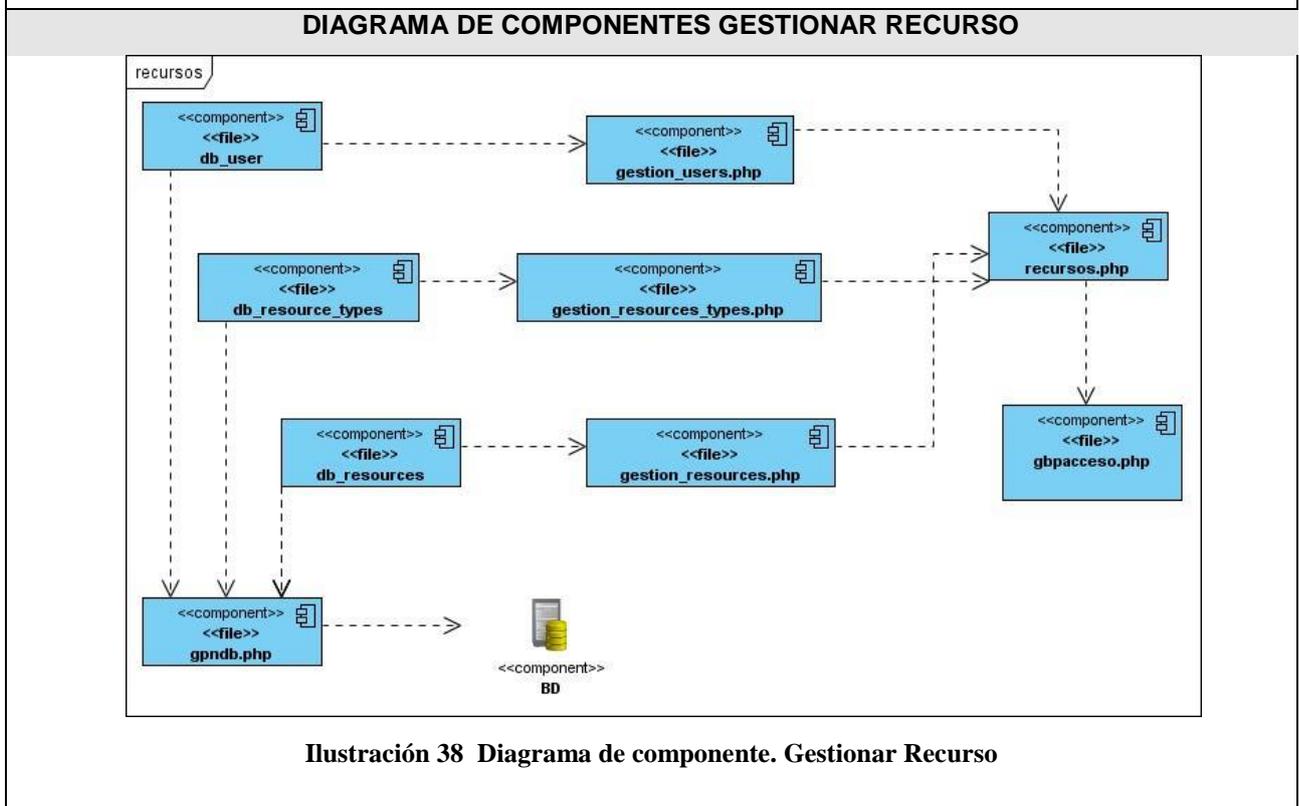


Ilustración 36 Diagrama de componente. Gestionar Hoarario día



**Ilustración 37** Diagrama de componente. Gestionar Horario Semana



**Ilustración 38** Diagrama de componente. Gestionar Recurso

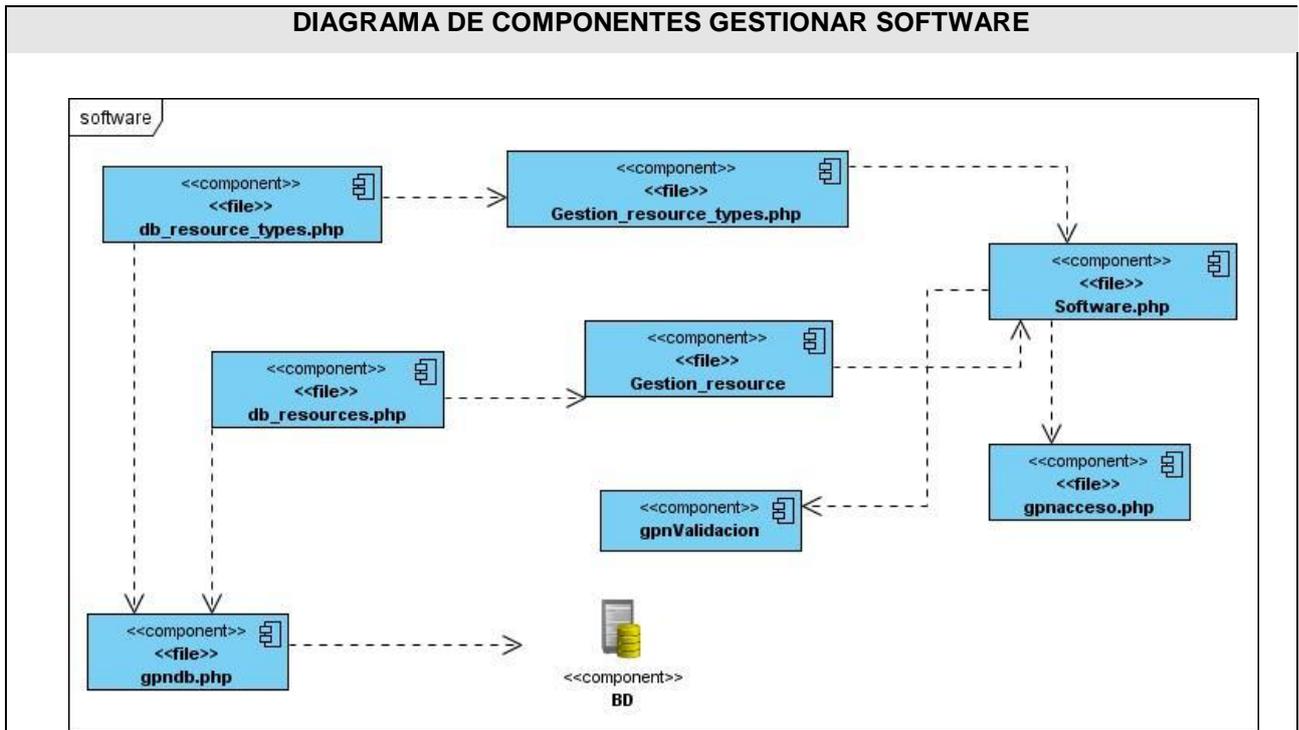


Ilustración 39 Diagrama de componente. Gestionar Software

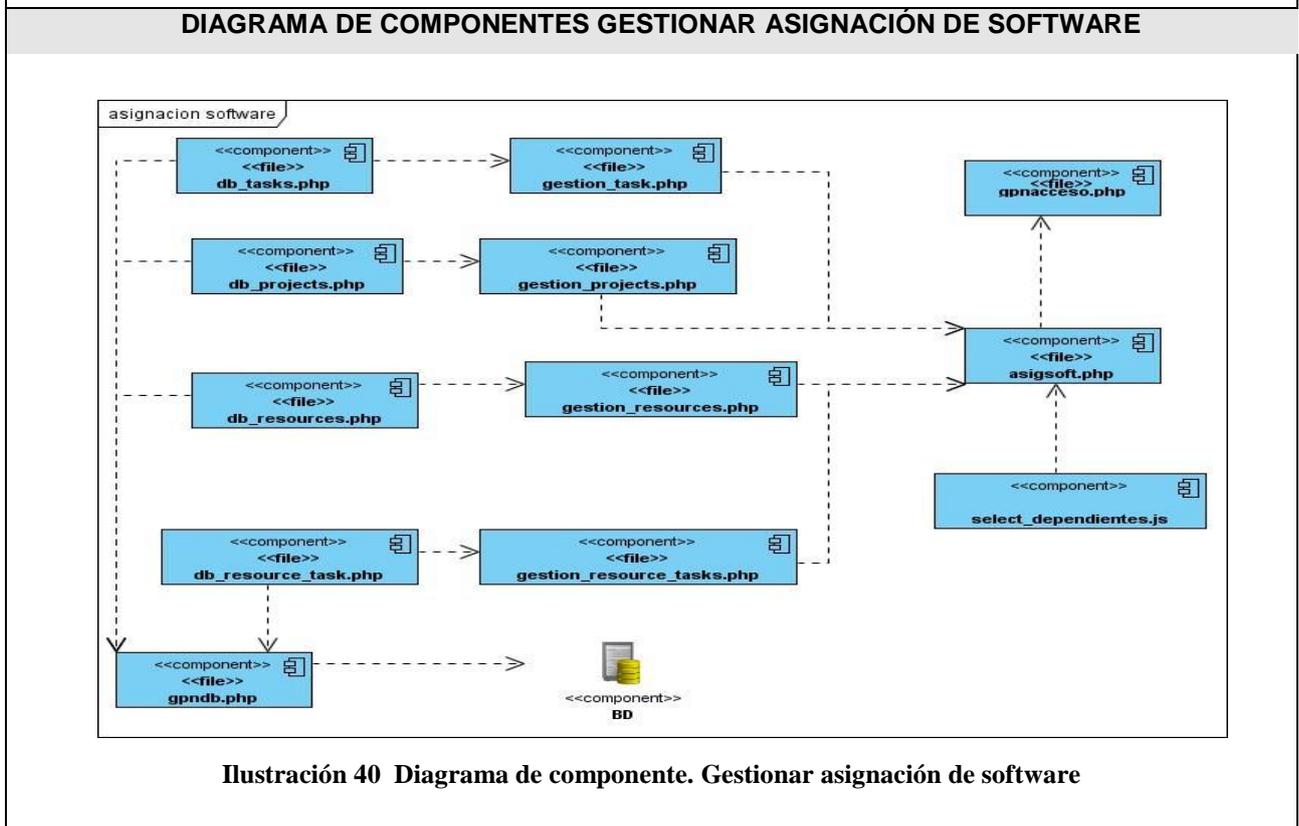
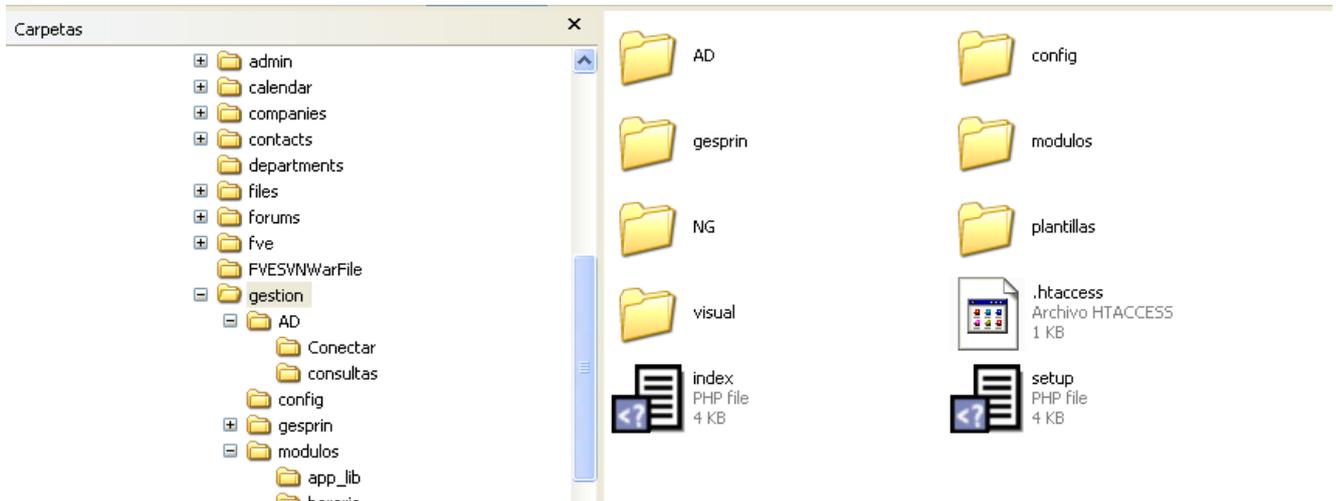


Ilustración 40 Diagrama de componente. Gestionar asignación de software

### 4.2 Modulo de la aplicación

Cuando se descomprime el paquete aparecen 7 carpetas más 3 archivos.



**Ilustración 41 Distribución de las Carpetas**

Breve descripción de algún de las carpetas más importantes:

El archivo Setup es el archivo de la instalación del modulo de gestión, el cual puedes tener acceso al mismo mediante la aplicación del Dotproject, en el modulo de agregar los mismo.

- AD:** (se configura la base de datos, parámetros del servidor de base de datos)
- Consultas:** (se guardan las consultas de cada tabla de la base de datos)
- config:** (corazón del proyecto, allí se configuran las características principales del proyecto)
- css:** (aquí se encuentran los css)
- gpn:** (aquí se encuentran los archivos de errores)
- js:** (aquí se encuentran los java script)
- módulos:** (en esta carpeta se organizan dentro de carpetas los distintos módulos del proyecto)
- inc:**(dentro de esta carpeta se pueden colocar los includes en común que tengan los módulos)
- js:**(aquí se colocan los java script include de los módulos del proyecto)
- NG:** (lógica de negocio del proyecto y validaciones de formularios)

## CAPITULO 4 IMPLEMENTACIÓN

**-validación:** (dentro de esta carpeta se encuentran las validaciones tanto del servidor como del cliente)

**-logica:**(aquí se colocan las clases que vaya a utilizar el proyecto, como parte de la lógica de negocio, estos archivos se llaman igual que la clase que contiene con el formato (*Gestion Tabla*))

**-gestión:**(clase de gestión del proyecto para cada tabla de la base de datos)

**-plantillas:** (en esta carpeta se guardan los distintos diseños que puede tener el proyecto, con sus css e imágenes y JS)

**-layout:** (en esta carpeta se guardan los layout de la plantilla de diseño)

La forma estructurada de las carpetas nos permite una mejor organización del trabajo, lo cual permite una adecuada comprensión para futuras modificaciones.

### **Conclusiones**

Con la realización de este capítulo se obtuvo una serie de artefactos desarrollados en la etapa de implementación. Los diagramas de componentes describieron la organización y dependencia entre nodos físicos en los que funciona la aplicación para dar cumplimiento a los requisitos planteados en el sistema.

## CONCLUSIONES GENERALES

### **Conclusiones generales**

Con la realización del presente trabajo se arribaron a las siguientes conclusiones:

Los módulos creados contribuirán al mejoramiento del trabajo en los proyectos, así como lograr una mejor distribución de los recursos y las tareas según el usuario, logrando organización del trabajo.

Permitirá además que los recursos no sea mal utilizado logrando una correcta organización de los mismo, y las tareas se cumplan con el tiempo requerido ya que nos permite asignar las mismas por usuario y en un horario específico y en una PC, quedando el trabajo de los proyectos mejor estructurado y organizado. También nos permite asignar a las tareas el software necesario para su realización, permitiendo la integridad en el software a utilizar.

Se considera además que el presente trabajo ha cumplido con los objetivos propuestos:

- Se realizó un análisis de la estructura e implementación del DotProject.
- Quedo definido los elementos del proceso de asignación de tareas y recursos.
- Quedó modificado el diseño de la base de datos del Dotproject, agregándole al mismo dos nuevas funcionalidades.
- Se implementaron dos nuevos módulos para la gestión de recursos y horarios en el Dotproject

## CONCLUSIONES GENERALES

### **Recomendaciones.**

Luego de haber concluido el presente trabajo se recomienda:

- Realizar pruebas de calidad para la liberación de los nuevos módulos.
- Confección de la ayuda y el manual de usuario.
- La realización un modulo que garantice el control del cumplimiento de las tareas asignadas en el horario previsto.

## Trabajos citados

1. **Drager, erich.** *Guía avanzada gestión de proyecto SAP R/3.* 1998. 12-13.
2. **Intitec.** [En línea] Netcraft Noticias, 2008.  
[http://news.netcraft.com/archives/web\\_serv](http://news.netcraft.com/archives/web_serv).
3. **R, Boullón Garzón y Rodríguez, Laredo P.** Intitec. *Proyecto Final de Carrera 2008.* [En línea] <http://www.intitec.com/variados/ComparativaS>.
4. **Zend Studio.** [En línea] <http://www.desarrolloweb.com/articulos/1178.php>.
5. **Zend.** [En línea] <http://www.zend.com/en/products/studio/>.
6. **Colectivo de autores.** UML y RUP, Clase . [En línea] 2007.  
<http://eva.uci.cu/mod/resource/view.p>.
7. **Conferencia 2 Fase de Inicio. Modelo del Negocio.** [En línea] 2007-2008.  
<http://eva.uci.cu/mod/resource/view.php?id=11553>.
8. **Colectivo de autores.** EVA. *fase de inicio. flujo de trabajo de requerimiento, ingeniería de software 1.* [En línea] 2008-2009. <http://eva.uci.cu>.
9. **Álvarez, Rubén Llovera.** *Planificación de proyecto.* Universidad de Salamanca : s.n., Febrero 2007.

## Bibliografía

**Álvarez, Rubén Llovera. Febrero 2007.** *Planificación de proyecto.* Universidad de Salamanca : s.n., Febrero 2007.

**Arce, Ronnie Fernández. 2008.** *Herramienta de ayuda para la confección de horario docente en la facultad 7.* [pdf] Cuba : UCI-facultad 7, 2008.

**Franco, Jose Angel Navarro. 2008.** *UML en acción. Modelado de Aplicaciones Web.* [doc] Cuba : CUJAE, 2008.

**González, Francy, Rodríguez, Ronald y Peérez, Kelly. Agosto 2007.** *Manual de DotProject.* [pfd] Venezuela : Ministerio del poder Popular para la educación (MPPE), Agosto 2007.

**Gonzalez, Joel Estrada. 2007.** Desarrollo Web con PHP y MySQL. *La Web del programador.* [En línea] 2007. <http://www.lawebdelprogramador.com>.

**Hernández, Roberto Sampieri, Fernández, Carlos Collado y Baptista, Pilar Lucio. Febrero 1998.** *Metodología de la investigación.* México : Programa educativo, S.A, Febrero 1998. 2.

**Jacobson, Ivar, Booch, Grady y James Rumbaugh.** El proceso unificado de desarrollo de software. [En línea] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.

**Ortiz, Maylen Ramos. 2008.** *Análisis y Diseño de un sistema para la planificación automatizada del horario docente de la facultad 4.* [pdf] Cuba : UCI-Facultad 4, 2008.

**Pereira, Yunek Salazar. 2008.** *Sistema de control de procesos en los ordenadores.* [pdf] Cuba : UCI, Facultad 4, 2008.

**Pérez, Yoandy Villazon.** *Curso de DotProject.* [ppt] Cuba : Proyecto Unicornios-Facultad 10.

**Schmuller, Joseph.** Biblioteca UCI. [En línea] <http://bibliodoc.uci.cu/pdf/reg00004.pdf>.

**Tamarit, Jorge Luis Cutiño y Santiesteban, Maykel Diaz. 2007.** *Asistente de ayuda para la confección de horario docente.* [pfd] Cuba : UCI, 2007.

## **Webgrafia**

[En línea] <http://office.microsoft.com/en-gb/project/FX100487771033.aspx>.

[En línea] <http://openproj.org/openproj>.

[En línea] <http://redmine.org>.

[En línea] <http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>.

[En línea] <http://www.php.net/archive/2008.php>.

[En línea] <http://www.php-es.com/>.

[En línea] <http://tutoriales.innox.com.mx/vhost/tutoriales/completo/index.html>.

[En línea] <http://www.visualparadigm.com/news/vpsuite40/vpuml70.jsp>.

### Glosario de Términos

**SO:** Sistema Operativo

**AJAX:** (Asynchronous Javascript And XML): Es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

**PHP:** (Hypertext PreProcessor): Lenguaje de programación para el desarrollo de web dinámicas, con sintaxis parecida a la C. Originalmente se conocía como *Personal Home Page tools*, herramientas para páginas personales en Internet.

**PERT:** Las traducción de las siglas en inglés significan: técnica de revisión y evaluación de programas, es una técnica de redes desarrollado en la década de los 50, utilizada para programar y controlar programas a realizar. Cuando hay un grado extremo de incertidumbre y cuando el control sobre el tiempo es más importante sobre el control del costo, PERT es mejor opción que CPM.

**MSP:** Microsoft Project, programa para la gestión de proyecto.

**Qt o QTK:** Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario multiplataforma para desarrollar interfaces gráficas de usuario (GUI).

**GUI:** Siglas de graphical user interface. Permite a los usuarios navegar e interactuar con las informaciones en la pantalla de su ordenador utilizando un ratón para señalar, pulsar y desplazar iconos y otros datos de la pantalla, en lugar de escribir palabras y frases. Los sistemas operativos de Windows y Macintosh son ejemplos de GUIs.

## BIBLIOGRAFÍA

**FTP:** Siglas de File Transfer Protocol. Método muy común para transferir uno o más ficheros de un ordenador a otro. FTP es un medio específico de conexión de un sitio Internet para cargar y descargar ficheros. FTP fue desarrollado durante los comienzos de Internet para copiar ficheros de un ordenador a otro.

**SGBD:** Sistema gestor de Base de Datos

**IDE:** Entorno de desarrollo integrado o en inglés Integrated Development Environment, es un programa compuesto por un conjunto de herramientas para un programador.

**RUP:** (Rational Unified Process) 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.

**UML:** Conjunto de herramientas, que permite modelar, analizar y diseñar, sistemas orientados a objetos.

**CMS:** (Content Management System) Sistema de gestión de contenidos, permite la creación y administración de contenidos, principalmente en páginas web. Consiste en una interfaz (unos menús y botones para manejar el sistema) que controla una o varias bases de datos donde se almacena el contenido la información o datos del sitio.

**MVC:** Modelo Vista Controlador

**CVS:** (Concurrent Version System) Controlador de versiones

# Anexos

Diagrama de Secuencia

Eliminar PC

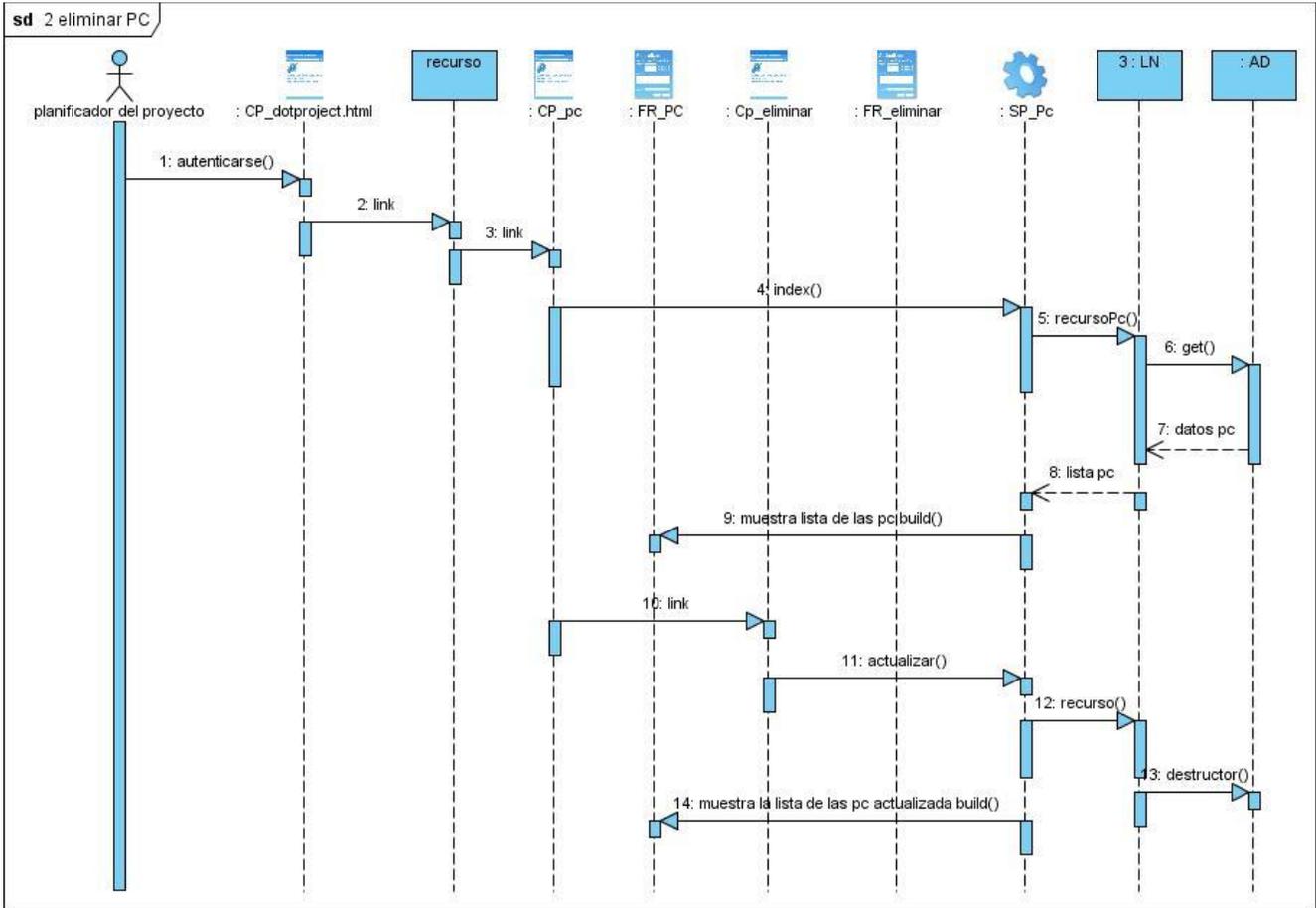
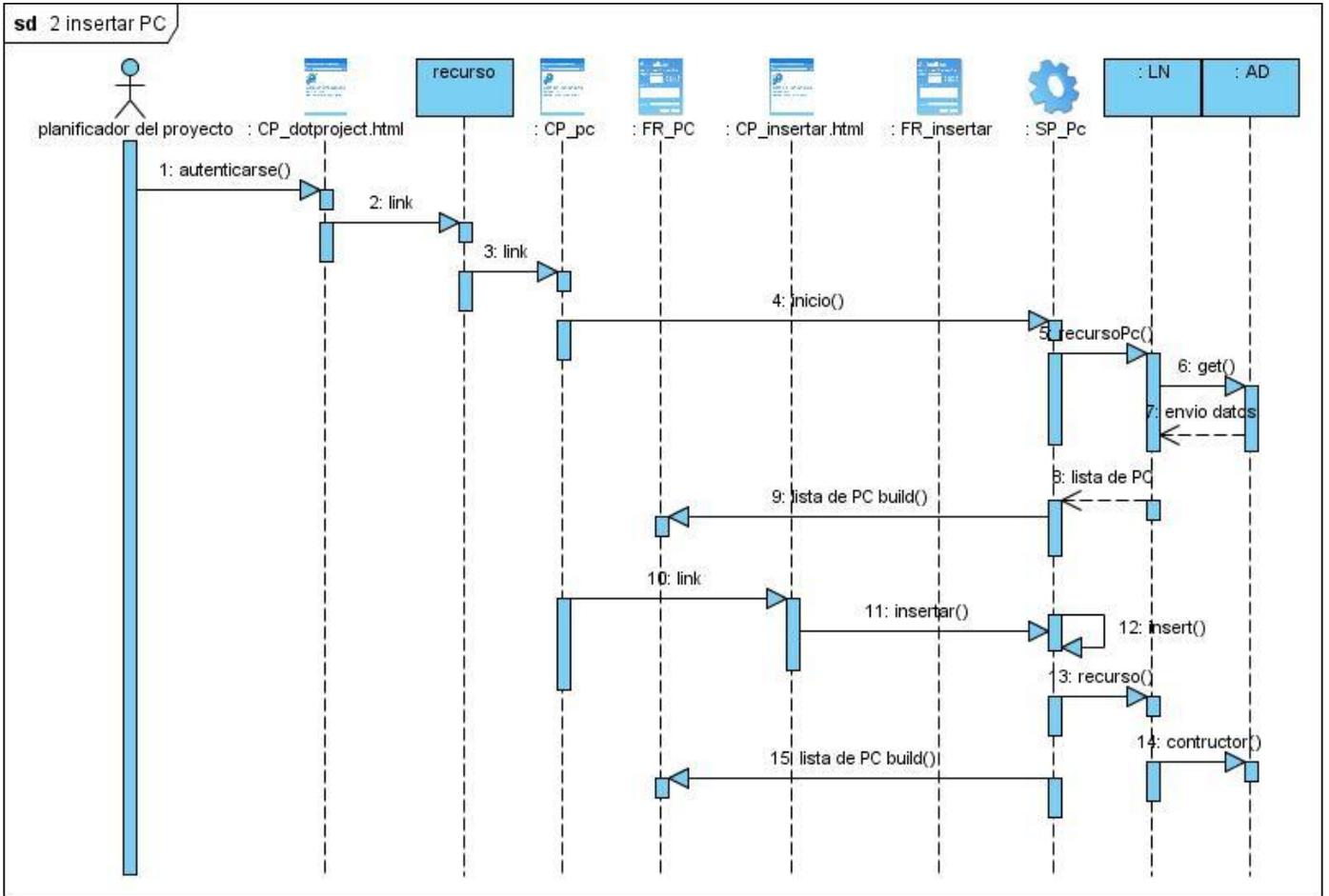


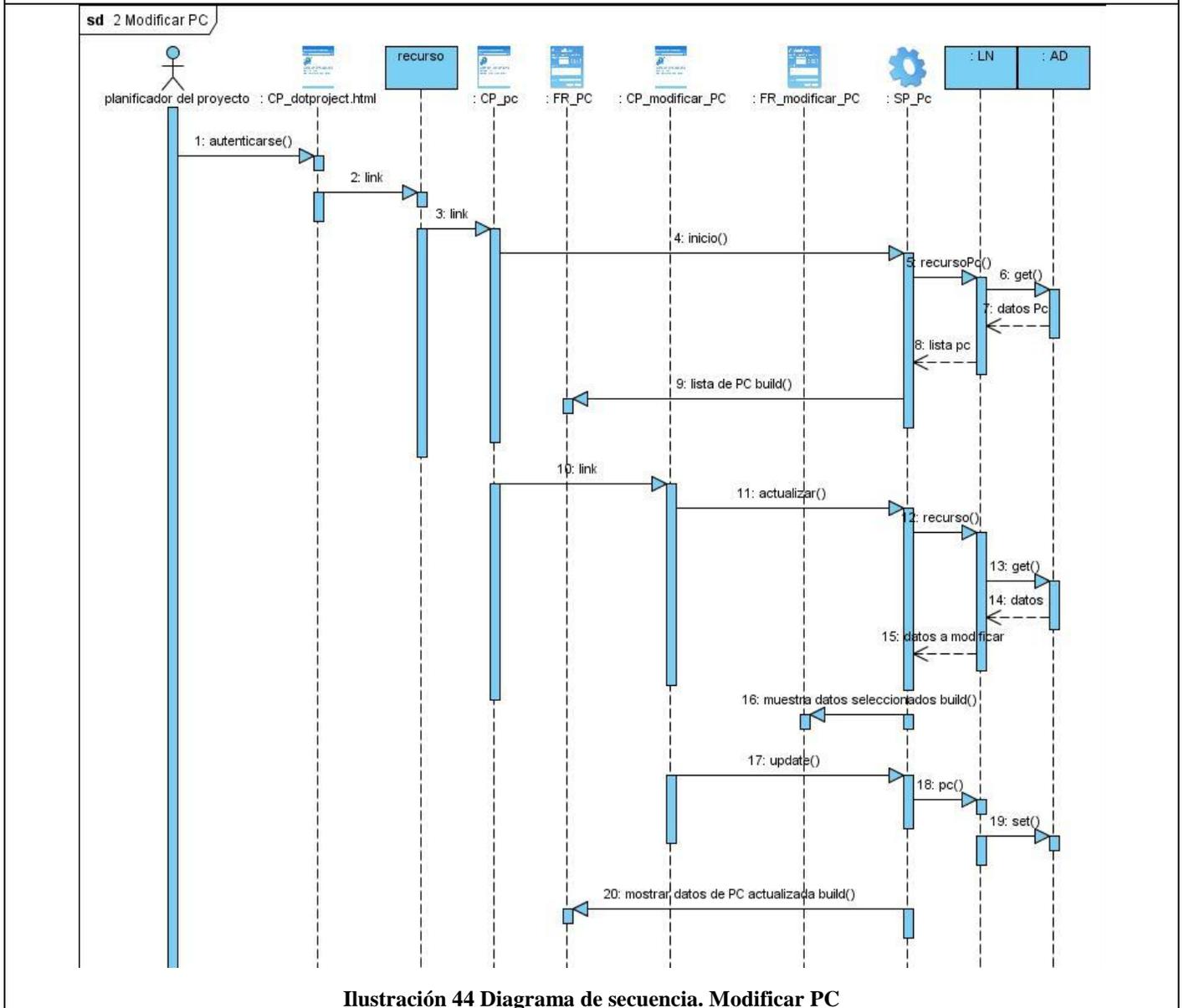
Ilustración 42 Diagrama de Secuencia. Eliminar PC

ANEXOS  
**Insertar PC**



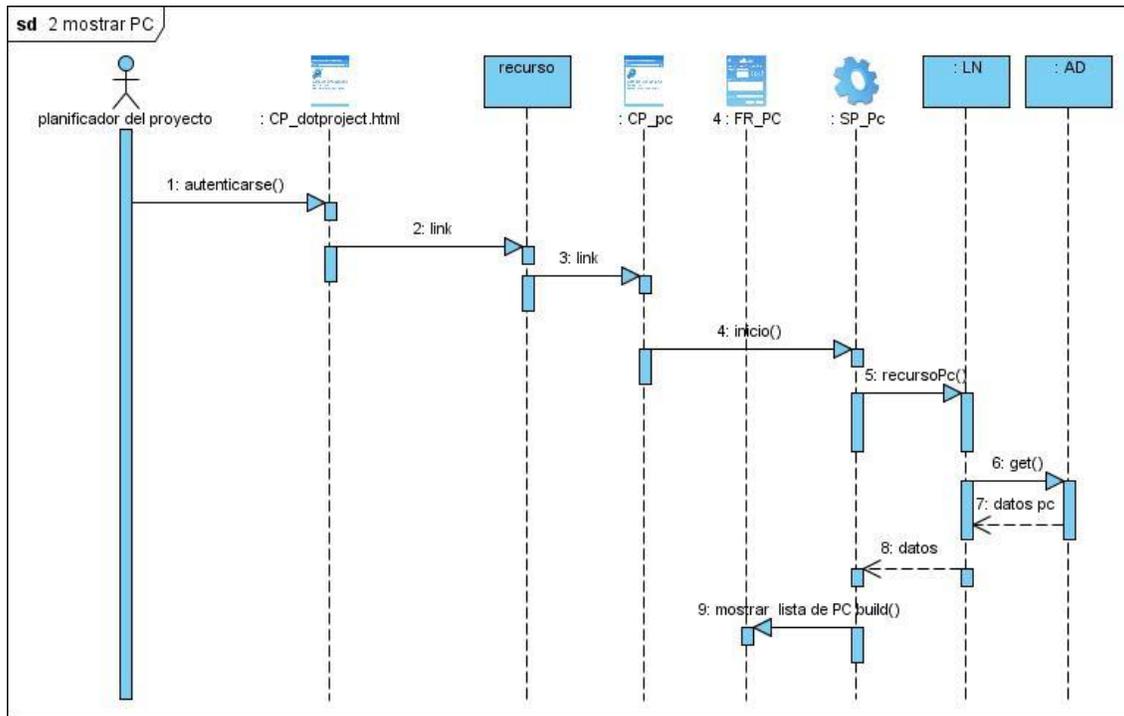
**Ilustración 43 Diagrama de Secuencia. Insertar PC**

**Modificar PC**



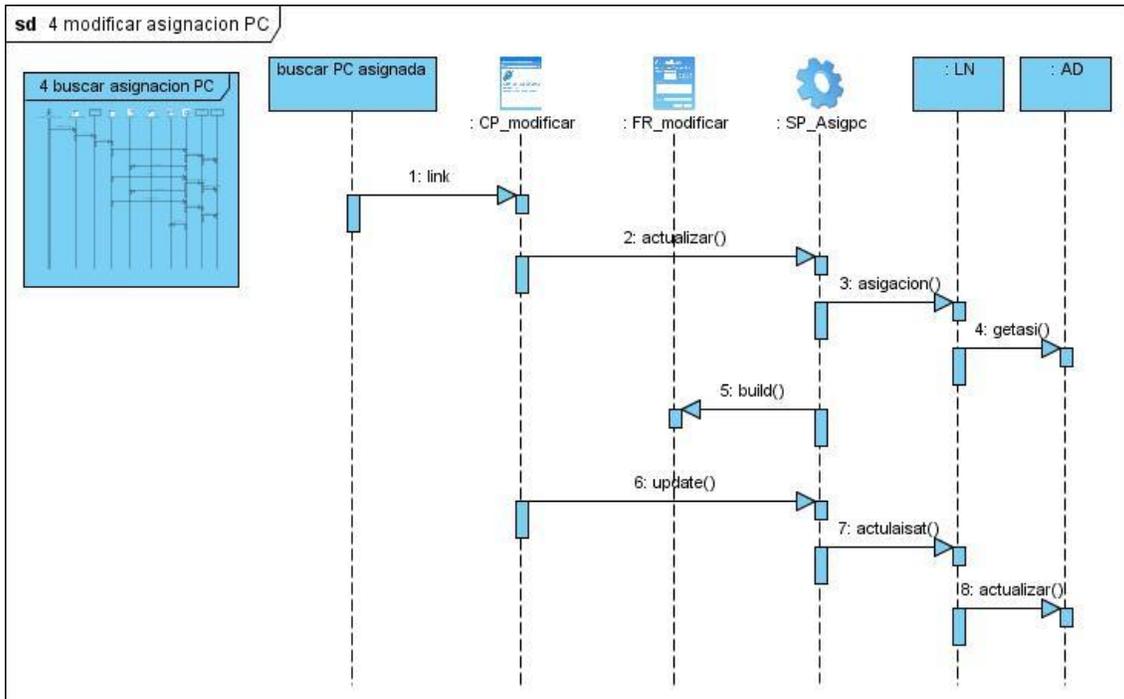
**Ilustración 44 Diagrama de secuencia. Modificar PC**

**Mostrar PC**



**Ilustración 45 Diagrama de secuencia Mostrar PC**

**Modificar Asignación PC**



**Ilustración 46 Diagrama de secuencia. Modificar asignación PC**

### Buscar Asignación de PC

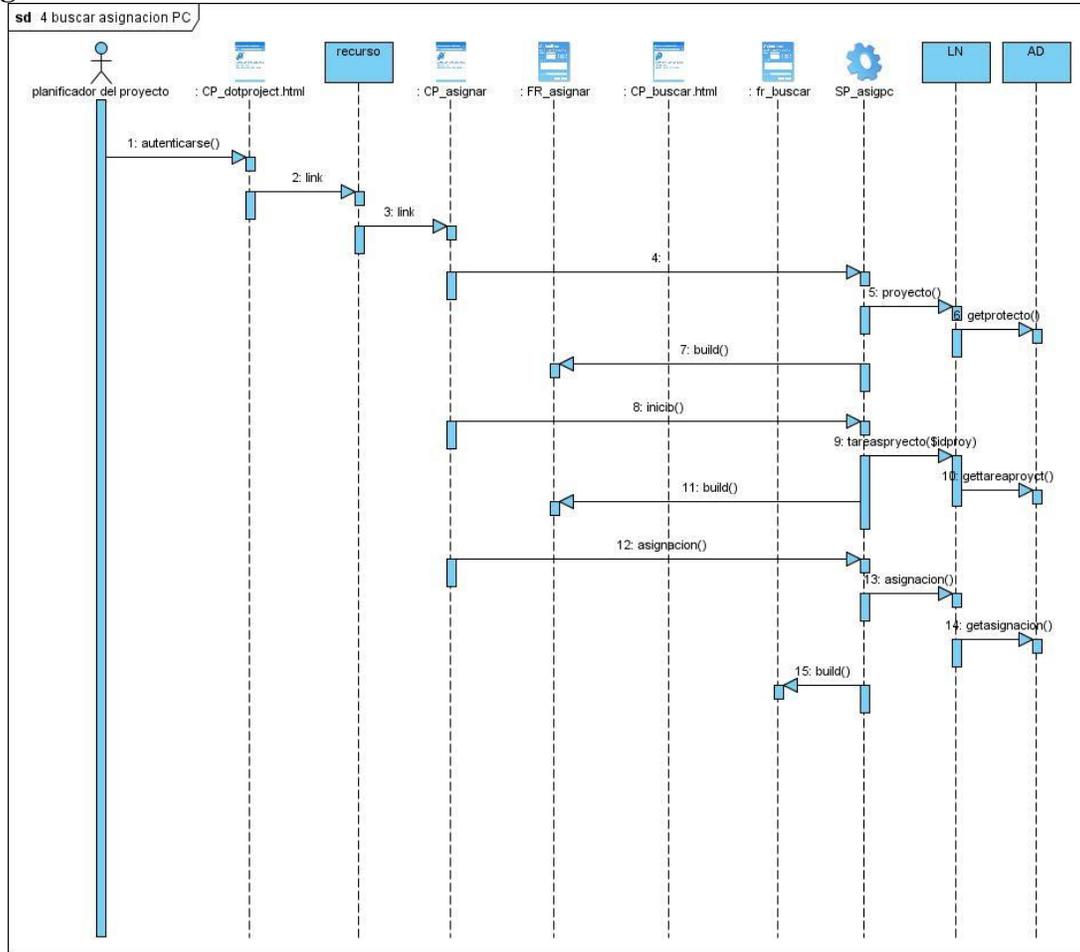


Ilustración 47 Diagrama de secuencia. Buscar Asignación de PC

### Eliminar Asignación de PC

sd 4 eliminar asignacion de PC

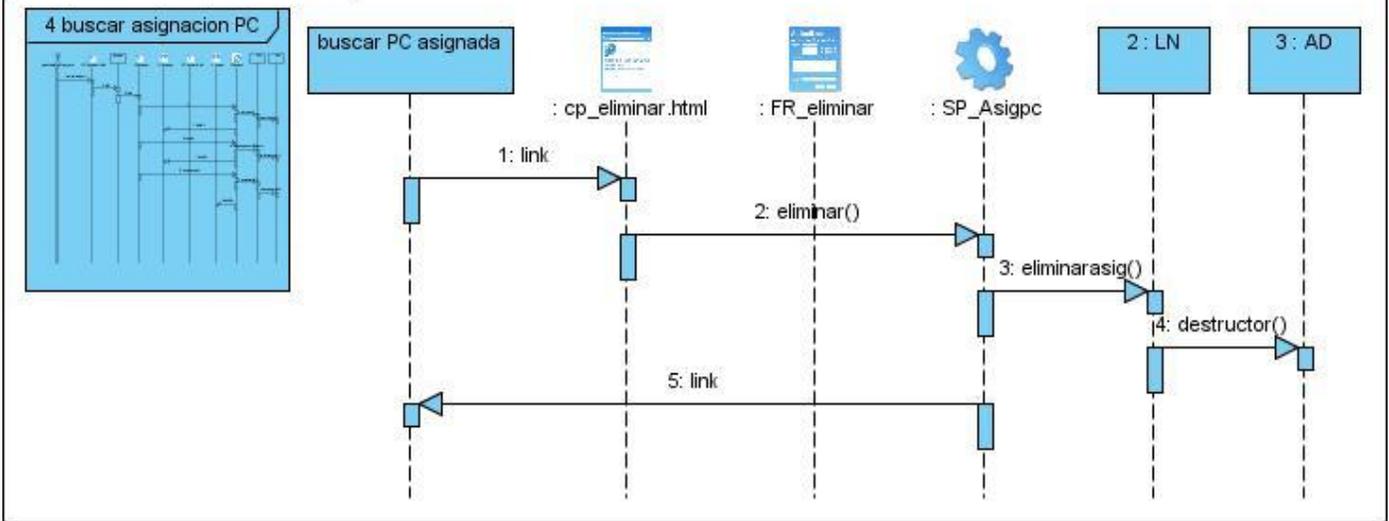


Ilustración 48 Diagrama de secuencia. Eliminar asignación de PC

ANEXOS

Insertar Asignación de PC

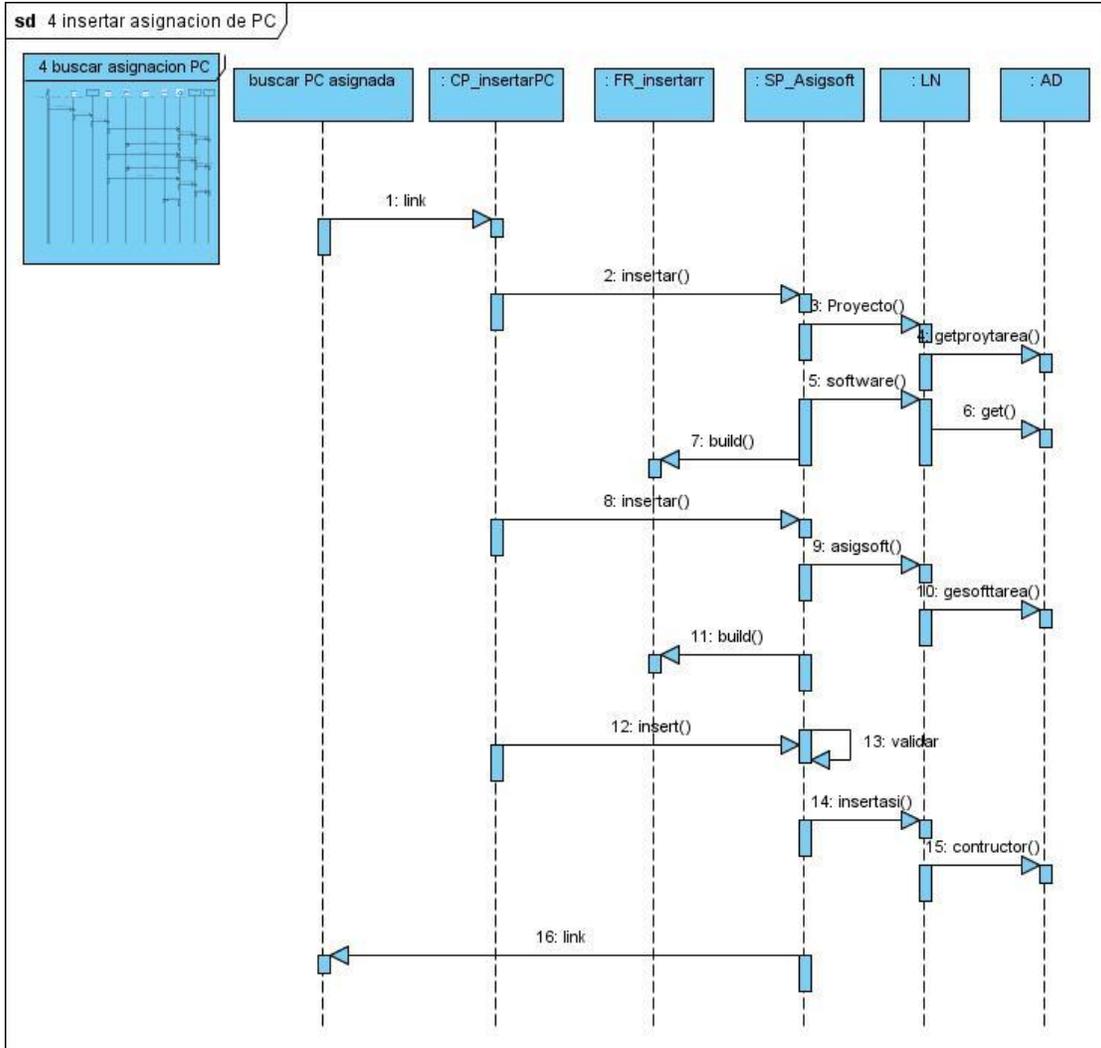
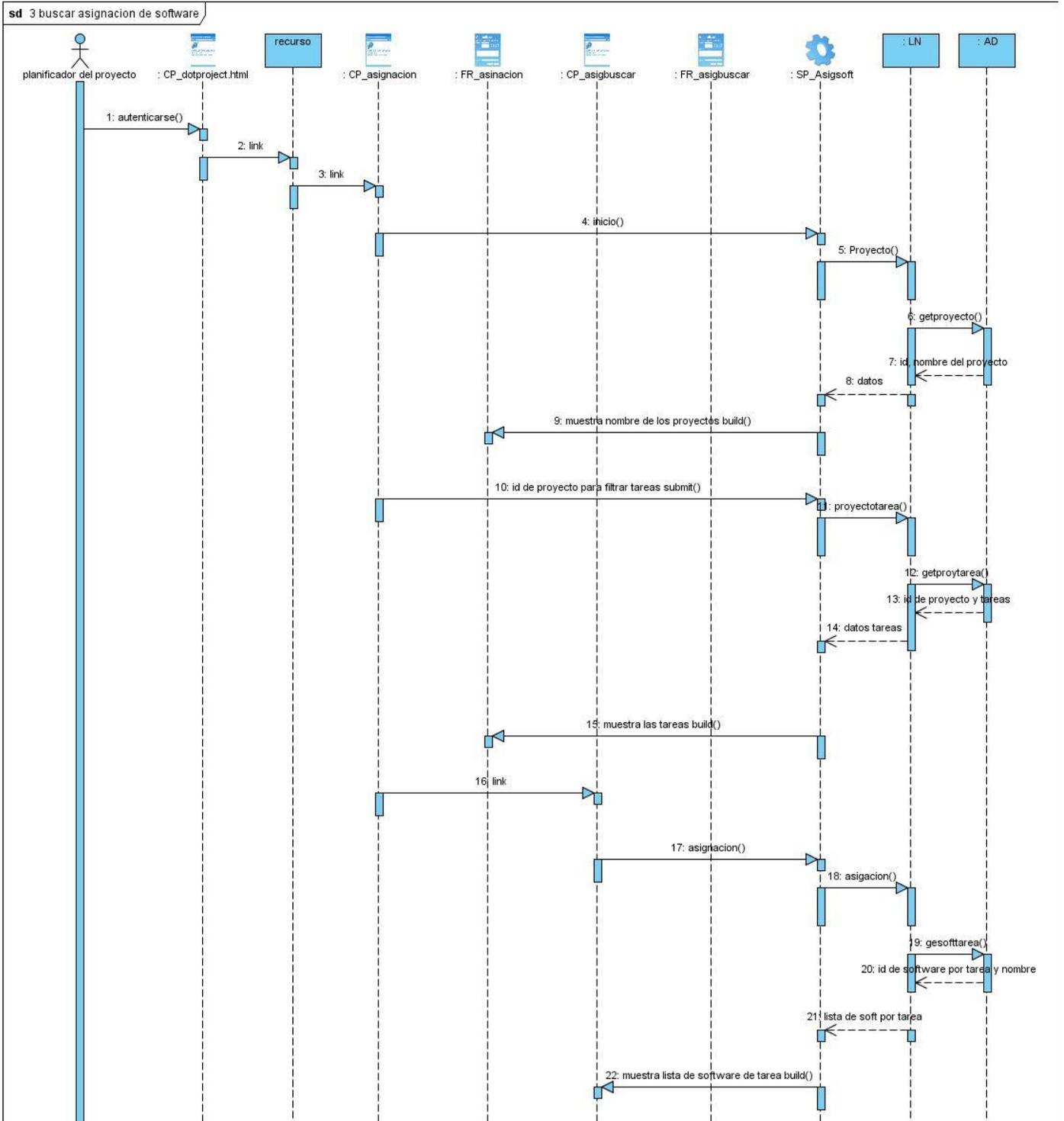


Ilustración 49 Diagrama de secuencia. Insertar asignación de PC

# Buscar Asignación de software



**Ilustración 50 Diagrama de secuencia. Buscar Asignación de software**

## Eliminar Asignación de software

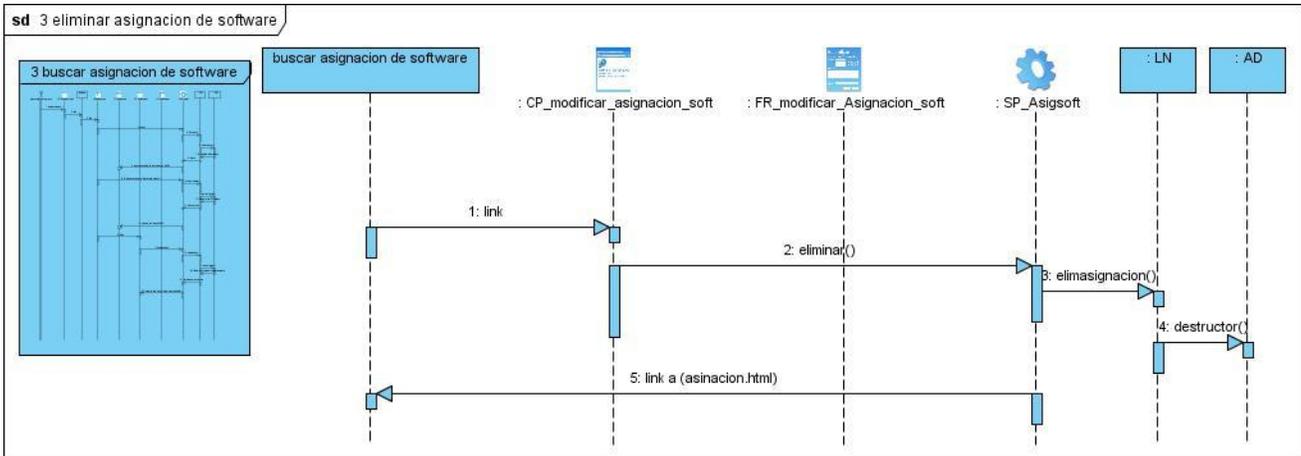


Ilustración 51 Diagrama de secuencia. Eliminar Asignación de software

## Modificar Asignación de software

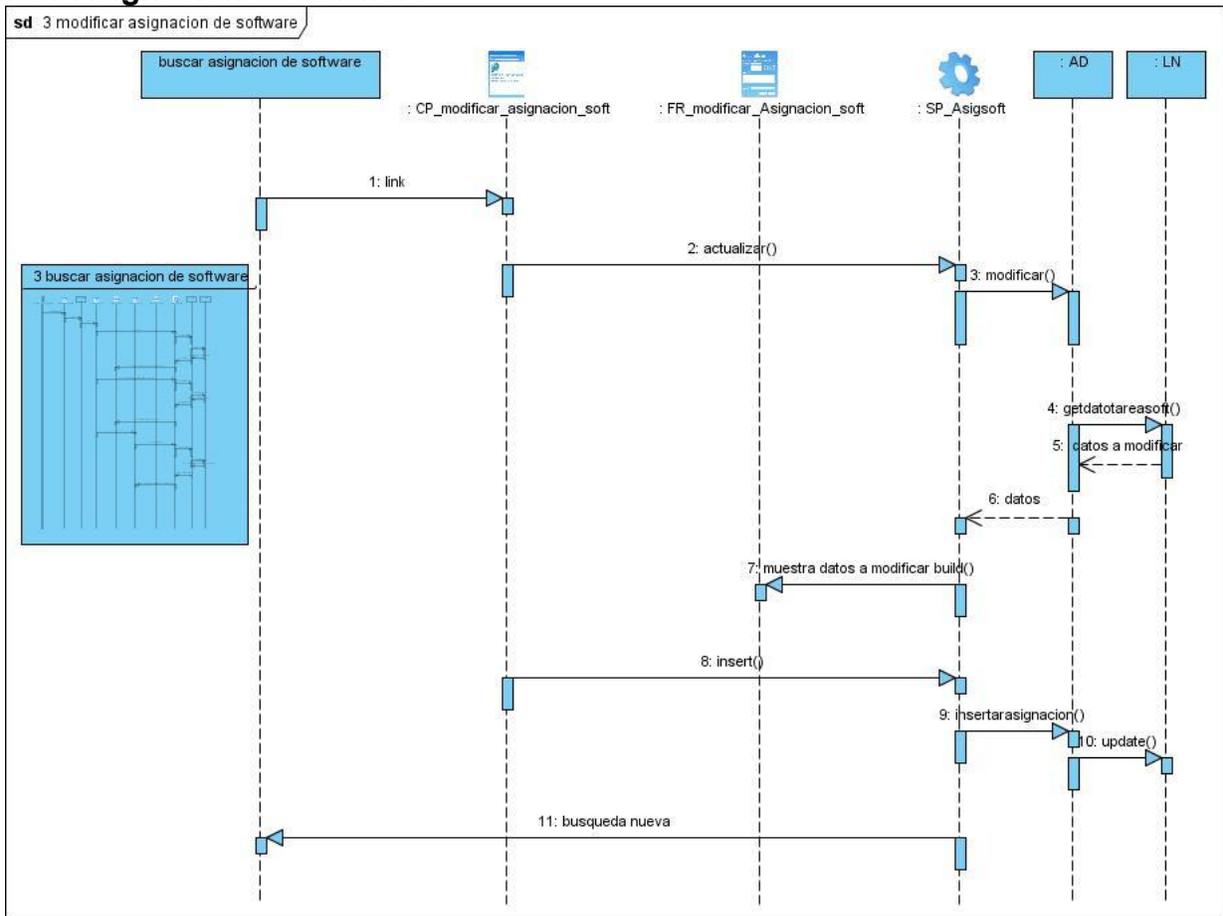


Ilustración 52 Diagrama de secuencia. Modificar Asignación de software

# Insertar Asignación de Software

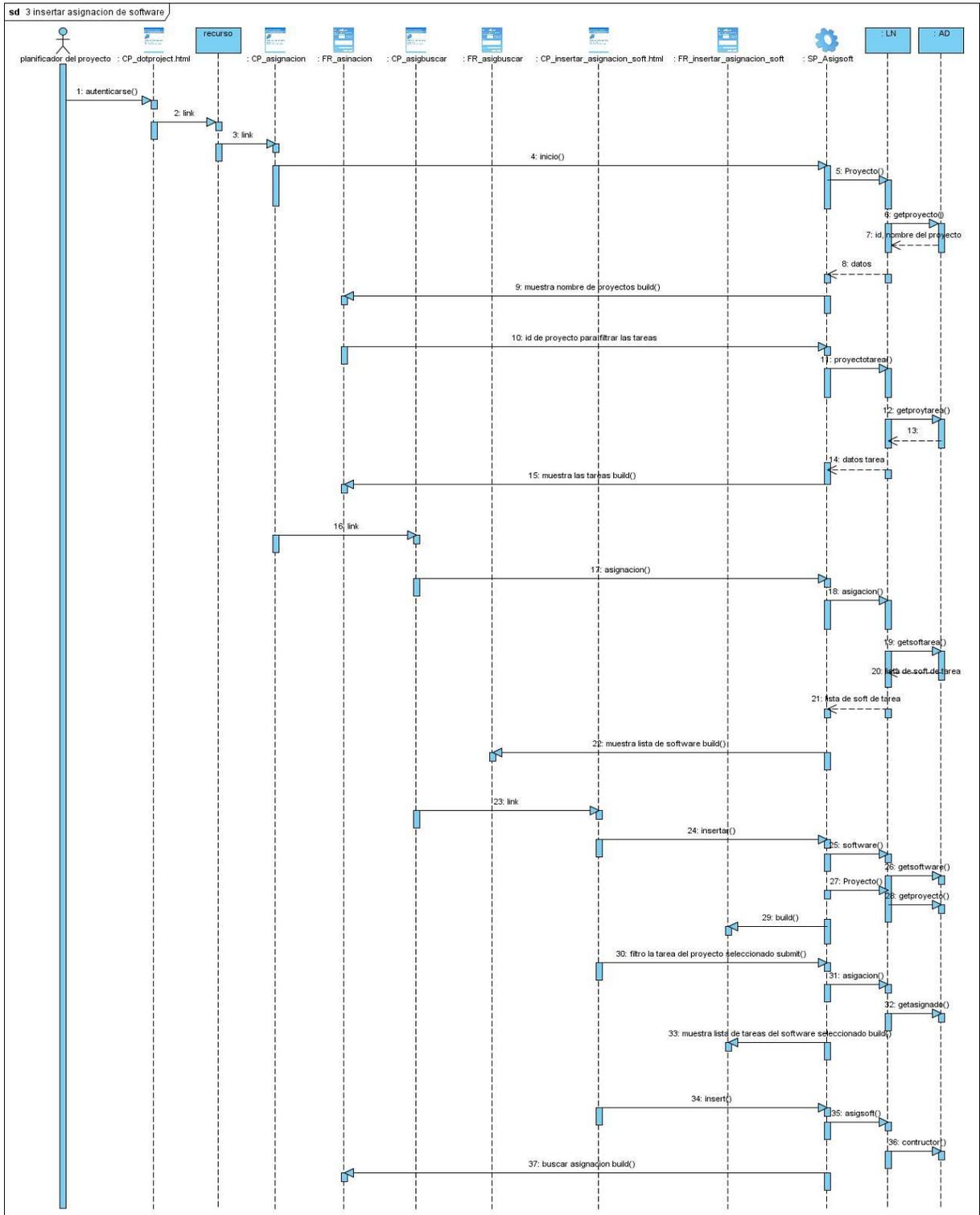


Ilustración 53 Diagrama de secuencia. Insertar Asignación de software

# ANEXOS

## Mostrar Recursos

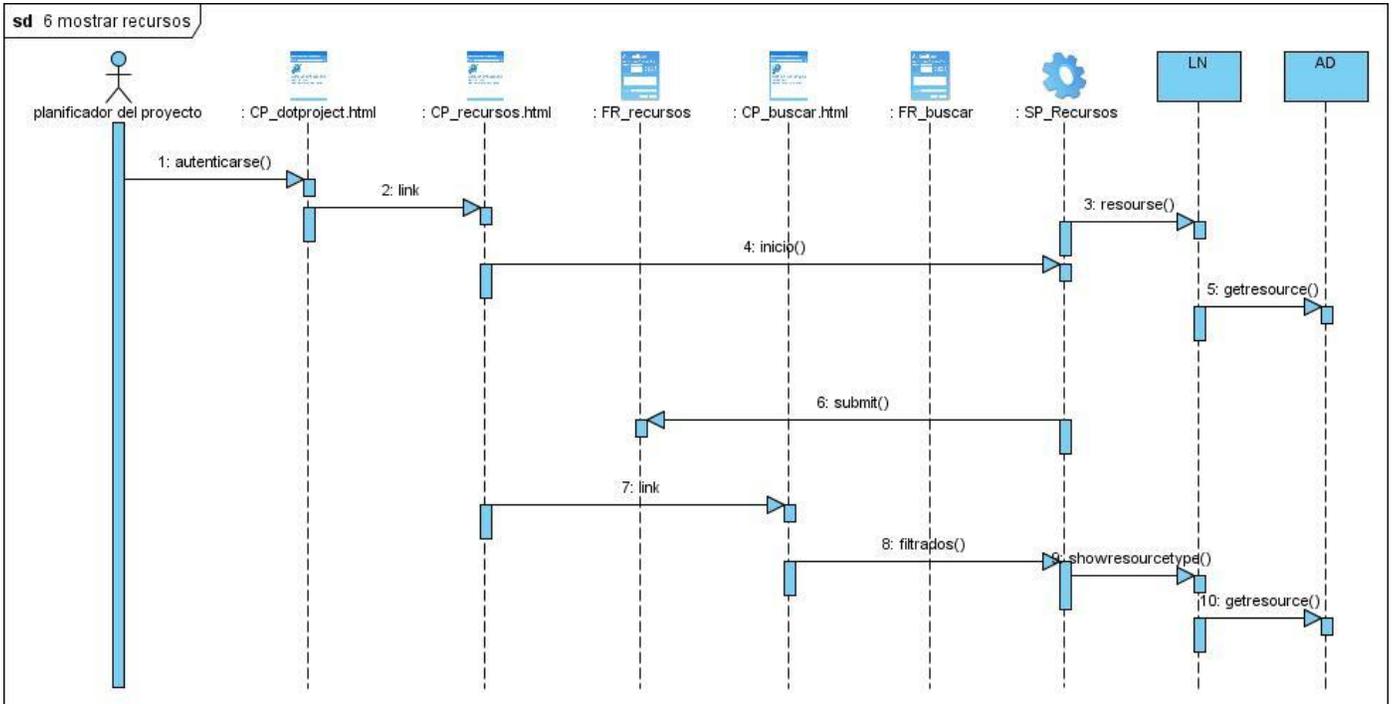


Ilustración 54 Diagrama de secuencia. Mostrar Recursos

## Prototipos funcionales

### HORARIO

✓	FECHA	USUARIO	PC	TAREA	COMIENZA	TERMINA	ACCIONES
<input type="checkbox"/>	2009-06-08	admin	ld02-317-01	implementacion	12:20:00	01:20:00	Modificar
<input type="checkbox"/>	2009-06-10	admin	ld02-317-01	diagrama de clases in	12:10:00	14:30:00	Modificar
<input type="checkbox"/>	2009-06-17	admin	ld02-317-01	analisis	15:00:00	16:00:00	Modificar

Registros Del 1 Al 3

### Modificar Horario

Fecha:

Usuario:

Pc:

Tarea:

Hora Inicio (00:00):

Hora Fin (00:00):

### Insertar de Horario por Día

# ANEXOS

dotProject 2.1.1 dotProject.net FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar horario de un día

Horarios | **Día** | Semana

Fecha:

Usuarios:

Pc:

Hora Inicio (00:00):

Hora Fin (00:00):

## Horario por semana

dotProject 2.1.1 dotProject.net FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar horario de la semana

Horarios | **Día** | **Semana**

Semana:

Usuario:

dotProject 2.1.1 dotProject.net FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar horario de la semana

Horarios | **Día** | **Semana**

**Días de la semana (06/06-13/06)**

Sunday (1)

Monday (2)

Tuesday (3)

Wednesday (4)

Thursday (5)

Friday (6)

Saturday (7)

Pc:

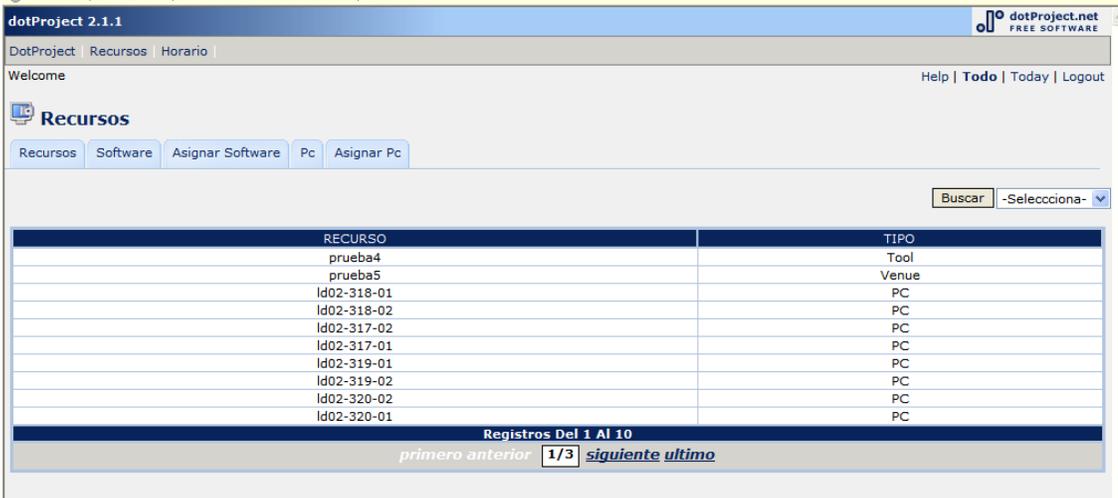
Tarea:

Hora Inicio (00:00):

Hora Fin (00:00):

## Modulo de recurso

# ANEXOS



dotProject 2.1.1

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Recursos

Recursos Software Asignar Software Pc Asignar Pc

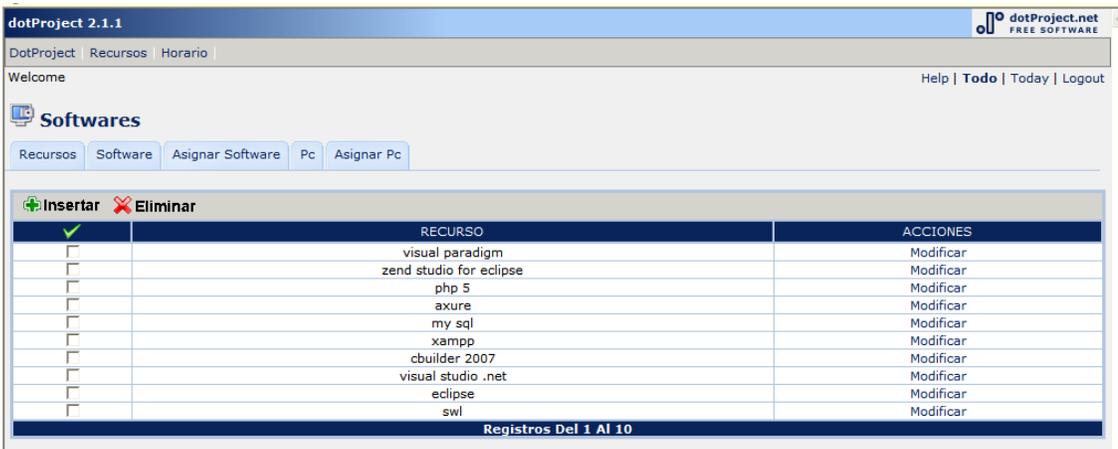
Buscar -Selecciona-

RECURSO	TIPO
prueba4	Tool
prueba5	Venue
ld02-318-01	PC
ld02-318-02	PC
ld02-317-02	PC
ld02-317-01	PC
ld02-319-01	PC
ld02-319-02	PC
ld02-320-02	PC
ld02-320-01	PC

Registros Del 1 Al 10

primero anterior 1/3 siguiente ultimo

## Gestionar Software



dotProject 2.1.1

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Softwares

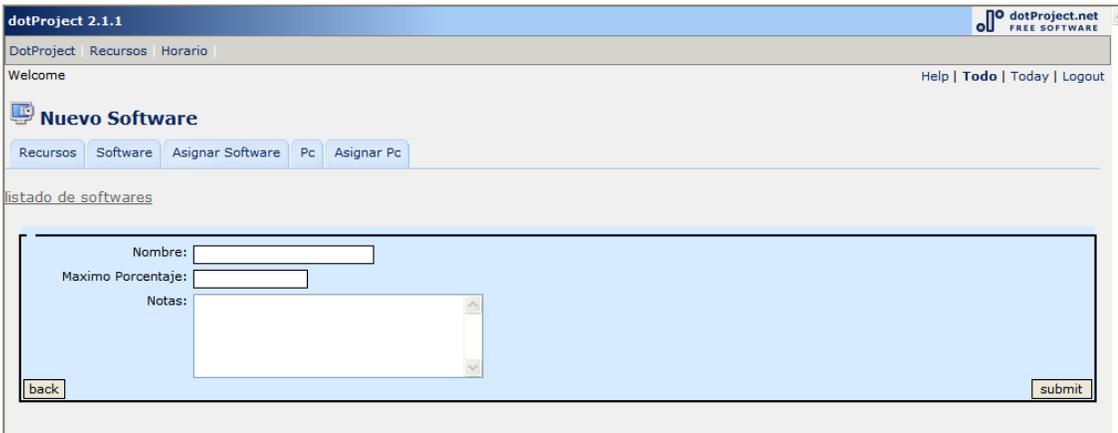
Recursos Software Asignar Software Pc Asignar Pc

Insertar  Eliminar

	RECURSO	ACCIONES
<input checked="" type="checkbox"/>	visual paradigm	Modificar
<input type="checkbox"/>	zend studio for eclipse	Modificar
<input type="checkbox"/>	php 5	Modificar
<input type="checkbox"/>	axure	Modificar
<input type="checkbox"/>	my sql	Modificar
<input type="checkbox"/>	xampp	Modificar
<input type="checkbox"/>	cbuilder 2007	Modificar
<input type="checkbox"/>	visual studio .net	Modificar
<input type="checkbox"/>	eclipse	Modificar
<input type="checkbox"/>	swi	Modificar

Registros Del 1 Al 10

## Insertar Software



dotProject 2.1.1

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Nuevo Software

Recursos Software Asignar Software Pc Asignar Pc

listado de softwares

Nombre:

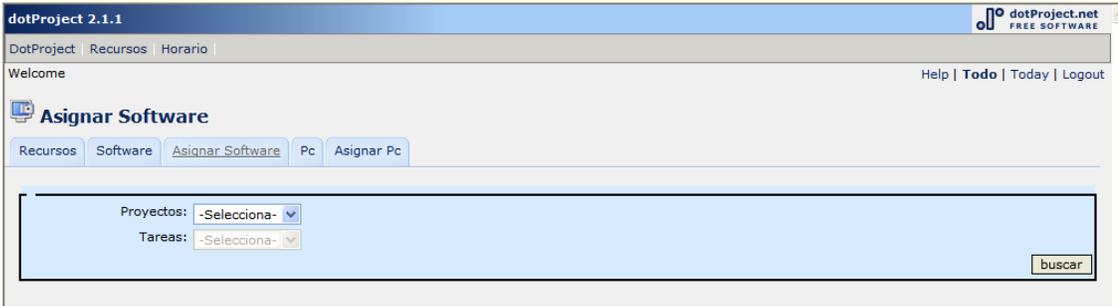
Maximo Porcentaje:

Notas:

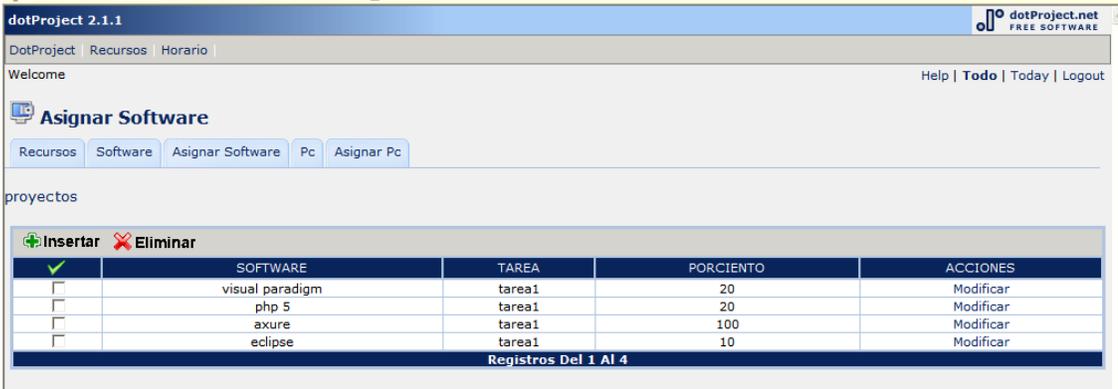
back submit

## Buscar proyectos según tarea

# ANEXOS



## Lista de software asignado a una tarea



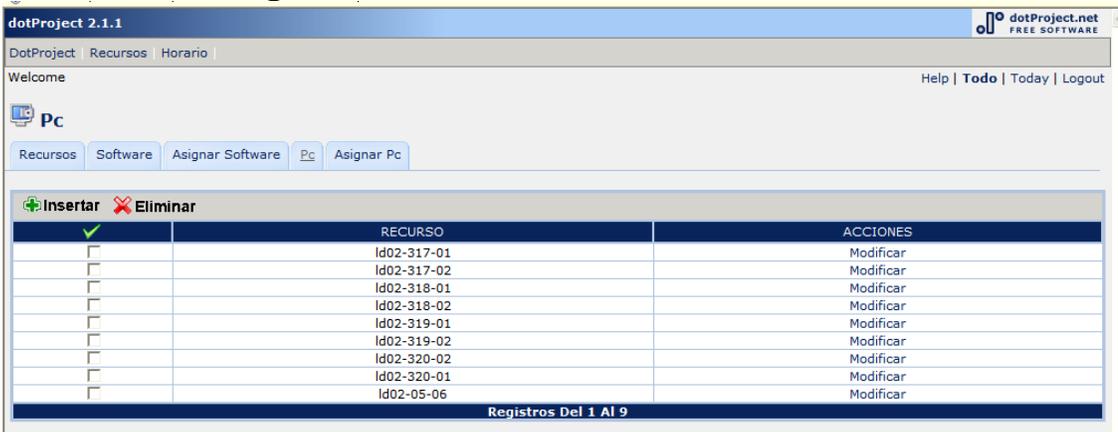
Insertar

## asignación de software



## Gestionar PC

### Lista de PC asignada



Insertar PC

# ANEXOS

dotProject 2.1.1 dotProject.net  
FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Nueva Pc

Recursos | Software | Asignar Software | Pc | Asignar Pc

listado de pc

Nombre:

IP:

Maximo Porcentaje:

Notas:

## Asignar PC a miembros

dotProject 2.1.1 dotProject.net  
FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar Pc

Recursos | Software | Asignar Software | Pc | Asignar Pc

Usuarios:

Pc:

## Lista de PC asignas de un usuario

dotProject 2.1.1 dotProject.net  
FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar Pc

Recursos | Software | Asignar Software | Pc | Asignar Pc

	PC	USUARIO	ACCIONES
<input type="checkbox"/>	ld02-317-01	Admin	Modificar

Registros Del 1 Al 1

## Insertar Asignación de PC

dotProject 2.1.1 dotProject.net  
FREE SOFTWARE

DotProject | Recursos | Horario

Welcome Help | **Todo** | Today | Logout

### Asignar nueva Pc

Recursos | Software | Asignar Software | Pc | Asignar Pc

Usuarios:

Pc: