

Instituto Superior Minero Metalúrgico de Moa

“Dr. Antonio Núñez Jiménez”



Desoft División Habana



**SISTEMA PARA LA GESTIÓN DE INFORMACIÓN DE
PROYECTOS DE IMPLEMENTACIÓN Y SOPORTE DE SOFTWARE
PARA LA EMPRESA DESOFT DIVISIÓN HABANA**

Trabajo de Diploma

En opción del título de Ingeniero Informático

Autor: Junier Garcia Nicot

Tutora: Ing. Yadira Arguelles Blanco

MOA, CUBA.

JULIO, 2010

Resumen

El trabajo de diploma constituye un proyecto novedoso para la Empresa Desoft División Habana al integrar soluciones informáticas: como el diseño Web, registro de información y gestión de Bases de Datos, todo desarrollado con herramientas que permiten ejecutarse sobre cualquier plataforma y con una orientación a objeto. Constituye una herramienta útil para la gestión de la información en el departamento de Implementación y Soporte de Software con el que se pretende facilitar el trabajo en el área, ya que la información actualmente se lleva aisladamente en diferentes formatos, lo cual dificulta el trabajo y el uso de un software minimizaría los errores y agilizaría la toma de decisiones.

Para el diseño del sistema, se utilizó el lenguaje UML y la metodología de desarrollo RUP, combinación que conlleva a que sea esta última la metodología más utilizada en la actualidad para el análisis y documentación de sistemas automatizados orientados a objetos. El sistema se instaló experimentalmente en el departamento de la empresa en cuestión. Se simularon situaciones críticas de funcionamiento y condiciones de error para valorar el comportamiento del sistema. Los resultados fueron satisfactorios.

Summary

The work of diploma constitutes an innovative project for the Desoft Division Havana Enterprise when integrating information-technology solutions: Like the design Web, Bases' record of information and steps of data, whole developed with tools that permit to run on any platform with an orientation and to object. He constitutes a useful tool for the steps of information at Implementation's department that the work in the area, since at present he is in for information separately in different formats intends to make easy itself with and Software's Support, which as he makes work difficult and the use of a software would minimize errors and would speed up the decision making.

For the systems design, language utilized UML and the methodology of development itself RUP, the ring that he bears to that he be this one ends the more methodology utilized at the present time for the analysis and documentation of automated systems guided to objects. System was installed experimentally at the company's in point department. They simulated criticisms of functioning and error conditions to appraise the behavior of system. Aftermath was satisfactory.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	3
1.1. GESTIÓN DE PROYECTO	4
1.1.1. Importancia de la gestión de proyecto	4
1.2. SISTEMAS AUTOMATIZADOS EXISTENTES	4
1.3. TENDENCIAS Y TECNOLOGÍAS ACTUALES	7
1.3.1. La Web	7
1.3.2. Servidor Web	8
1.3.2.1. Internet Information Services (IIS)	9
1.3.2.2. Zope	9
1.3.2.3. Apache	9
1.3.2.4. Selección del servidor Web: Apache	10
1.3.3. Lenguaje de programación	10
1.3.3.1. PERL	10
1.3.3.2. JSP	10
1.3.3.3. PHP	11
1.3.3.4. Selección del lenguaje de programación: PHP5	12
1.3.4. Selección del Framework de desarrollo: CodeIgniter	13
1.3.5. Selección del IDE de desarrollo: phpDesigner	14
1.3.6. Gestor de base de datos	15
1.3.6.1. MySql	15
1.3.6.2. Oracle	16
1.3.6.3. PostgreSQL	16
1.3.6.4. Selección del gestor de base de datos: PostgreSQL	18
1.3.7. Arquitectura cliente/servidor	19
1.3.8. Estilo arquitectónico	20
1.3.9. Metodología para el desarrollo de software	21
1.3.9.1. XP (Extreme Programming)	21
1.3.9.2. MSF (Microsoft Solution Framework)	22
1.3.9.3. RUP (Rational Unified Process)	22

1.3.9.4. Selección de la metodología para el desarrollo de software: RUP	24
1.3.10. Herramienta de modelación	24
1.3.10.1. Visual Paradigm for UML	24
1.3.10.2. Rational Rose	25
1.3.10.3. Selección de la herramienta de modelación: Rational Rose	26
1.3.11. Selección del lenguaje para el modelado del software: UML	26
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	28
2.1. FLUJO ACTUAL DE LOS PROCESOS VINCULADOS AL CAMPO DE ACCIÓN	29
2.2. OBJETO DE AUTOMATIZACIÓN	29
2.3. ACTORES DEL NEGOCIO	30
2.4. TRABAJADORES DEL NEGOCIO	30
2.5. DIAGRAMA DE CASO DE USO DEL NEGOCIO	30
2.6. DESCRIPCIÓN DEL CASO DE USO DEL NEGOCIO	31
2.7. DIAGRAMA DE ACTIVIDADES	32
2.8. MODELO DE OBJETO DEL NEGOCIO	32
2.9. REQUISITOS FUNCIONALES	33
2.10. REQUISITOS NO FUNCIONALES	35
2.11. ACTORES DEL SISTEMA.....	36
2.12. DIAGRAMA DE CASOS DE USO DEL SISTEMA	36
2.13. DESCRIPCIÓN DE LOS CASO DE USO DEL SISTEMA	37
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN.....	39
3.1. DISEÑO	40
3.2. DEFINICIÓN DE ELEMENTOS DE DISEÑO	40
3.3. DIAGRAMA DE CLASES DEL DISEÑO	40
3.4. DIAGRAMA DE SECUENCIAS	42
3.5. INTERFAZ DE USUARIO	45
3.6. MODELO FÍSICO DE DATOS	45
3.7. DIAGRAMA DE DESPLIEGUE.....	46
3.8. DIAGRAMA DE COMPONENTES.....	47
CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD	49

4.1 PLANIFICACIÓN POR PUNTO DE FUNCIÓN	50
4.1.1 Características del proyecto	53
4.2. BENEFICIOS TANGIBLES E INTANGIBLES	57
4.2.1. Beneficios tangibles	57
4.2.2. Beneficios intangibles	57
4.3. ANÁLISIS DE COSTOS Y BENEFICIOS	57
CONCLUSIONES.....	59
RECOMENDACIONES.....	60
REFERENCIAS BIBLIOGRÁFICAS.....	61
GLOSARIO DE TÉRMINOS.....	63
ANEXO 1: DESCRIPCIÓN DE LOS CASOS DE USO DEL SISTEMA.....	65
ANEXO 2: DIAGRAMAS DE CLASES DEL DISEÑO	70
ANEXO 3: DESCRIPCIÓN TEXTUAL DE LAS CLASES PERSISTENTES.....	79
ANEXO 4: DIAGRAMAS DE SECUENCIA.....	83
ANEXO 5: DIAGRAMAS DE COMPONENTES	98

ÍNDICE DE TABLAS

Tabla 2.1: Actores del negocio	30
Tabla 2.2: Trabajadores del negocio.....	30
Tabla 2.3: Descripción del caso de uso del negocio «generar información»	32
Tabla 2.4: Descripción de los actores del sistema.	36
Tabla 2.5: Descripción textual del caso de uso «gestionar cliente»	37
Tabla 2.6: Descripción textual del caso de uso «gestionar proyecto»	38
Tabla 3.3. Terminología de los diagramas de clases de diseño.	41
Tabla 4.1: Entradas externas (EI)	51
Tabla 4.2: Salidas externas (EO)	52
Tabla 4.3: Consultas externas (EQ).....	52
Tabla 4.4: Ficheros lógicos internos (ILF)	53
Tabla 4.5: Puntos de función desajustados (UFP)	53
Tabla 4.5: Factor de escala	54
Tabla 4.6: Multiplicadores de esfuerzo (EM).....	54
Tabla 4.7: Cantidad de instrucciones fuentes	54
Tabla 2.7: Descripción textual del caso de uso «gestionar organismo»	65
Tabla 2.8: Descripción textual del caso de uso «gestionar empresa».....	65
Tabla 2.9: Descripción textual del caso de uso «gestionar contacto»	66
Tabla 2.10: Descripción textual del caso de uso «gestionar producto».....	66
Tabla 2.11: Descripción textual del caso de uso «gestionar implementador».....	67
Tabla 2.12: Descripción textual del caso de uso «gestionar avance de proyecto»	67
Tabla 2.13: Descripción textual del caso de uso «gestionar usuario»	68
Tabla 2.14: Descripción textual del caso de uso «gestionar municipio».....	68
Tabla 2.15: Descripción textual del caso de uso «descargar reporte de implantación»	69
Tabla 2.16: Descripción textual del caso de uso «descargar reporte de seguimiento»	69

ÍNDICE DE FIGURAS

Figura 1.1: Modelo tradicional para aplicaciones Web (izquierda) comparado con el modelo AJAX (derecha)	8
Figura 1.2: Funcionamiento de las aplicaciones Web	12
Figura 1.3: Modelo cliente/servidor distribuido en 3 capas	19
Figura 1.4: Sistema cliente/servidor	20
Figura 2.1: Diagrama de caso de uso del negocio.	31
Figura 2.2: Diagrama de actividad «generar información».....	32
Figura 2.3: Modelo de objeto de negocio	33
Figura 2.4: Diagrama del caso de uso del sistema.....	37
Figura 3.1. Diagrama de clases del diseño «autenticar».....	42
Figura 3.2. Diagrama de secuencia del caso de uso «gestionar cliente»	43
Figura 3.3. Diagrama de secuencia del caso de uso «gestionar proyecto»	45
Figura 3.4. Interfaz de usuario.....	45
Figura 3.5. Modelo de datos.....	46
Figura 3.6. Diagrama de despliegue	47
Figura 3.7. Diagrama de componentes «general»	48
Figura 3.8. Diagrama de componentes «gestionar usuarios»	48
Figura 3.9. Diagrama de clases del diseño «Gestionar Organismo»	70
Figura 3.10. Diagrama de clases del diseño «Gestionar Empresa»	71
Figura 3.11. Diagrama de clases del diseño «Gestionar Cliente»	72
Figura 3.12. Diagrama de clases del diseño «Gestionar Contacto».....	72
Figura 3.13. Diagrama de clases del diseño «Gestionar Producto».....	73
Figura 3.14. Diagrama de clases del diseño «Gestionar Implementador».....	74
Figura 3.15. Diagrama de clases del diseño «Gestionar Proyecto»	75
Figura 3.16. Diagrama de clases del diseño «Gestionar Avance del Proyecto».....	76
Figura 3.17. Diagrama de clases del diseño «Gestionar Usuario»	77
Figura 3.18. Diagrama de clases del diseño «Gestionar Municipio»	78
Figura 3.19. Diagrama de secuencia «gestionar organismo»	84
Figura 3.20. Diagrama de secuencia «gestionar empresa».....	85
Figura 3.21. Diagrama de secuencia «gestionar cliente»	87
Figura 3.22. Diagrama de secuencia «gestionar contacto».....	88
Figura 3.23. Diagrama de secuencia «gestionar producto».....	90
Figura 3.24. Diagrama de secuencia «gestionar implementador»	91
Figura 3.25. Diagrama de secuencia «gestionar proyecto»	93

Figura 3.26. Diagrama de secuencia «gestionar avance de proyecto»	94
Figura 3.27. Diagrama de secuencia «gestionar usuario»	96
Figura 3.28. Diagrama de secuencia «gestionar municipio»	97
Figura 3.29. Diagrama de componentes «gestionar proyecto».....	98
Figura 3.30. Diagrama de componentes «gestionar organismo».....	98
Figura 3.31. Diagrama de componentes «gestionar empresa»	99
Figura 3.32. Diagrama de componentes «gestionar cliente»	99
Figura 3.33. Diagrama de componentes «gestionar contacto»	100
Figura 3.34. Diagrama de componentes «gestionar producto»	100
Figura 3.35. Diagrama de componentes «gestionar implementador»	101
Figura 3.36. Diagrama de componentes «gestionar municipio»	101
Figura 3.37. Diagrama de componentes «descargar reporte de implementación»	102
Figura 3.38. Diagrama de componentes «descargar reporte de seguimiento»	102

Introducción

Con el actual desarrollo de las redes de computadoras, de la programación y de las bases de datos, el mundo ha visto nacer una nueva era de sistemas informáticos denominados Sistemas de Gestión, los cuales han surgido para manipular y procesar, de una manera eficiente y segura, los flujos de información en todos los campos del conocimiento.

El buen manejo de la información puede significar la diferencia entre el éxito o el fracaso para todos los proyectos que se emprendan dentro de un organismo. Es por ello que para las empresas de hoy en día la gestión rápida y con calidad de la información, se ha convertido en una necesidad imperante, fundamentalmente para aquellas que ofrecen servicios, como es el caso de Desoft División Habana.

Desoft División Habana es una empresa cubana de desarrollo de software, que tiene como objeto social producir y comercializar software y aplicaciones informáticas de todo tipo, representar y distribuir en el país productos y soluciones informáticas nacionales o extranjeras y brindar algunos servicios como instalación, mantenimiento, garantía y postventa relacionados con sus producciones.

Dicha empresa está constituida por 14 divisiones provinciales, que se subordinan a la Casa Matriz de Desoft División Habana. Cada división provincial está conformada por diferentes subgerencias entre las que se encuentran: Recursos Humanos, Economía, Desarrollo e Implementación y Soporte.

Desoft División Habana tiene como misión crear una red municipal de servicios y desarrollo de soluciones integrales en provincia Habana. El área de Implementación y Soporte (I + S) de esta división se encarga específicamente de implementar y dar soporte a los productos de la cartera de Desoft División Habana, en las distintas entidades de la provincia que soliciten dichos servicios. Cuando un cliente solicita un servicio, se le asigna uno o varios implementadores, para atender a dicho cliente por un período de tiempo determinado, y a esto se le denomina proyecto.

En la actualidad los proyectos de esta área se registran en una hoja de cálculo generada por la suite ofimática Microsoft 2007, en el cual se recogen los datos de todos los proyectos, con sus clientes y los implementadores que están a cargo del mismo. De esta forma se dificulta el eficiente control de los proyectos, pues no pueden registrarse con efectividad el estado de los mismos, y además los implementadores se encuentran dispersos en los diferentes municipios. No se pueden generar de forma rápida los informes necesarios para realizar el análisis sistemático de los proyectos, así como del trabajo de los implementadores, lo cual es fundamental para la evaluación del desempeño de estos.

Los encargados de controlar los proyectos tienen que realizar grandes esfuerzos para mantener actualizada la información y entregar los resultados de su trabajo en tiempo. La empresa Desoft División Habana no

cuenta con una herramienta capaz de realizar una gestión de información de proyectos de forma eficaz. En la actualidad el área manipula mucha información diariamente de distintos proyectos a la vez, esto implica que con frecuencia la gestión no resulte correcta lo que se traduce en pérdida de tiempo y recursos, la planificación de los proyectos se altera y esto no sólo afecta a un cliente sino a varios porque en ocasiones la solución a estos problemas es detener un proyecto para terminar otro.

Dada la **situación problémica** anterior, se identificó el siguiente **problema científico**: ¿Cómo garantizar la gestión de información de proyectos de implementación y soporte de software en Desoft División Habana logrando un eficiente flujo informativo que permita organizar, controlar y centralizar con rapidez toda la información?

Dicho problema está enmarcado dentro del **objeto de estudio**: El proceso de gestión de información de proyectos, centrándose **el campo de acción** en: El proceso de gestión de información de proyectos del área de implementación y soporte de software en Desoft División Habana. Ante la necesidad de dar solución al problema antes expuesto, se plantea como **objetivo general**: Desarrollar un sistema que facilite la gestión de información de proyectos del área de implementación y soporte de software en Desoft División Habana.

Del objetivo general planteado se desglosan los siguientes **objetivos específicos**:

- Realizar una investigación sobre las herramientas similares existentes.
- Realizar la modelación del negocio.
- Elaborar el levantamiento de requisitos.
- Confeccionar el modelo de casos de usos del sistema a implementar.
- Realizar el diseño del sistema.
- Implementar el software.
- Realizar el estudio de factibilidad.

Tomando en consideración todo lo analizado hasta el momento se defiende la siguiente **idea**: Con el desarrollo de un sistema para la gestión de información de proyectos del área de implementación y soporte de software en Desoft División Habana se pretende garantizar la rapidez en el trabajo del área, acceso a la información de manera efectiva y facilidades para la obtención de reportes.

CAPÍTULO

1

Fundamentación Teórica

1.1. Gestión de proyecto

La gestión de proyectos es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos. Un proyecto es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único. [1]

1.1.1. Importancia de la gestión de proyecto

La administración de proyectos es usada en una gran diversidad de campos, como, por ejemplo, en bancos, desarrollo de sistemas, lanzamientos de productos, proyectos especiales, en la industria petroquímica, en telecomunicaciones, en defensa nacional, y en muchos otros ámbitos e industrias. Los cambios tecnológicos, la necesidad de introducir nuevos productos al mercado, las cambiantes exigencias de los consumidores de productos, entre otras cosas, incrementan el flujo de operaciones en una organización, esto hace que los métodos de administración convencionales sean inadecuados. Por esta razón, la administración de proyectos es importante, ya que ofrece nuevas alternativas de organización, control y seguimiento a las empresas. Asimismo, la administración de una cartera de proyectos sirve para aprovechar de mejor manera los recursos críticos cuando están limitados en cantidad y/o tiempo de disponibilidad.

Toda organización tiene proyectos Las organizaciones están divididas en muchas áreas y necesitan administrar trabajos similares. [2]

¿Por qué se requiere gestión de proyectos?

Para administrar eficientemente: los esfuerzos, el tiempo, controlar los bienes o servicios producidos y los costos y recursos asociados a todo esto.

1.2. Sistemas automatizados existentes

Existen en la actualidad y desde hace algún tiempo, disímiles sistemas automatizados, pues la gestión de proyectos es un proceso que se hace necesario en cualquier empresa del mundo. La investigación solo incluye empresas que a nivel mundial presentan resultados satisfactorios en la automatización de esta actividad.

EML: Aplicación analítica desarrollada por NORADM, Noruega para la gestión de proyectos. Asegura que se plantean las preguntas fundamentales y se analizan las debilidades, brindando a los que toman decisiones una información mejor y más pertinente. [3]

Ventajas

- Guía el análisis sistemático y lógico de los elementos claves interrelacionados que constituyen un proyecto bien diseñado.
- Mejora la planificación al resaltar los lazos que existen entre los elementos del proyecto y los factores externos.
- La administración y la dirección se benefician de procedimientos normalizados.
- Facilita el entendimiento común y una mejor comunicación entre los que toman decisiones.

Desventajas

- Puede surgir una rigidez en la dirección del proyecto cuando se absolutizan los objetivos y los factores externos especificados al comienzo.
- Es políticamente neutral en cuanto a cuestiones de distribución del ingreso, oportunidades de empleo y acceso a recursos. Por lo tanto es solamente una herramienta entre muchas que se pueden usar.

MyWorkPlan: Herramienta Web desarrollada por SESCOI, España para las empresas de software, está pensado para ser fácil de aprender y manejar. En colaboración con sus clientes se ha analizado los métodos de trabajo y las mejores prácticas utilizadas para la gestión de proyectos. La aplicación cuenta con una amplia experiencia en las soluciones de gestión. La empresa ha desarrollado una metodología (FastStart) muy ágil y propia para la gestión de proyectos, que permite a las empresas agilizar sus actividades además de hacerlas con calidad. [4]

Ventajas

- La aplicación permite a los clientes controlar las variables claves y los proyectos con más eficacia.
- Es una nueva generación de sistemas ERP (Enterprise Resource Planning) para empresas o fabricantes por encargo en base a proyectos en distintos sectores.
- Es un sistema inteligente capaz de cubrir toda la gestión de proyectos.
- Las funciones ERP permiten a las empresas automatizar y controlar todas las actividades de la gestión y calidad.

Desventajas

- Su gran problema es que requiere de banda ancha.
- Es infalible, puede fallar.

MAERA: Esta aplicación fue desarrollada en el área de América Latina y el Caribe para promover información del estado del proyecto, la cual puede ser usada desde cualquier parte del mundo contado con una conexión de red. Cuenta con un dispositivo de seguridad que sus administradores están en disposición de otorgar permisos de seguridad estructurados para asociados seleccionados, permitiéndole acceso solo a la información relacionada con su(s) proyecto(s). Esta aplicación solo se ofrece a los proyectos sin costos. [5]

La base de datos de MAERA se actualiza diariamente – excepto los domingos – con una copia de la base de datos de Atlas. Debido a las distintas diferencias horarias existentes entre los países de América Latina y el Caribe más el lugar donde se encuentran ubicados los servidores de Atlas (Phoenix, Arisona, USA) en condiciones normales la información tendrá un retraso entre 24 y 36 horas respecto a Atlas

Ventajas

- Brinda un reporte del estado del proyecto, el cual es configurable según la necesidad de su jefe de equipo y se puede descargar en una hoja de cálculo generada por la suite ofimática Microsoft 2007.
- Los derechos de usuario son administrados por la oficina del PNUD en cada país. Estos derechos constituyen el conjunto de funcionalidades de la aplicación a las que un usuario tiene acceso.
- Los derechos de acceso de cada usuario constituye su “perfil”. El idioma es un atributo del perfil de cada usuario y puede seleccionarse entre Español, Inglés y Francés.

Desventajas

- Para nadie es secreto las relaciones USA – Cuba, por tanto el solo hecho de que los servidores de respaldo de MAERA se encuentre en ese país ya representa un problema para el nuestro.
- El problema de la conexión de red de nuestro país no siempre es la adecuada para estos tipos de trabajo donde se necesita de una pronta respuesta de la aplicación y un confiable resultado.

- Incremento de los costos de la empresa por tantas horas de conexión a la red.

1.3. Tendencias y tecnologías actuales

Hoy día es de gran importancia que las empresas y organizaciones alcancen sus objetivos empresariales adquiriendo los resultados deseados y brindando su aporte a la sociedad. Los sistemas informáticos son herramientas de gran valor a través de las cuales las empresas pueden obtener grandes beneficios, como lo pueden ser una eficiente comunicación e interrelación entre todo el personal de la misma y la reducción de costos y tiempos de producción tanto para bienes como para servicios. La herramienta seleccionada para el desarrollo de un producto software es un elemento sumamente importante, pues brinda a los desarrolladores la posibilidad de crear productos de alta calidad. Puesto que es importante la tarea de seleccionar las metodologías y tecnologías que se ajusten para desarrollar la presente propuesta, se ha realizado una investigación acerca de las tendencias que existen y las tecnologías que son utilizadas actualmente en el desarrollo de aplicaciones Web.

1.3.1. La Web

Se puede considerar como una plataforma o “sistema operativo” en el cual los recursos están distribuidos en la red y están siendo extendidos en todo momento con posibilidades ilimitadas.

La Web se ha hecho popular con aplicaciones tales como clientes de correo, buscadores, portales, foros, chats y otros servicios. Además de estas aplicaciones de propósito general existe adicionalmente una gran diversidad de soluciones que se acomodan al ambiente Web como son: Administradores de contenido (CMS), Administrador de proyectos, Suites para trabajo colaborativo, Administración de relaciones con el cliente (CRM).

La Web se modifica día a día, lo que ayer parecía imposible hoy es una realidad. Si bien es cierto que la arquitectura cliente/servidor de la Web ha ofrecido muchas ventajas también es cierto que carece de la riqueza gráfica de las aplicaciones de escritorio que cuentan con controles inteligentes que dan mayor fluidez al trabajo del usuario, esto ha sido resuelto con varias estrategias o tecnologías tales como AJAX (Asynchronous JavaScript And XML), flash y Web 2 entre otras. Así que en vez de ir perdiendo fuerza debido a la pobreza en sus interfaces gráficas, la Web busca alternativas que le permitan ofrecer todas sus ventajas pero con la posibilidad de ofrecer controles visuales más amigables al trabajo del usuario.

XAJAS: Es una clase realizada con PHP que nos permite utilizar AJAX, combinado con PHP, para la

creación de aplicaciones interactivas, de una manera muy simplificada. Con XAJAX podemos fácilmente ejecutar funciones PHP, que se ejecutan en el servidor, cuando el usuario realiza acciones en la página. Luego, los resultados de esas funciones PHP se producen en la misma página, sin que se tenga que recargarse. XAJAX es un producto Open Source (Código Abierto) gratuito y compatible con los navegadores más comunes, como Firefox, Internet Explorer, Opera, etc. [6]

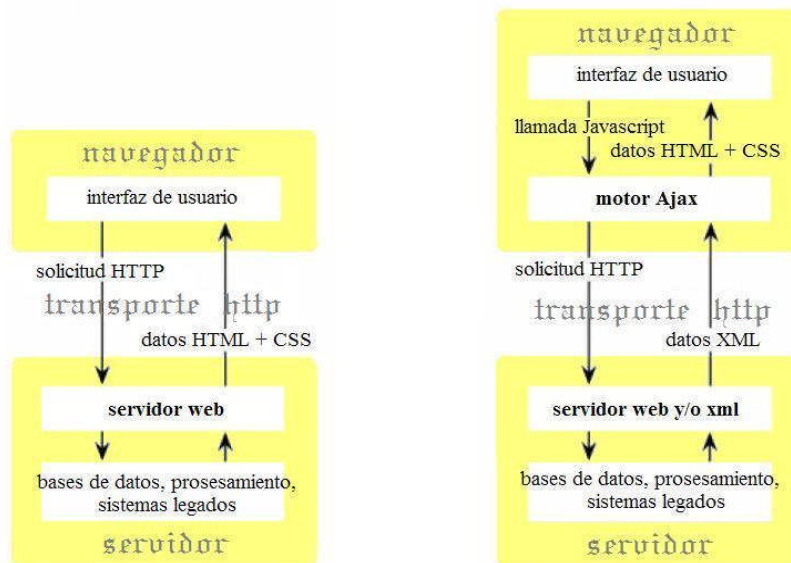


Figura 1.1: Modelo tradicional para aplicaciones Web (izquierda) comparado con el modelo AJAX (derecha)

No se puede despreciar el enorme impacto que ha tenido el computador personal o PC en la actualidad, el haber puesto al servicio de un usuario normal el poder de una computadora en vez de un terminal (NetWork) simple ha potenciado una gran diversidad de usos. Las aplicaciones de escritorio se han usado y se seguirán usando y tienen un campo enorme (suite de oficina, aplicaciones gráficas, juegos, utilidades o el mismo sistema operativo), no todo está en la Web, hay cosas que se necesita que se ejecuten estrictamente en su máquina para aprovechar el poder que tiene al alcance. Pero la fusión e integración de servicios de los computadores, las aplicaciones de escritorio y la extensión de las facultades de comunicación con las aplicaciones Web que hace posible la Internet es la plataforma óptima que sirve de infraestructura para todos los usuarios desde el tipo empresarial, institucional o personal.

1.3.2. Servidor Web

Un servidor web es el encargado de construir y devolver las páginas web que solicitan los navegadores.

1.3.2.1. Internet Information Services (IIS)

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS. Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web. [7]

1.3.2.2. Zope

Es un servidor de aplicaciones web de código abierto escrito en el lenguaje de programación Python. Puede ser manejado casi totalmente usando una interfaz de usuario basada en páginas Web. Un sitio web de Zope está compuesto de objetos en lugar de archivos, como es usual con la mayoría de los otros sistemas de servidores web. Las ventajas de usar objetos en lugar de archivos son:

- Combinan el comportamiento y los datos en una forma más natural que los archivos de texto plano.
- Alientan el uso de componentes estándares que se ocupan de una parte particular de las que forman una aplicación Web, permitiendo flexibilidad y buena descomposición.
- Posibilitan procesos automáticos de gestión de información.

Lo más característico de Zope es su base de datos orientada a objetos, llamada ZODB o Zope Object Database. Esta base de datos almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación es muy diferente de las base de datos relacionales habituales. Sin embargo, Zope dispone de múltiples conectores para las diferentes bases de datos relacionales y ofrece sistemas básicos de conexión y consulta abstrayéndolos como objetos. [8]

1.3.2.3. Apache

Es un servidor Web libre. Su nombre procede de "patchy server", por ser una versión *parcheada* en 1995 de uno de los primeros servidores Web, el NCSA HTTPD, y actualmente corre en muy diversas plataformas

(Unix, GNU/Linux. Windows, etc.). Es desarrollado y mantenido por la comunidad del software libre a través de la [Fundación Apache]. Es la auténtica "kill app" del software libre en el ámbito de los servidores y el ejemplo de software libre más exitoso (por delante incluso del kernel Linux): desde hace años, más del 70% de los servidores Web de Internet corren este magnífico software, estadísticas según Netcraft. Considerado como uno de los mejores servidores en cuanto a eficiencia, funcionalidad y velocidad. Contar con Apache es contar con un servidor modular, extensible, popular y totalmente gratis. [9]

1.3.2.4. Selección del servidor Web: Apache

Analizando las potencialidades de cada servidor expuesto en este documento comprendimos, dadas sus características que la mejor opción para nuestra aplicación era el servidor Web Apache. Este servidor logra una magnífica integración con el lenguaje PHP, ya que las funciones y clases más importante de este lenguaje están predefinidas para este servidor. La característica principal de este servidor es su código libre y actualizado, así podremos crear una aplicación fiable y robusta.

1.3.3. Lenguaje de programación

1.3.3.1. PERL

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje libre de uso. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows y Linux. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP. [10]

1.3.3.2. JSP

Es un acrónimo de Java Server Pages, que traducido es algo así como Páginas de Servidor Java. Es una tecnología orientada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con

etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual. [11]

1.3.3.3. PHP

Que debe sus siglas, originalmente, a “Personal Home Page” y en la actualidad a “Hypertext Pre-Processor”, es un lenguaje gratuito, rápido, multiplataforma y compatible con los gestores de bases de datos más comúnmente empleados, que ha devenido en uno de los lenguajes de programación Web más usados en Internet.

La comunidad de desarrolladores que colaboran en las mejoras a este lenguaje ofrecen una amplia librería de funciones que cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, y cobertura documental a las nuevas implementaciones. Es innegable como PHP ha incrementado sus adeptos desde su surgimiento hasta la actualidad.

El número creciente de desarrolladores que han hallado en este lenguaje la mejor variante para las implementaciones Web constituye una amenaza para otros lenguajes y tecnologías propietarias que se sentían muy seguras en el mercado de las aplicaciones Web. El uso de PHP ha proliferado muy rápidamente en los últimos años. Hasta abril del 2010 PHP es usado por 20 016 421 sitios. [12]

¿Cómo Funciona?

A diferencia de Java que se ejecuta en el navegador, PHP se ejecuta en el servidor por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HyperText Markup Language o Lenguaje de Etiquetas de Hipertexto (HTML). Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo, para que sus páginas PHP funcionen el servidor donde están alojadas debe soportar PHP

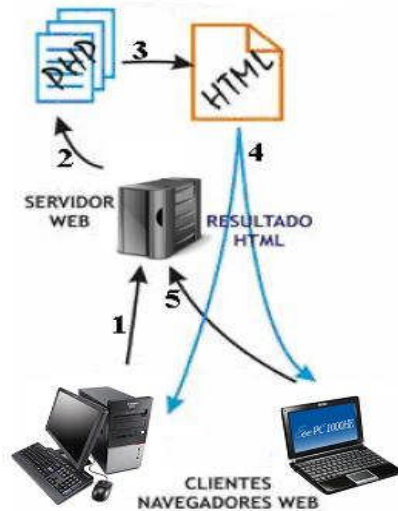


Figura 1.2: Funcionamiento de las aplicaciones Web

Ventajas

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los gestores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre y totalmente gratis, por lo que se presenta como una alternativa para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear los formularios para la Web.
- Biblioteca nativa de funciones sumamente amplia.

1.3.3.4. Selección del lenguaje de programación: PHP5

PHP (Hypertext Pre-Processor) es un lenguaje interpretado de propósito general muy usado, diseñado especialmente para el desarrollo de sistemas Web el cual puede ser embebido dentro de código HTML.

Generalmente se ejecuta en un servidor Web, donde es tomado el código como entrada, creándose posteriormente como salidas, páginas Web. Puede ser puesto en marcha por la mayoría de los servidores Web así como en la casi totalidad de sistemas operativos y plataformas sin ningún tipo de costo.

PHP es un lenguaje de fácil uso y comparte muchas similitudes con varios de los lenguajes de programación más comunes, esto propicia una mejor asimilación ya que la curva de aprendizaje es más suave y a los desarrolladores se le hace más sencillo identificarse con este lenguaje. Puede ser conectado a un gran número de Gestores de Bases de Datos como MySQL, PostgreSQL, Oracle, DB2, Microsoft SQL Server, Firebird y SQLite. El hecho de que tenga una alta integración con PostgreSQL lo hace un candidato potente para su selección.

1.3.4. Selección del Framework de desarrollo: CodeIgniter

Un framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework web, se puede definir como un conjunto de componentes que forman un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. [13]

Existe muchos frameworks para PHP, en su mayoría inspirados en el patrón MVC (Modelo Vista Controlador), pero ninguno es tan ligero y flexible como CodeIgniter. Lo mejor de todo es que CodeIgniter puede ser tan sólo VC (Vista Controlador), es decir no obliga al usuario a utilizar una base de datos para un desarrollo. Se considera que puede ser liviano, bastante fácil de instalar y sencillo de configurar. Está pensado para brindar un alto rendimiento con una óptima configuración. Cumple con el fin de cualquier framework, una estructura bien precisa que da soporte a un proyecto web y que ayuda a que este nuevo proyecto sea organizado y desarrollado. La documentación que ofrece es excelente. El manual y los tutoriales son ejemplos de cómo se debe documentar y enseñar un framework en la actualidad. Entre las características del CodeIgniter se puede mencionar:

- Es libre: se encuentra bajo una licencia open source.
- Se ejecuta sobre PHP4 y PHP5.

- Es muy rápido.
- Usa el patrón MVC.
- Genera URL's limpias y amigables.
- Posee paquetes de librerías.
- Es extensible: fácilmente extendido a través del uso de plugins y librerías asistentes.
- Está ampliamente documentado.

Razones por las cuales elegir CodeIgniter:

- Documentación. Tiene una muy buena documentación en Internet, además de una comunidad de usuarios muy activos.
- MVC. El uso de CodeIgniter se basa en el patrón Modelo Vista Controlador, separando la vista, del negocio y del controlador.
- Facilidad de uso. La curva de aprendizaje es sumamente corta. Perfectamente se puede comenzar a utilizar las características habiendo leído el tutorial de CodeIgniter o las guías en Internet.
- Active Record. El manejo y abstracción de base de datos es realmente bueno implementando este patrón.
- Estructura organizada. La estructura de archivos es muy clara y ofrece gran flexibilidad en caso que necesite cambiarse por algún motivo. La división en modelo, vista, controlador, librerías, ayuda, configuración, etc. es muy práctica y comprensible.
- Ruteo. Hoy en día la optimización para buscadores es fundamental. Los mecanismos de ruteo de CodeIgniter favorecen a este y otras aplicaciones.

CodeIgniter es recomendable para desarrollar una aplicación web rápidamente, y que necesite un framework liviano y rápido, porque de lejos es el más sencillo, mejor documento y altamente flexible.

1.3.5. Selección del IDE de desarrollo: phpDesigner

Este IDE de desarrollo brinda un gran número de funcionalidades a la hora de programar cualquier aplicación web, cuenta con soporte de sintaxis no solo para PHP en nuestro caso, sino para la gran mayoría de los lenguajes de desarrollo web, así como el hecho de ser fácil de manejar, aún para personas con poca experiencia en programación web por su forma tan amigable y su gran poder expresivo para el desarrollador.

Algunas características que podemos mencionar:

Destacado de código: Soporta los lenguajes típicos que se utilizan a la hora de crear aplicaciones web: HTML, CSS, Javascript, XML, además de PHP, A esta opción le llaman "Intelligent Highlightting" y en las opciones de personalización del programa se puede desactivar la funcionalidad si no nos gusta.

Completamiento de código con ayudas: Algunas como cierre automático de paréntesis, comillas y otras. Pero lo que realmente ayuda son los mensajes contextuales a la hora de escribir llamadas a funciones, consistentes en pequeños textos flotantes con declaraciones de las funciones que estás utilizando, sus parámetros, una breve descripción de la funcionalidad y en las versiones de PHP donde está disponible. Las ayudas de completado de código incluyen todas las funciones de PHP y de las librerías que estés utilizando en un proyecto o hayas incluido en un script en particular.

Administración de proyectos: Con múltiples opciones de configuración.

Debug de código PHP: Por medio de Xdebug se puede probar el debug en cualquier fichero de código.

Conexión con servidores FTP: Muy funcional para editar un archivo que está en un servidor remoto. Además, los proyectos también se pueden asociar con un espacio de hosting por medio de FTP.

Búsqueda avanzada de textos: Así como funciones de reemplazo de textos. Códigos y plantillas ya listos para simplificar ciertas tareas repetitivas.

Integración con la documentación de PHP

Todas estas características lo hace un producto muy apetecible para desarrolladores en PHP. [14]

1.3.6. Gestor de base de datos

Debido a la necesidad de mantener la información almacenada y poder manipularla cuando sea el caso, es que se debe utilizar un Sistemas de Gestión de Bases de Datos (SGBD). Estos sistemas cuentan con dos lenguajes fundamentales para la definición y la manipulación de los datos, uno es el Data Definition Language (DDL), o Lenguaje de Definición de Datos, que cómo bien dice su nombre constituye una herramienta para definir los datos que serán almacenados, con precisión; el otro es el Data Management Language (DML) o Lenguaje de Manipulación de Datos que se encarga de facilitar el acceso y la manipulación de los datos. Existe gran cantidad de SGBD.

1.3.6.1. MySql

Es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto. Su principal objetivo de diseño fue la velocidad. Otra característica importante es que consume muy pocos recursos, tanto de CPU como de memoria. Tiene licencia GPL a partir de la versión 3.23.19. Algunas de sus ventajas respecto a otros gestores de base de datos son:

Alto rendimiento. Buena velocidad al procesar información y demás. Aunque se cuelgue, no suele perder información ni corromper los datos. Muy buena integración con PHP. No tiene límites en el tamaño de los registros. Control de acceso, mediante permisos especiales para cada usuario y restricción por tablas. [15]

1.3.6.2. Oracle

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Pero de código cerrado y con un alto costo por licencia. Además ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos del 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre del 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia del Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux. [16]

1.3.6.3. PostgreSQL

Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia Berkeley Software Distribution (BSD). Está considerado como el sistema de bases de datos de código abierto más avanzada del mundo, proporciona un gran número de características que normalmente sólo se

encontraban en los sistemas de bases de datos comerciales. Intenta ser un sistema de bases de datos de mayor nivel que MySQL, a la altura de Oracle, Sybase o Interbase. Como muchos otros proyectos OpenSource, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones las cuales trabajan en su desarrollo. [17]

Características

- Alta concurrencia: PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.
- Amplia variedad de tipos nativos.
- Claves ajenas también denominadas Llaves Ajenas o Llaves Foráneas (Foreign Keys).
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.
- Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- Tiene soporte para disparadores (*triggers*) y procedimientos en el servidor.
- Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un costo añadido en el rendimiento).
- Es de 2 a 3 veces más lenta que MySQL.
- Tiene menos funciones en PHP que MySQL.

Ventajas

- Soporta diferentes tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios e incorpora una estructura de datos array.
- Cuenta con distintos tipos de funciones como: geométricas, orientadas a operaciones con redes, para el manejo de fechas, etc.
- Permite la declaración de funciones propias y la definición de disparadores (*triggers*).
- Incluye el uso de índices, reglas, vistas y claves ajenas.

- Permite herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que este gestor de bases de datos se cuenta como uno de los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

1.3.6.4. Selección del gestor de base de datos: PostgreSQL

A pesar de ser un buen gestor de bases de datos la velocidad de respuesta con bases de datos pequeñas puede parecer baja, aunque mantiene la misma velocidad en bases de datos mucho más grandes. Consume más recursos que MySQL. No obstante, PostgreSQL es una buena alternativa a utilizar como gestor fundamentalmente en bases de datos grandes.

Instalación ilimitada: Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Mejor soporte que los proveedores comerciales: Además de las ofertas de soporte, tiene una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribución.

Ahorros considerables en costos de operación: Este software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Además de esto, los programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

Estabilidad y confiabilidad legendarias: En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

Multiplataforma: PostgreSQL está disponible en casi cualquier sistema Unix (34 plataformas en la última versión estable), y una versión nativa de Windows

1.3.7. Arquitectura cliente/servidor

Este modelo es definido por IBM como “la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo en múltiples plataformas”. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligente o “clientes”, resultan en un trabajo realizado por otros computadores llamados “servidores”. [Wikipedia, 2010]

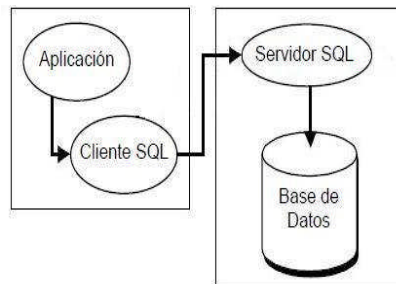


Figura 1.3: Modelo cliente/servidor distribuido en 3 capas

Esta arquitectura consiste básicamente en que un programa, el cliente realiza peticiones a otro programa, el servidor, que les da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema multiusuario distribuido a través de una red de computadoras. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Una disposición muy común son los sistemas multicapas en los que el servidor se descompone en diferentes programas y servicios que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema y su eficiencia. La arquitectura cliente/servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. [Wikipedia, 2010]

Las aplicaciones Web, son aplicaciones cliente/servidor, con una arquitectura compuesta por:

Cliente: Navegadores (Firefox, Internet Explorer, Netscape, Flock, Opera, Epiphany y otros).

Servidor: Servidores Web (Apache, Microsoft Internet Information Services y otros).

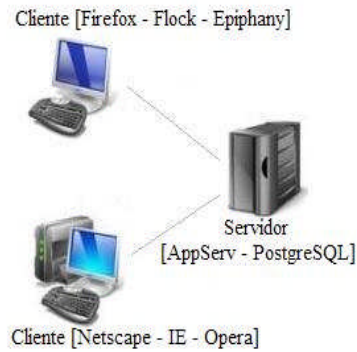


Figura 1.4: Sistema cliente/servidor

Ventajas

Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.

Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. El servidor de cliente es la arquitectura de red que separa al cliente de un servidor. Cada caso del software del cliente puede enviar peticiones a un servidor. Los tipos específicos de servidores incluyen los servidores de la Web, los servidores del uso, los servidores de archivo, los servidores terminales, y los servidores de correo. Mientras que sus propósitos varían algo, la arquitectura básica sigue siendo igual. [Wikipedia, 2010]

1.3.8. Estilo arquitectónico

Estilo arquitectónico MVC (Modelo Vista Controlador).

Para el diseño de aplicaciones con interfaces sofisticadas se utiliza el patrón de diseño Modelo Vista Controlador. La lógica de un interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, una forma de mezclar los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar la interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error. Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. [18]

Elementos del estilo arquitectónico:

- **Modelo:** Datos y reglas de negocio.
- **Vista:** Muestra la información del modelo al usuario.
- **Controlador:** Gestiona las entradas del usuario.

El **modelo** es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Definir las reglas de negocio (la funcionalidad del sistema). (Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor").
- Llevar un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero .bat que actualiza los datos, un temporizador que desencadena una inserción, etc).

La **vista** es el responsable de:

- Recibir datos del modelo y mostrarlo al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de actualizar, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El **controlador** es el responsable de:

- Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada a actualizar. (Una petición al modelo puede ser "Obtener_tiempo_de_entrega dado una nueva orden").

1.3.9. Metodología para el desarrollo de software

1.3.9.1. XP (Extreme Programming)

Esta metodología está muy extendida principalmente para los proyectos pequeños que cuentan con poco personal y poco tiempo. Se basa en ir construyendo el software y hacerle pruebas al mismo tiempo de modo que el cliente -que se convierte en parte del equipo- dé su opinión y de esa manera haya una retroalimentación; que permite construir el software de la manera que el cliente quiera. La metodología está diseñada para entregar al cliente el software que necesita, cuando lo necesita. [19]

1.3.9.2. MSF (Microsoft Solution Framework)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Originalmente creado en 1994 para conseguir resolver los problemas a los que se enfrentaban las empresas en sus respectivos proyectos, se ha convertido posteriormente en un modelo práctico. MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación. [20]

1.3.9.3. RUP (Rational Unified Process)

La metodología **RUP**, llamada así por sus siglas en inglés (Rational Unified Process), divide en 4 fases el desarrollo del software:

- Inicio: El objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura base.
- Construcción: En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- Transición: El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas 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. [21]

RUP está compuesto por tres elementos fundamentales

- Actividades: Son los procesos que se llegan a determinar en cada iteración.
- Trabajadores: Son las personas o entes involucrados en cada proceso.
- Artefactos: Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Dentro de cada iteración de cada fase se llevan a cabo nueve flujos de trabajo dentro de los cuales los seis primeros son llamados de ingeniería y los demás son de apoyo.

Flujos de Trabajo de Ingeniería

- **Modelado del negocio:** Este flujo identifica los procesos de negocio, los que estarán sujetos a automatización y quiénes intervienen en los mismos.
- **Requerimientos:** Se identifican las restricciones que se imponen y lo que el sistema debe hacer.
- **Análisis y Diseño:** Describe cómo el programa será realizado y define cómo será programado.
- **Implementación:** Define cómo estarán los nodos ubicados y la ubicación de los objetos y clases en paquetes.
- **Prueba:** Se localizan los defectos del software.
- **Instalación:** Se entrega una versión operacional.

Flujos de trabajo de apoyo

- **Administración de proyecto:** Encargado de organizar el trabajo y de que se termine el proyecto en el tiempo previsto.
- **Administración de configuración y cambio:** Describe el uso y actualización concurrente de los elementos, control de versiones entre otras actividades.
- **Ambiente:** Describe los procesos y herramientas que soportarán al equipo de trabajo del proyecto.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Esta metodología es muy utilizada en el proceso de desarrollo de software por ser flexible; además, al ser iterativa, permite que se vaya construyendo el software por ciclos, por lo cual se pueden detectar errores con tiempo de antelación. Es una metodología confiable pues desde su surgimiento ha tenido una gran aceptación. Por todos estos elementos el sistema propuesto estará guiado por el Proceso Unificado de Desarrollo de Software.

Las principales características de RUP son las siguientes:

Dirigido por Casos de Uso: Tiene a los casos de uso como el hilo conductor que orienta las actividades de desarrollo. Se centra en la funcionalidad que el sistema debe poseer para satisfacer las necesidades de un usuario (persona, sistema externo, dispositivo) que interactúa con él.

Centrado en la arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos, manejadores de bases de datos, protocolos; consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. El término que se emplea de Arquitectura de Software es similar al de la arquitectura de un edificio que requiere para su construcción de la elaboración de varios planos para conformar la imagen final del mismo.

Iterativo e incremental: Propone la descomposición de proyectos grandes en mini-proyectos, cada mini-proyecto es una iteración, y cada iteración debe estar controlada y tratar un determinado grupo de casos de uso.

Esta última característica de las iteraciones brinda beneficios al permitir reducir el coste del riesgo al coste de un solo incremento; que exista menos riesgo de no sacar el producto al mercado en fecha; así como acelerar el ritmo de desarrollo. Además hay que tener en cuenta que las necesidades del usuario y correspondientes requisitos no pueden definirse completamente al principio, de ahí la importancia de iteraciones sucesivas, que sirvan de retroalimentación.

1.3.9.4. Selección de la metodología para el desarrollo de software: RUP

Después de analizar las diferentes metodologías a adoptar para el desarrollo del sistema en cuestión se decidió que fuera RUP, por su potencialidad, ser la más conocida y la integración que posee con UML. Otro aspecto significativo que se tomó en cuenta fue que la mayor parte de las herramientas CASE poseen un entorno adaptado para el trabajo con RUP.

Seleccionar la metodología adecuada permite en gran medida guiar el proceso de desarrollo, encaminado a obtener un software con calidad, flexible y escalable, que cumpla con las necesidades de los usuarios. Para el desarrollo de la presente propuesta se utilizó RUP ya que brinda un conjunto básico de las mejores prácticas a seguir en el desarrollo moderno de software y proporciona una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. “RUP junto con UML (Unified Modeling Language) constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

1.3.10. Herramienta de modelación

1.3.10.1. Visual Paradigm for UML

Es una herramienta CASE (Computer Aided Software Engineering) en Español (Ingeniería de Software Asistida por Ordenador), profesional que soporta el ciclo de vida del desarrollo del 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 CASE también proporciona abundantes tutoriales, demostraciones interactivas de UML y proyectos UML. [22]

Características

- Generación automática de código.
- Multiplataforma.
- Capacidad para la ingeniería directa e inversa en Java, C++ y PHP.

Ventajas

- Posibilidad de usarlo conjuntamente con un servidor de trabajo en equipo (Team Work Server). Ofrece un entorno de creación de diagramas para UML, un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Usa un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Garantiza la disponibilidad de múltiples versiones de acuerdo a la necesidad.
- Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo.
- Es muy útil para la generación de código fuente en PHP.

Desventajas

- Carece de los estereotipos del diseño para aplicaciones Web.

1.3.10.2. Rational Rose

Es una herramienta para “modelado visual”, que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Incluye un conjunto de herramientas de ingeniería inversa y generación de código que facilitan el producto hasta el producto final. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción. [23]

1.3.10.3. Selección de la herramienta de modelación: Rational Rose

Estas razones fueron más que suficientes para que fuera elegida esta herramienta para el desarrollo de este proyecto. Fue elegida porque, a entender de los desarrolladores, es una de las que mejor se adapta a la incorporación de medidas y procedimientos de seguridad así como a las prácticas de diseño. Rational Rose es una herramienta que mejora la productividad del equipo de trabajo y entrega las mejores funcionalidades de la aplicación a todos los miembros del mismo.

1.3.11. Selección del lenguaje para el modelado del software: UML

Desde los inicios de la informática se han estado utilizando distintas formas de representar los diseños de una manera más bien personal o con algún modelo gráfico, La falta de estandarización en la representación gráfica de un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores, con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML es un lenguaje de estandarización que permite modelar, documentar y desarrollar los elementos que forman parte de un sistema software orientado a objetos. [24]

La naturaleza de su éxito radica principalmente en la naturalidad de su uso, pues se trata de un concepto nuevo ya que se emplean diagramas de distintos tipos para abordar el problema, de forma que la comprensión del problema en sí resulta mucho más "amigable" tanto para el desarrollador como para el cliente, pues éste también conocerá mucho mejor el producto que va a adquirir. Desde un punto de vista más técnico se podría decir que UML son un grupo de especificaciones de notación orientadas a Objeto, las cuales están compuestas por distintos diagramas, que representan las diferentes etapas del desarrollo de un

proyecto de software. Dada la gran integración de este lenguaje de modelado con la aplicación Rational Rose antes seleccionada no se pasó a hacer un estudio más profundo con otros lenguajes ya que este es el más acorde para nuestro proyecto y posee el mejor soporte por parte del Rational Rose.

Características del Sistema

2.1. Flujo actual de los procesos vinculados al campo de acción

Al Desoft División Habana establecer un contrato con algún cliente, lo primero que se hace es el “Acta de inicio de la implementación”, que debe ser firmada por ambas empresas y mediante la cual se establece la fecha de inicio del servicio de implementación del software solicitado. A partir de esta fecha el jefe de proyecto crea un cronograma en el cual definirá todas las etapas que tendrá el mismo, con sus fechas de inicio y de fin, se lo presenta al cliente y este debe aprobarlo.

Luego teniendo en cuenta lo definido en el cronograma, cada implementador deberá realizar sus tareas y al finalizar el mes debe elaborar el “Informe de Avance del Proyecto” que debe estar firmado por el cliente y el implementador a cargo del mismo. Es en este punto donde se enmarca el proceso a tratar por este trabajo de diploma. Este informe es entregado al subgerente del área de Implementación y Soporte, quien lo revisa y se lo entrega al registrador, que será el encargado de manipular esa información como es debido, es decir, digitalizar todo ese informe para que quede registrada la información, que es almacenada por la subgerente del área. Todo continúa de esa manera hasta la fecha de culminación del producto.

Con la información digitalizada y registrada completamente en los primeros 5 días hábiles, se publica en el FTP de la Casa Matriz y se procede a realizar un informe nacional.

2.2. Objeto de automatización

El objeto a automatizar es la gestión de información de proyectos, o sea, todo el proceso de documentación de la información generada actualmente por el implementador, se procura hacer de manera digital pero usando la aplicación que se pretende implementar. De esta manera se evitará ambigüedad en la información y generación de errores. La aplicación permitirá que el departamento tenga su información actualizada y disponible en todo momento beneficiando a la registradora encargada de la revisión de la documentación generada por el implementador. Así mismo la información estará disponible para el jefe de departamento que es el encargado de presentar el avance y seguimiento de cada proyecto ante la gerencia de la empresa. De forma general la aplicación traerá como beneficios una mayor facilidad en el trabajo aportando así eficiencia y claridad al proceso, logrando que exista unicidad en la información. Además el personal autorizado a trabajar con dicha aplicación contará siempre con información actualizada de manera que podrá generar un reporte fácil y siempre que se necesite.

2.3. Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio.

Nombre del actor	Descripción
Implementador	Es el encargado de crear toda la información correspondiente al proyecto en el cual se encuentra trabajando.

Tabla 2.1: Actores del negocio

2.4. Trabajadores del negocio

Un trabajador del negocio es una abstracción de un humano o de un sistema de software que representa un rol dentro del desarrollo de una realización de un caso de uso del negocio. Los trabajadores del negocio son los responsables de darle seguimiento a las peticiones o estímulos que le realizan los actores a dicho negocio y representan roles internos de la organización, mientras que los actores son los roles externos. El trabajador del negocio manipula objetos y colabora con otros para desempeñar sus responsabilidades. A cada trabajador del negocio se le debe identificar con un nombre que esté en correspondencia con el rol que desempeña dentro de éste

Trabajador	Descripción
Subgerente de Área	Es la responsable de revisar toda la información procedente de los implementadores y entregarla a la técnica en gestión para su manipulación.
Técnica en gestión	Se encarga de recibir la información que le entrega la subgerente y digitalizarla.

Tabla 2.2: Trabajadores del negocio

2.5. Diagrama de caso de uso del negocio

Para obtener una visión general de los diferentes procesos de negocio de la organización se construye un diagrama de casos de uso del negocio, donde cada caso de uso representa un proceso de negocio y cada actor representa un rol.

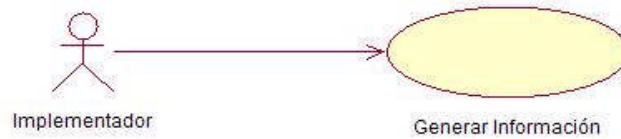


Figura 2.1: Diagrama de caso de uso del negocio.

2.6. Descripción del caso de uso del negocio

Caso de uso del negocio	Generar información
Actor del negocio	Implementador
Trabajador del negocio	Subgerente de Área, Técnica en gestión
Resumen	El caso de uso inicia después de estar aprobado el contrato con el cliente, es aquí cuando el implementador comienza con el seguimiento del proyecto de software. Donde se comienza a llevar información del estado del proyecto.
Acción del actor	Respuesta del negocio
Genera información	<p>1.1 El implementador tiene ya su proyecto asignado.</p> <p>1.2 Comienza el registro de información en la planilla “seguimiento de proyecto”.</p> <p>1.3 Luego se pasa a la conformación de la planilla “avance de proyecto” donde el encargado monitorea el trabajo diario de los programadores de los proyecto.</p> <p>1.4 Entrega la documentación mensual al subgerente de área.</p>
Mejoras	Con la automatización de este proceso, se genera información digital dando la posibilidad, de realizar revisiones y comparaciones posteriores para determinar si se han cumplido con los objetivos propuestos y lograr facilitar el trabajo del área a la

Tabla 2.3: Descripción del caso de uso del negocio «generar información»

2.7. Diagrama de actividades

Los diagramas de actividades representan gráficamente las actividades realizadas siguiendo una secuencia lógica durante la realización de un caso de uso del negocio.

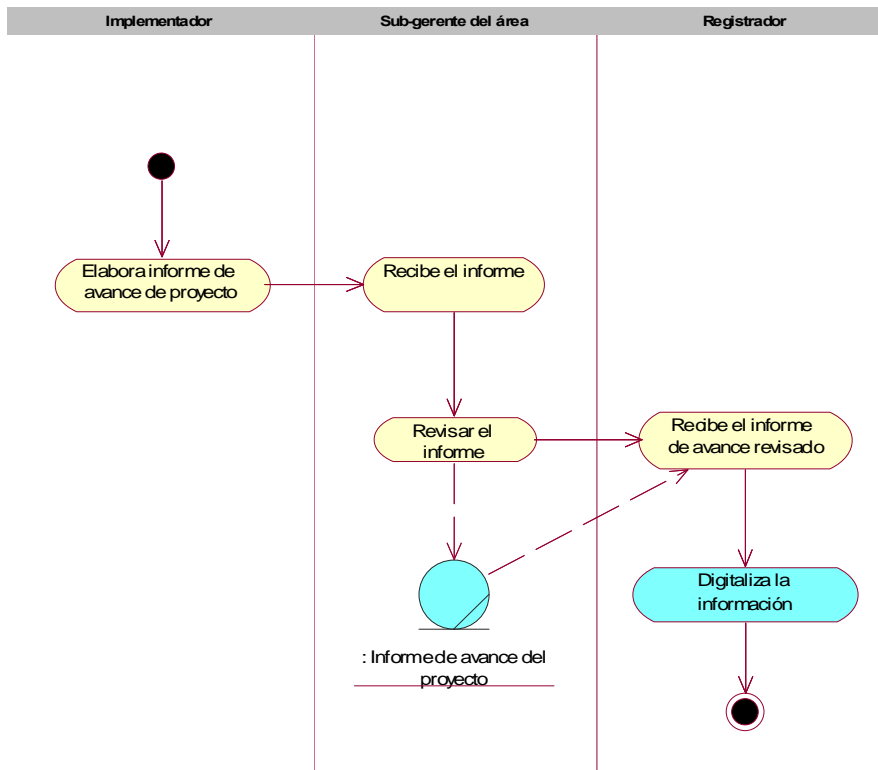


Figura 2.2: Diagrama de actividad «generar información»

2.8. Modelo de objeto del negocio

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

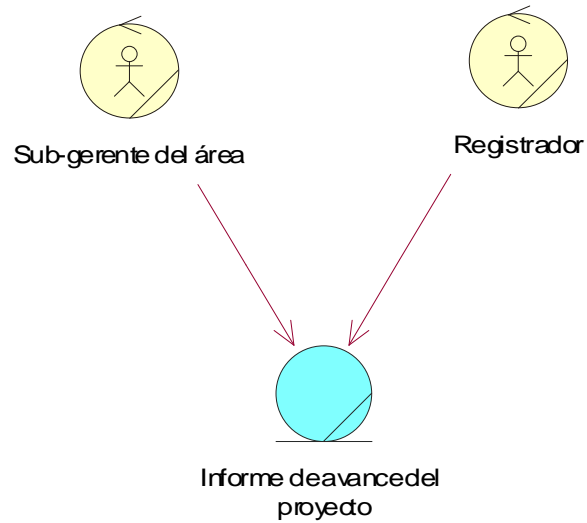


Figura 2.3: Modelo de objeto de negocio

2.9. Requisitos funcionales

R1. Gestionar organismo

- R1.1 Registrar organismo
- R1.2 Listar organismo
- R1.3 Modificar organismo
- R1.4 Eliminar organismo
- R1.5 Descargar organismo

R2. Gestionar empresa

- R2.1 Registrar empresa
- R2.2 Listar empresa
- R2.3 Modificar empresa
- R2.4 Eliminar empresa
- R2.5 Descargar empresa

R3. Gestionar cliente

- R3.1 Registrar cliente
- R3.2 Listar cliente
- R3.3 Modificar cliente

R3.4 Eliminar cliente

R3.5 Descargar cliente

R4. Gestionar contacto

R4.1 Registrar contacto

R4.2 Listar contacto

R4.3 Modificar contacto

R4.4 Eliminar contacto

R4.5 Descargar contacto

R5. Gestionar producto

R5.1 Registrar producto

R5.2 Listar producto

R5.3 Modificar producto

R5.4 Eliminar producto

R5.5 Descargar producto

R6. Gestionar implementador

R6.1 Registrar implementador

R6.2 Listar implementador

R6.3 Modificar implementador

R6.4 Eliminar implementador

R6.5 Descargar implementador

R7. Gestionar proyecto

R7.1 Registrar proyecto

R7.2 Listar proyecto

R7.3 Modificar proyecto

R7.4 Eliminar proyecto

R7.5 Descargar proyecto

R8. Gestionar avance de proyecto

R8.1 Registrar avance de proyecto

R8.2 Listar avance de proyecto

R8.3 Modificar avance de proyecto

R8.4 Eliminar avance de proyecto

R8.5 Descargar avance de proyecto

R9. Gestionar usuario

- R9.1 Registrar usuario
- R9.2 Listar usuario
- R9.3 Modificar usuario
- R9.4 Eliminar usuario
- R9.5 Descargar usuario

R10. Gestionar municipio

- R10.1 Registrar municipio
- R10.2 Listar municipio
- R10.3 Modificar municipio
- R10.4 Eliminar municipio
- R10.5 Descargar municipio

R11. Descargar reporte de implementación

R12. Descargar reporte de seguimiento

2.10. Requisitos no funcionales

Apariencia o interfaz interna

El sistema debe tener una interfaz legible, amigable, seria, profesional y simple de usar ya que los usuarios no necesariamente deben ser diestros en el uso de la computadora.

Usabilidad

El software debe ser fácil de usar por personas sin experiencia previa con las computadoras, permitiendo el acceso a opciones informativas y de carácter general, así como la disponibilidad de opciones más específicas para el administrador. La interfaz de usuario y edición deben ser consistentes.

Portabilidad

Para su Implementación se usaron Herramientas de Programación y Gestión de Bases de Datos que son multiplataforma, por lo que se garantiza que el sistema se pueda utilizar en SO Windows y/o Linux.

Seguridad

Existe el usuario administrador, que garantiza que el sistema será protegido de acceso no autorizado.

Confiabilidad

El sistema debe ser confiable, los fallos deben ser mínimos y de ocurrir, la recuperación debe ser lo más rápida posible. Cada cierto tiempo en el día el administrador realizará copias de seguridad (*backups*) a la base de datos para garantizar no perder toda la información.

Software

Se necesita un servidor Web con un intérprete PHP y un gestor de bases de datos PostgreSQL.

En las computadoras de los usuarios solo se requiere de un navegador (Firefox o Internet Explorer) para visualizar las páginas Web.

Hardware

Requerimientos mínimos para la conexión en el cliente: 64MB de RAM.

Requerimientos mínimos Servidor de Base de Datos: 128MB de RAM 10 GB de disco duro.

Requerimientos mínimos Servidor Web: 128Mb de RAM 10 GB de disco duro.

2.11. Actores del sistema

Actor	Descripción
Usuario	El cual solo tendrá permiso a la funcionalidad de generar reportes.
Registrador	Gestionar toda la información referente a los proyectos.
Administrador	Es el encargado de manipular los usuarios, municipios. Además tendrá acceso a todas las funcionalidades del sistema.

Tabla 2.4: Descripción de los actores del sistema.

2.12. Diagrama de casos de uso del sistema

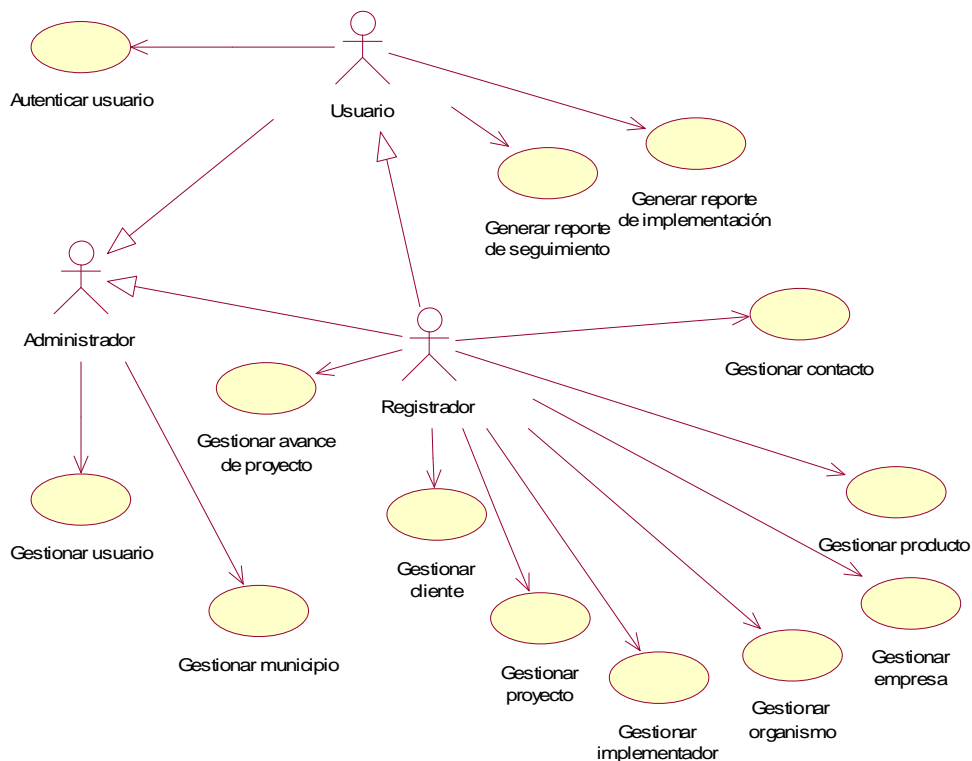


Figura 2.4: Diagrama del caso de uso del sistema

2.13. Descripción de los caso de uso del sistema

Caso de uso del sistema	Gestionar cliente
Actores del sistema	Registrador (inicia)
Propósito	Tener registrado en el sistema la información necesaria de un cliente de la Empresa Desoft División Habana.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del cliente en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R3, R3.1, R3.2, R3.3, R3.4, R3.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El cliente fue gestionado.
Prioridad	Crítico

Tabla 2.5: Descripción textual del caso de uso «gestionar cliente»

Caso de uso del sistema	Gestionar proyecto
Actores del sistema	Registrador (inicia)
Propósito	Tener registrado en la base de datos de la aplicación la información necesaria de un proyecto de la Empresa Desoft División Habana.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del proyecto en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R7, R7.1, R7.2, R7.3, R7.4, R7.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El proyecto fue gestionado.
Prioridad	Crítico

Tabla 2.6: Descripción textual del caso de uso «gestionar proyecto»

Las otras descripciones se podrán encontrar en el Anexo 1.

CAPÍTULO

3

Diseño e Implementación

3.1. Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso. Para ello se centra en el impacto sobre el sistema a considerar. En él incurren los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de programación, convirtiéndose en la entrada fundamental para las actividades de implementación.

3.2. Definición de elementos de diseño

El sistema se encuentra dividido en tres capas bien definidas, que son la presentación, la lógica del negocio y la capa de acceso a datos todo enmarcado en el estilo arquitectónico MVC.

- **Presentación:** Incluye las clases interfaces, a través de las cuales se establece la comunicación con los usuarios, y para dar respuesta a los pedidos de estos se invocarán los métodos de la lógica del negocio.
- **Lógica del negocio:** Incluye las clases controladoras, así como las entidades del negocio, que representan objetos que van a ser manejados o consumidos por toda la aplicación.
- **Acceso a datos:** Contiene clases que interactúan con la base de datos y permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio y los usuarios.

3.3. Diagrama de clases del diseño

Estos diagramas son utilizados para mostrar cómo puede ser construido el sistema. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase, es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente y obtener así el producto final con la calidad requerida.

Clases	Descripción
sv	Representa la clase que tiene código que se ejecuta

	en el servidor, la cual se encarga de construir (build) o generar el resultado HTML y/o realizar peticiones a la capa inferior.
cl	Es una página Web con formato XHTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script, que son accesibles para cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.
form	Es una colección de elementos de entrada que están contenidos en la página cliente. Sus atributos son los elementos de entrada del formulario. Estos se comunican con las páginas servidores mediante submit.
<<build>>	Significa que una página servidor construye a una página cliente.
<<submit>>	Significa que los formularios envían sus datos al código servidor para ser procesados los pedidos.

Tabla 3.3. Terminología de los diagramas de clases de diseño.

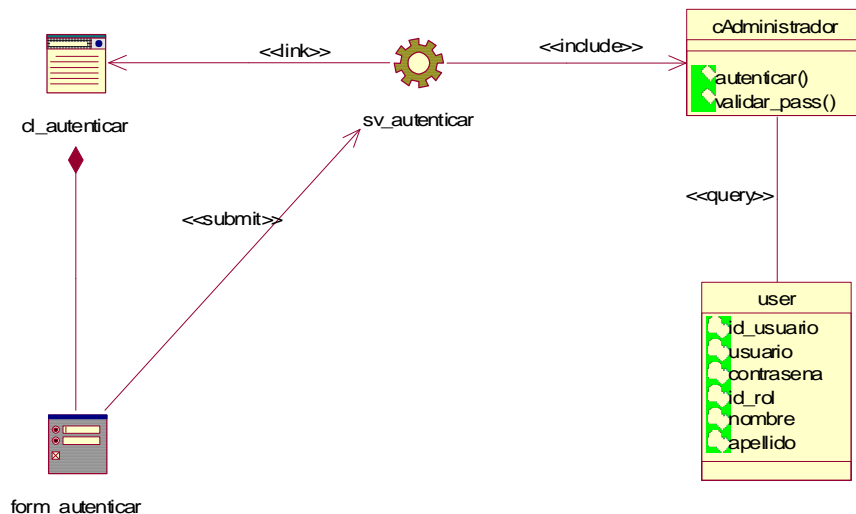
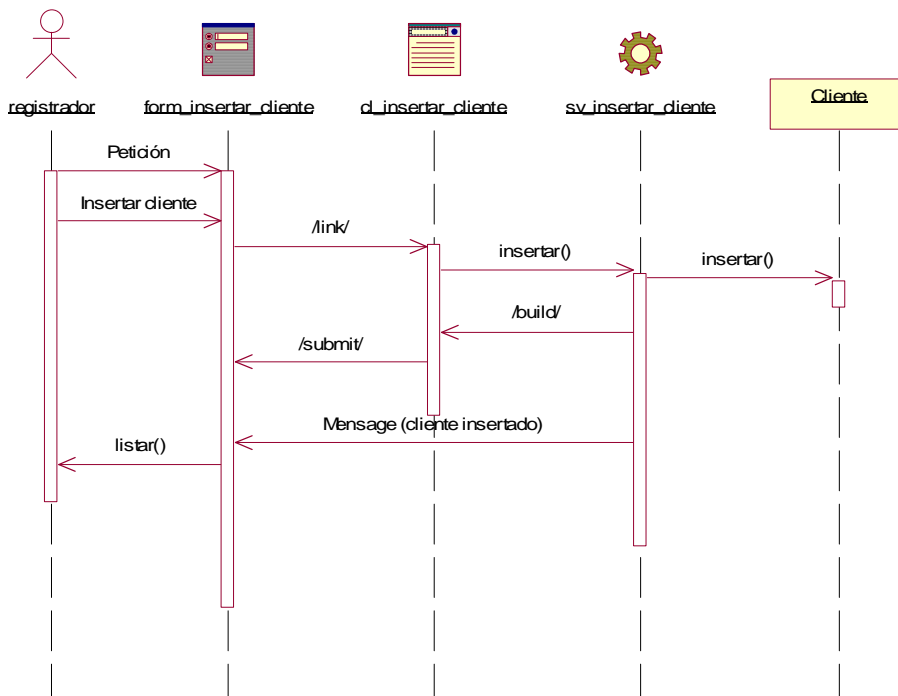


Figura 3.1. Diagrama de clases del diseño «autenticar»

Los otros diagramas de clases se podrán encontrar en el Anexo 2.

3.4. Diagrama de secuencias

El **diagrama de secuencia** es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.



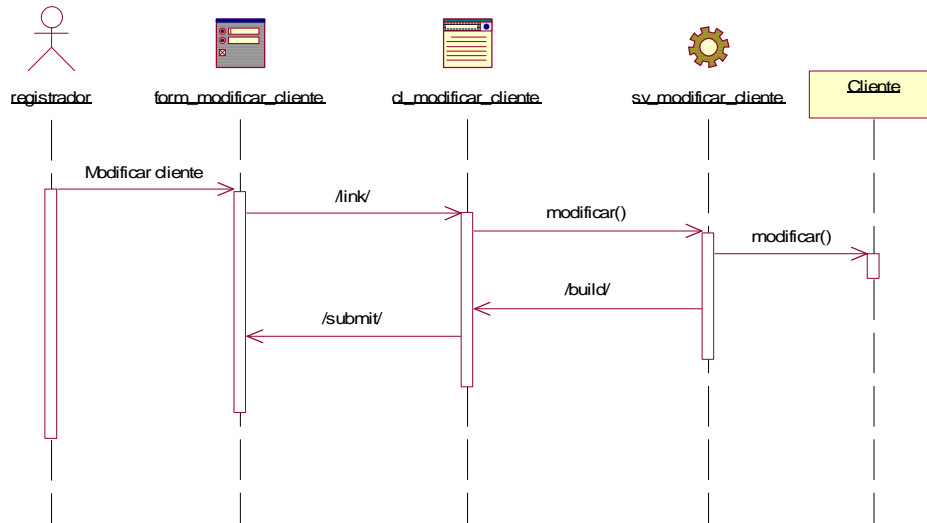
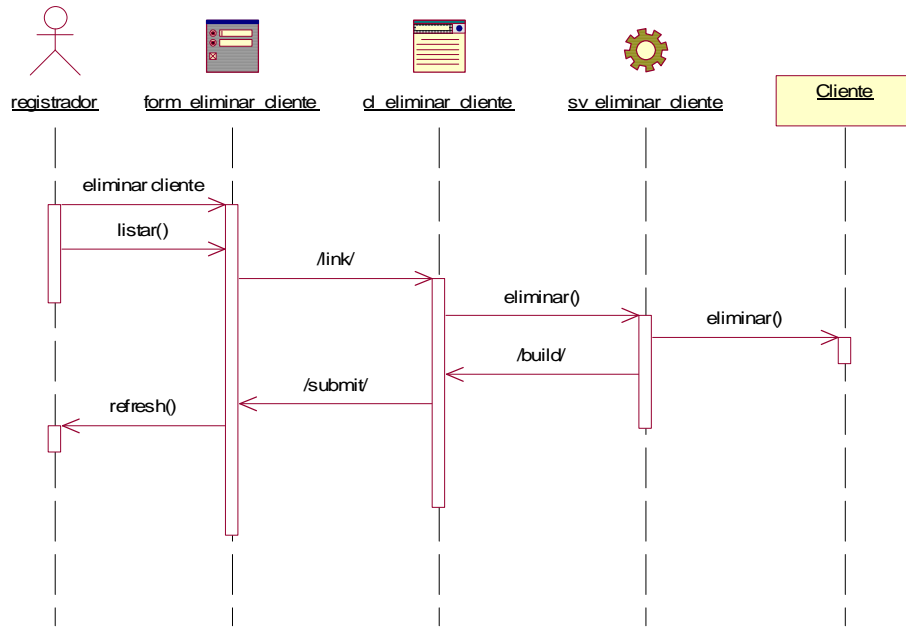
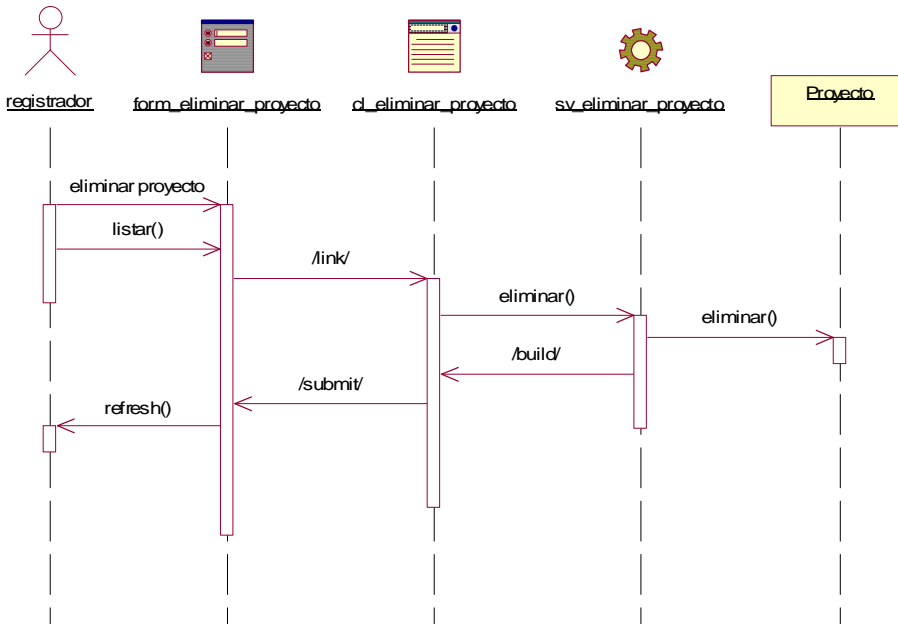
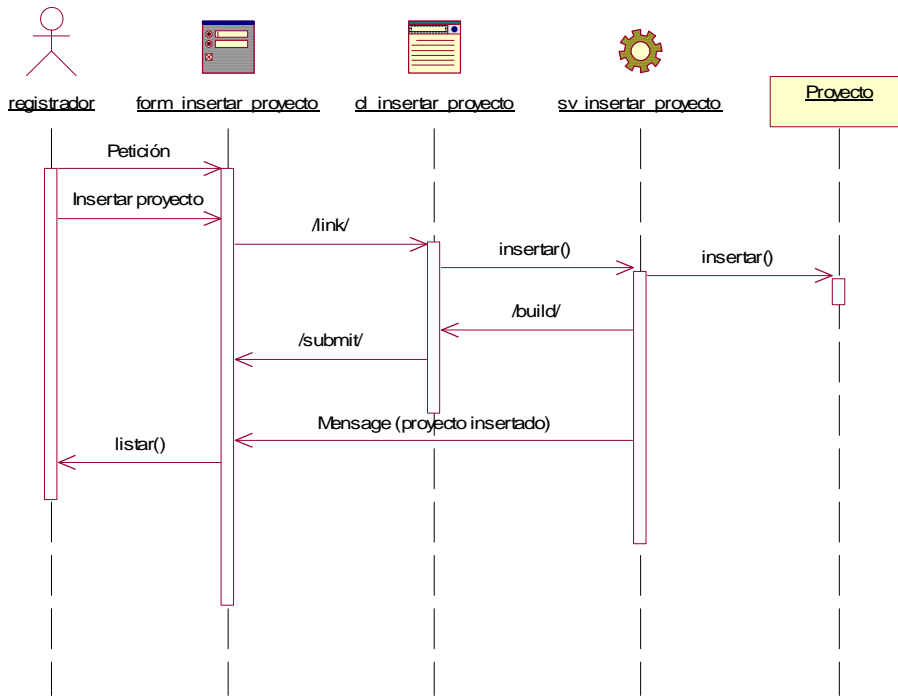


Figura 3.2. Diagrama de secuencia del caso de uso «gestionar cliente»



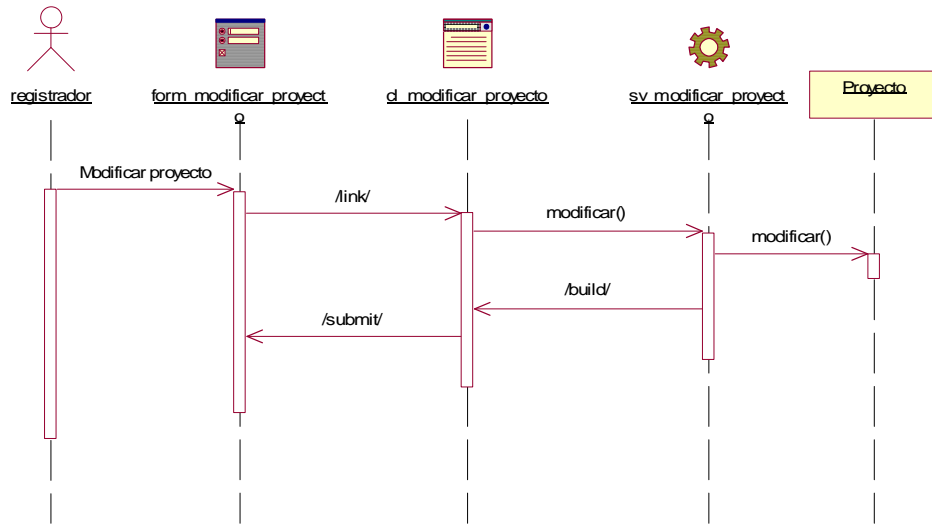


Figura 3.3. Diagrama de secuencia del caso de uso «gestionar proyecto»

Los otros diagramas de secuencias se podrán encontrar en el Anexo 4.

3.5. Interfaz de usuario



Figura 3.4. Interfaz de usuario

3.6. Modelo físico de datos

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema.

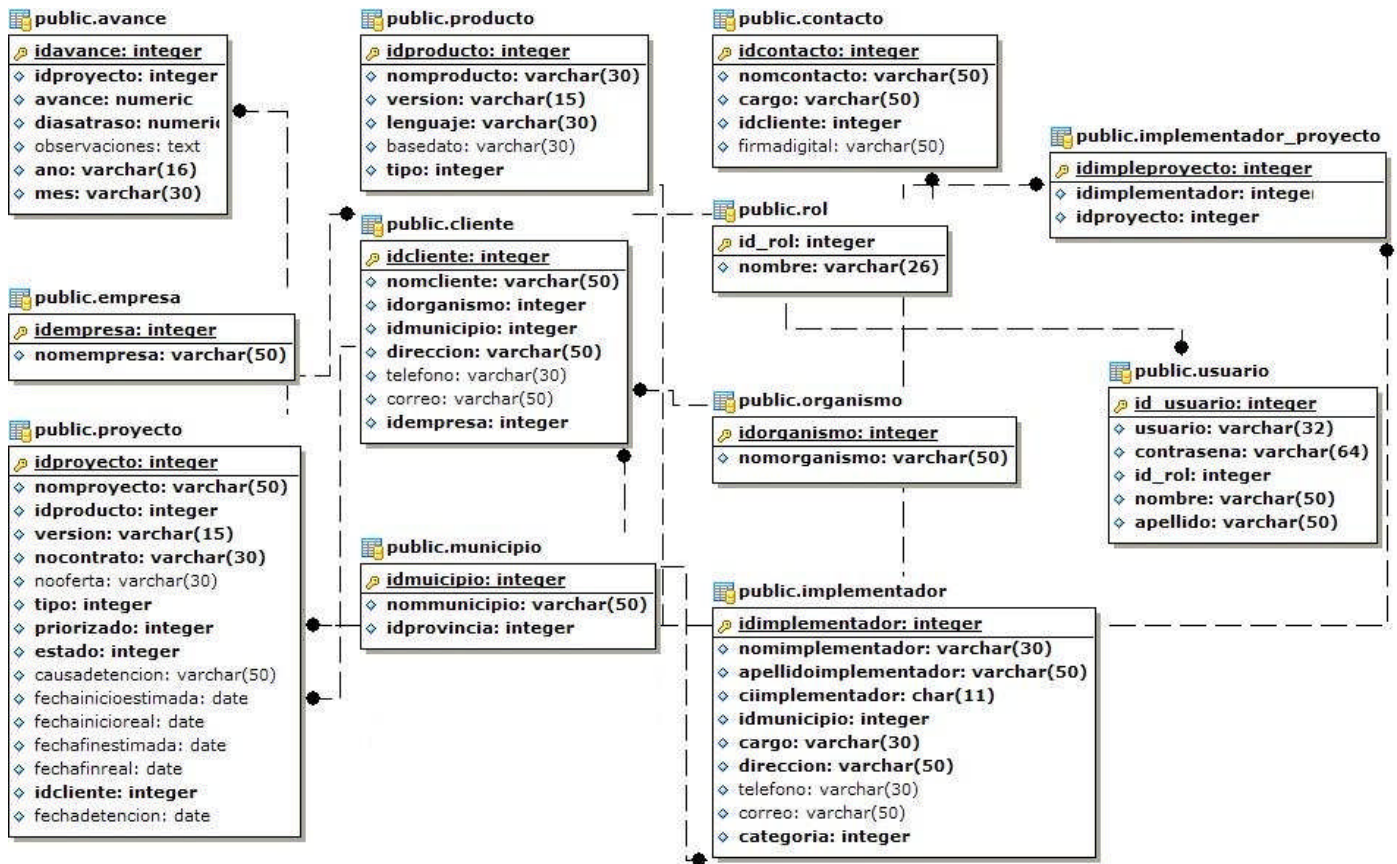


Figura 3.5. Modelo de datos.

3.7. Diagrama de despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que describe la arquitectura física del sistema. Para poder realizar el despliegue en la institución, la misma debe contar con una red LAN y PC clientes que posean un navegador Web. También es necesario un servidor Web y un servidor de Base de Datos. La comunicación entre las PC clientes y el servidor Web es mediante el protocolo HTTP la comunicación entre el servidor Web y el servidor Base Datos es mediante el protocolo TCP/IP.

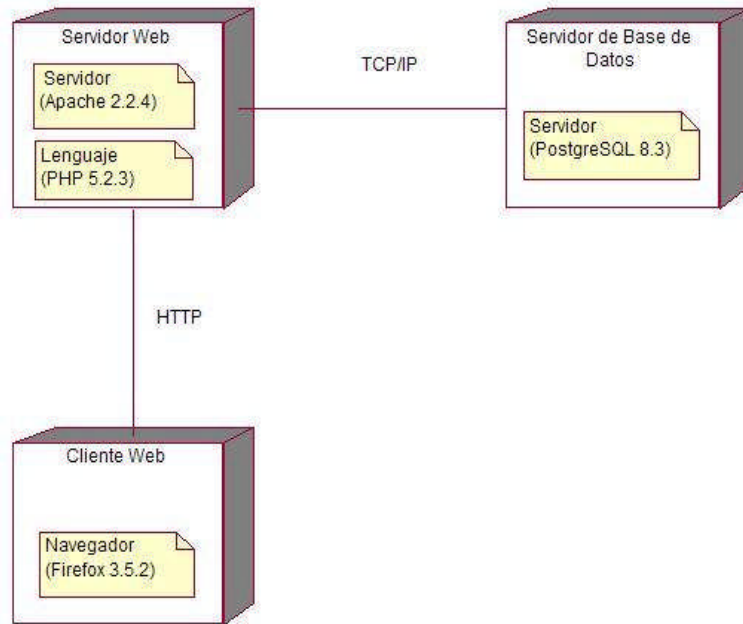


Figura 3.6. Diagrama de despliegue

3.8. Diagrama de componentes

Un componente es una parte física y reemplazable de un sistema. Es la implementación física de un conjunto de otros elementos lógicos, como clases y colaboraciones. El diagrama de componentes se utiliza para modelar la vista estática de un sistema. Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes.

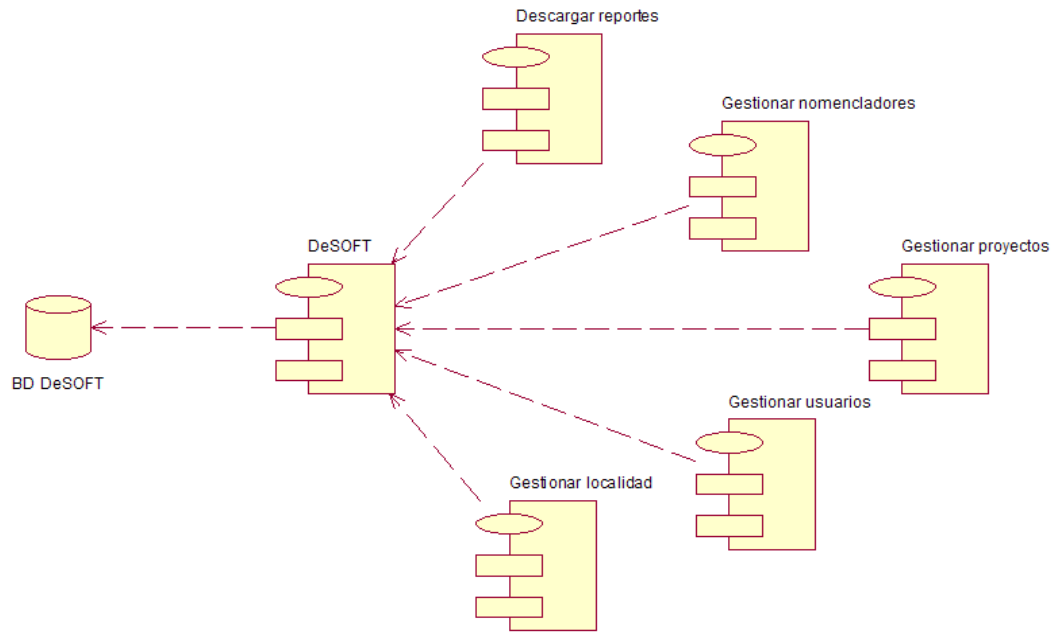


Figura 3.7. Diagrama de componentes «general»

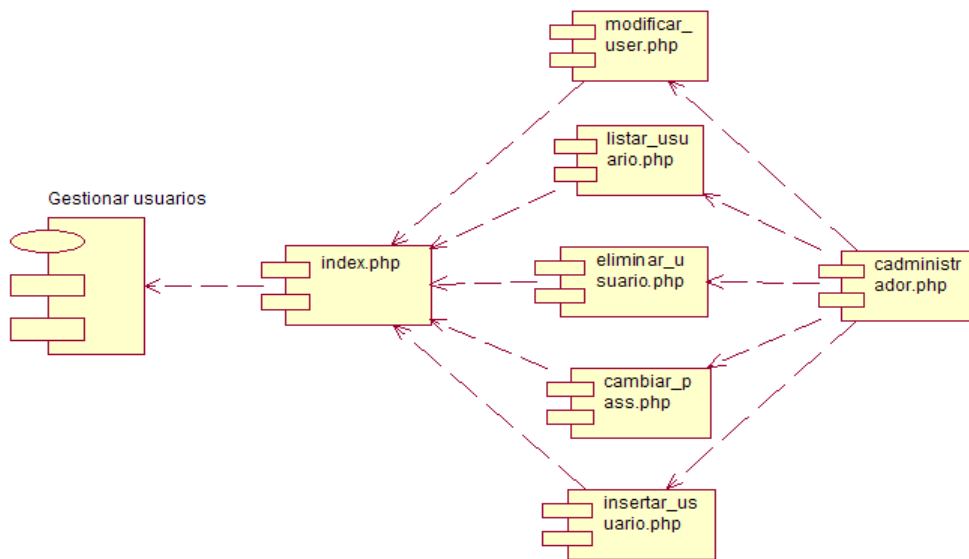


Figura 3.8. Diagrama de componentes «gestionar usuarios»

Los otros diagramas de componentes se podrán encontrar en el Anexo 5.

CAPÍTULO

4

Estudie Factibilidad

4.1 Planificación por punto de función

La estimación del proyecto se realizó mediante los puntos de función desajustados utilizando el modelo COCOMO II, los cuales se utilizan para el cálculo de las instrucciones fuentes. De esta forma se estima la magnitud del sistema y se obtienen además indicadores como la cantidad de hombre, el esfuerzo, el tiempo de duración y el costo del mismo. Las funciones del sistema se agrupan: Entradas externas, Salidas externas, Ficheros internos e Interfaces externas y estas se clasificarán por su nivel de complejidad en: Bajo, Medio, Alto.

Nombre de la entrada externa (EI)	Cantidad de Ficheros	Cantidad de elementos de datos	Calcificación (Bajo, Medio, Alto)
Registrar organismo	1	2	Bajo
Modificar organismo	1	2	Bajo
Registrar empresa	1	2	Bajo
Modificar empresa	1	2	Bajo
Registrar cliente	1	8	Bajo
Modificar cliente	1	8	Bajo
Registrar contacto	1	5	Bajo
Modificar contacto	1	5	Bajo
Registrar producto	1	6	Bajo
Modificar producto	1	6	Bajo
Registrar implementador	1	10	Bajo
Modificar implementador	1	10	Bajo
Registrar proyecto	1	16	Bajo
Modificar proyecto	1	16	Bajo
Registrar avance de proyecto	1	7	Bajo
Modificar avance de proyecto	1	7	Bajo
Registrar usuario	1	6	Bajo
Modificar usuario	1	6	Bajo
Registrar municipio	1	3	Bajo
Modificar municipio	1	3	Bajo

Cantidad de ficheros		
Bajo	Medio	Alto
19	0	0

Tabla 4.1: Entradas externas (EI)

Nombre de la salida externa (EO)	Cantidad de Ficheros	Cantidad de elementos de datos	Calcificación (Bajo, Medio, Alto)
Listar organismo	1	2	Bajo
Descargar organismo	1	2	Bajo
Listar empresa	1	2	Bajo
Descargar empresa	1	2	Bajo
Listar cliente	1	8	Bajo
Descargar cliente	1	8	Bajo
Listar contacto	1	5	Bajo
Descargar contacto	1	5	Bajo
Listar producto	1	6	Bajo
Descargar producto	1	6	Bajo
Listar implementador	1	10	Bajo
Descargar implementador	1	10	Bajo
Listar proyecto	1	16	Bajo
Descargar proyecto	1	16	Bajo
Listar avance de proyecto	1	7	Bajo
Descargar avance de proyecto	1	7	Bajo
Listar usuario	1	6	Bajo
Descargar usuario	1	6	Bajo
Listar municipio	1	3	Bajo
Descargar municipio	1	3	Bajo
Descargar reporte de implementación	1	2	Bajo
Descargar reporte de seguimiento	1	2	Bajo
Cantidad de ficheros			

Bajo	Medio	Alto
19	0	0

Tabla 4.2: Salidas externas (EO)

Nombre de las peticiones (EQ)	Cantidad de Ficheros	Cantidad de elementos de datos	Calcificación (Bajo, Medio, Alto)
Eliminar organismo	1	2	Bajo
Eliminar empresa	1	2	Bajo
Eliminar cliente	1	8	Bajo
Eliminar contacto	1	5	Bajo
Eliminar producto	1	6	Bajo
Eliminar implementador	1	10	Bajo
Eliminar proyecto	1	16	Bajo
Eliminar avance de proyecto	1	7	Bajo
Eliminar usuario	1	6	Bajo
Eliminar municipio	1	3	Bajo
Cantidad de ficheros			
Bajo	Medio	Alto	
10	0	0	

Tabla 4.3: Consultas externas (EQ)

Nombre de los ficheros lógicos (ILF)	Cantidad de Ficheros	Cantidad de elementos de datos	Calcificación (Bajo, Medio, Alto)
avance	1+	7	Bajo
cliente	1+	8	Bajo
contacto	1+	5	Bajo
empresa	1+	2	Bajo
implementador	1+	10	Bajo
implementador_proyecto	1+	3	Bajo
municipio	1+	3	Bajo

organismo	1+	2	Bajo
producto	1+	6	Bajo
proyecto	1+	16	Bajo
rol	1+	2	Bajo
usuario	1+	6	Bajo
Cantidad de ficheros			
Bajo	Medio		Alto
12	0		0

Tabla 4.4: Ficheros lógicos internos (ILF)

Elementos	Bajo		Medio		Alto		Subtotal de puntos de función
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	12	3	0	4	0	6	36
Entradas externas	19	4	0	5	0	7	76
Salidas externas	19	7	0	10	0	15	133
Consultas externas	10	3	0	4	0	6	30
Total (UFP)							275

Tabla 4.5: Puntos de función desajustados (UFP)

4.1.1 Características del proyecto

Cálculo de la cantidad de instrucciones fuentes (SLOC) mediante la fórmula siguiente:

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

$$\text{SLOC} = 275 \times 69$$

$$\text{SLOC} = 18\,975$$

$$\text{KSLOC} = 18.975 \text{ (miles de líneas de código)}$$

UFP: Total de puntos de función desajustados.

ratio: Constante para las SLOC de cada lenguaje de programación, en el caso de PHP es 69.

Factor de escala	Valor	Justificación
Precedencia (PREC)	2.48	Resulta familiar para los desarrolladores el tipo de aplicación.
Flexibilidad (FLEX)	2.03	Hubo cierto acuerdo de forma general en cuanto a las interfaces de diseño y los requisitos del software.
Riesgos (RESL)	4.24	Se tomó ciertas estrategias para tener el mínimo de riesgos en el entorno de la aplicación.
Cohesión del equipo (TEAM)	0.00	No se requiere más de un desarrollador de la aplicación.
Madures (PMAT)	7.80	Existe gran madurez en cuanto a la complejidad del software.

Tabla 4.5: Factor de escala

Multiplicador (+)	Valor
RCPX	1.00
RUSE	1.07
PDIF	1.29
PERS	0.83
PREX	1.00
FCIL	0.63
SCED	1.00

Tabla 4.6: Multiplicadores de esfuerzo (EM)

Características	
Puntos de función desajustados	275
ratio (PHP)	69
SLOC = UFP × Ratio	18 975
KSLOC	18. 975

Tabla 4.7: Cantidad de instrucciones fuentes

Esfuerzo de desarrollo (PM).

Sumatoria de los factores de escala (ΣSF):

$$\Sigma SF = \text{PREC} + \text{FLEX} + \text{RESL} + \text{TEAM} + \text{PMAT}$$

$$\Sigma SF = 2.48 + 2.03 + 4.24 + 0.00 + 7.80$$

$$\Sigma SF = 16.55$$

Producto de los multiplicadores de esfuerzo:

$$\Pi EM = \text{RCPX} \times \text{RUSE} \times \text{PDIF} \times \text{PERS} \times \text{PREX} \times \text{FCIL} \times \text{SCED}$$

$$\Pi EM = 1.00 \times 1.07 \times 1.29 \times 0.83 \times 1.00 \times 0.63 \times 1.00$$

$$\Pi EM = 0.7217$$

$$E = B + 0.01 \times \Sigma SF$$

$$E = 0.91 + 0.01 \times 16.55$$

$$E = 1.0755$$

Donde:

PM: esfuerzo de desarrollo.

A: es una constante, tomamos el valor por defecto del modelo, ajustado en **2.94**.

Size: tamaño estimado del software, en KSLOC (miles de líneas de código).

B: es una constante denominada “factor escalar”, su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto **0.91**

Cálculo del esfuerzo:

$$PM = A \times (\text{Size})^E \times \Pi EM$$

$$PM = 2.94 \times (18.975)^{1.0755} \times 0.7217$$

$$PM = 50.2792 \approx 50$$

Se necesitan 50 personas para un mes de trabajo.

Tiempo de desarrollo (TDEV)

Donde:

C = valor constante 3.67

D = valor constate 0.28

Cálculo del exponente de escala:

$$F = D + 2.0 \times (E - B)$$

$$F = 0.28 + 2.0 \times (1.0755 - 0.91)$$

$$F = 0.611$$

Cálculo del tiempo de desarrollo:

$$TDEV = C \times (PM)^F$$

$$TDEV = 3.67 \times (50.2792)^{0.611}$$

$$TDEV = 40.1989 \approx 40$$

Se necesitan 40 meses para el desarrollo del software

Cálculo de la cantidad de hombres:

$$CH = PM / TDEV$$

$$CH = 50.2792 / 40.1989$$

$$CH = 1.2507 \approx 2$$

Se necesitan 2 personas para el desarrollo del software

Costos del software.

El costo del proyecto depende del salario promedio de las personas que lo ejecutan y del esfuerzo que ellas realizan para el desarrollo del mismo.

Donde:

C: es el costo del proyecto.

CHM: es el costo de hombres por mes.

SP: salario básico de un ingeniero.

Cálculo del costo hombres por mes:

$$CHM = CH \times SP$$

$$CHM = 1.2507 \times 100$$

$$\text{CHM} = 125.07$$

Cálculo del costo del proyecto:

$$C = \text{CHM} \times \text{PM}$$

$$C = 125.07 \times 50.2792$$

$$C = 6\,280.42$$

El software cuesta desarrollarlo \$6 280.42

4.2. Beneficios tangibles e intangibles

Estos beneficios pueden contribuir sustancialmente a la economía, a la calidad de los productos finales y al prestigio de la Empresa.

4.2.1. Beneficios tangibles

El resultado del presente trabajo traería consigo beneficios en cuanto al ahorro en tiempo y recursos, posibilitaría la gestión de los proyectos a un alto grado, así como ayudaría con la moral del equipo de desarrollo prestando una gran ayuda a la técnica en gestión de información, pasando a ser la herramienta principal del departamento.

4.2.2. Beneficios intangibles

El producto permitirá publicar y compartir la información asociada a cada proyecto de software, mejora en la calidad de la información por la integridad, oportunidad y confiabilidad, mejoramiento de las condiciones de trabajo del personal del departamento.

4.3. Análisis de costos y beneficios

El desarrollo de un producto informático siempre tiene un costo. Este puede estar justificado por los beneficios del mismo. El costo, que significa la implementación en este caso no es elevado si se tiene en cuenta las contribuciones que trae consigo al área de implementación. Además, las tecnologías y herramientas utilizadas para el desarrollo del sistema son en casi todo su conjunto libres y código abierto,

por tanto no es necesario incurrir en gastos en el pago de licencias de uso. Por ello se plantea que es factible el desarrollo del mismo.

Conclusiones

Con el desarrollo de la aplicación para la gestión de información de proyectos de la Empresa Desoft División Habana se dio cumplimiento a los objetivos propuestos por este trabajo, pues se obtuvo como resultado un producto informático en el que se aplicaron los resultados de la investigación realizada arribándose a la siguiente conclusión:

Se elaboró el marco teórico metodológico que fundamenta la investigación, permitiendo analizar el proceso y la estructura de la gestión de información de proyectos, así como un detallado análisis para determinar las principales herramientas usadas en el desarrollo de la aplicación. Se logró un sistema informático que permite el control de la información de los proyectos y un estudio de factibilidad donde se obtuvo como resultado el tiempo de estimación para el desarrollo del sistema, el esfuerzo y la cantidad de personas necesarias para la realización del mismo, así como el costo final del mismo.

Recomendaciones

Por la importancia que tiene el presente trabajo para la eficiencia del trabajo en el departamento de implementación de la Empresa Desoft División Habana se recomienda lo siguiente:

- Realizar un estudio más profundo del funcionamiento actual del sistema de gestión de información de proyectos en el departamento respaldado por una mejor coordinación con el propósito de mejorar aún más el presente sistema.
- Implementar un módulo especializado para el cálculo detallado del avance de cada uno de los proyectos en ejecución.
- Implementar un módulo de reportes para los proyectos detenidos enviando un mensaje al jefe de departamento con las causas asociadas a la detención.

Referencias bibliográficas

1. [en línea] [Consultado: 19 de Diciembre de 2009]
http://es.wikipedia.org/wiki/Gestión_de_proyectos
2. [en línea] [Consultado: 12 de Enero de 2010]
http://es.wikipedia.org/wiki/Gestión_de_proyectos
3. Haugland Cato “El EML” [en línea] [Consultado: 17 de Enero de 2010]
<http://www.eml.org/NORADManualdeMarcoLogico.pdf>
4. [en línea] [Consultado: 21 de Diciembre de 2009]
<http://www.myworkplan.com>
5. [en línea] [Consultado: 23 de Diciembre de 2009]
http://intra.undp.org/newsletters/ea_registration_project_co.doc
6. [en línea] [Consultado: 9 de Enero de 2010]
<http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>
7. [en línea] [Consultado: 12 de Enero de 2010]
<http://blogs.technet.com/b/dbordini/archive/2008/01/24/iis-7-0-internet-information-services-parte-3-pilares.aspx>
8. [en línea] [Consultado: 10 de Enero de 2010]
<http://es.wikipedia.org/wiki/Zope>
9. Alonso Sergio A. “Técnicas para armado de redes” [en línea] [Consultado: 11 de Enero de 2010]
<http://www.bunker.org.ar/incubadora/redes.pdf>
10. [en línea] [Consultado: 10 de Enero de 2010]
www.buscandohost.com/hosting_glosario.php
11. [en línea] [Consultado: 10 de Enero de 2010]
http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html
12. Introducción a PHP 5. [en línea] [Consultado: 11 de Enero de 2010]
<http://www.desarrolloweb.com/articulos/1696.php>.
13. Gutiérrez., Javier J. Framework. [en línea] [Consultado: 15 de Enero de 2010]
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
14. [en línea] [Consultado: 15 de Enero de 2010]
<http://www.desarrolloweb.com/articulos/php-designer.html>
15. [en línea] [Consultado: 16 de Enero de 2010]

www.solodisenio.com/mysql-sistemas-gestores-de-bases-de-datos/

16. Lindstrom, Gary (2002, Sept.). Lecture 14: Oracle Extended Relational Features.
<http://www.cs.utah.edu/classes/cs5530/lectures/lecture14x2.pdf>, Nov. 2, 2002.
17. Introducción a PostgreSQL 8.3 [en línea] [Consultado: 21 de Enero de 2010]
<http://www.aplicacionesempresariales.com/stag/ventajas-y-desventajas-de-postgresql.html>
18. [en línea] [Consultado: 18 de Enero de 2010]
<http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml>
19. [en línea] [Consultado: 18 de Enero de 2010]
<http://rafinguer.blogspot.com/2010/03/metodologias-tradicionales-versus.html>
20. [en línea] [Consultado: 19 de Enero de 2010]
http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html
21. [en línea] [Consultado: 20 de Enero de 2010]
<http://www.willydev.net/InsiteCreation/EIFlecha.aspx?q=msf>
22. [en línea] [Consultado: 20 de Enero de 2010]
<http://www.monografias.com/trabajos73/herramientas-case-proceso-desarrollo-software/herramientas-case-proceso-desarrollo-software2.shtml>
23. [en línea] [Consultado: 24 de Enero de 2010]
http://www.angelfire.com/planet/carlosweiss/universidad_nacional_mayor_de_san_marcos_tesis2_final.doc/
24. Getting Started with UML. [en línea] [Consultado: 7 de Enero de 2010]
<http://www.omg.org/UML>.

Glosario de términos

Listado de términos ordenados alfabéticamente, que son usados en este trabajo de diploma.

Administrador: Es la persona que tiene privilegios para determinadas funcionalidades del sistema.

Apache: Servidor web de código abierto multiplataforma (Unix, Linux, Windows, ect.).

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

Cartera: Define las aplicaciones que fueron realizados o tienen un contrato establecido con la empresa a los cuales se le da un seguimiento (soporte, implementación o mantenimiento).

FTP: Protocolo de transferencia de archivo.

HTML: *HiperText Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 1879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: Protocolo de Transferencia de Hipertexto.

HTTPS: Protocolo de Transferencia de Hipertexto Seguro: Es una combinación del HTTP con el protocolo SSL/TLS para proveer encriptación y asegurar la identificación del servidor. Usa el puerto 443. Las conexiones de HTTPS sirven a menudo para transacciones sensitivas en los sistemas de información corporativos.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

NNTP: Protocolo de Transferencia de Noticias de Red: Es un protocolo de aplicación de Internet usado para el transporte se artículos y noticias.

NCSA HTTPD: fue un servidor de trama originalmente desarrollado en la NCSA por Robert McCool y estuvo entre los primeros servidores de la red.

Open Source: Código Abierto. Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer modificaciones, mejoras o reutilizarlo en otras aplicaciones. También conocido como free software.

PHP: *Hypertext Pre-processor* Lenguaje de programación para el desarrollo de aplicaciones web, con sintaxis parecida a la de C. Originalmente se conocía como *Personal Home Page*.

SGBD: Sistema de Gestión de Bases de datos. Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

SMTP: Protocolo de Transferencia de Correo Simple: Es un estándar de la Internet para la transmisión del correo electrónico a través del Protocolo Internet (IP).

TCP/IP: Es el conjunto básico de protocolos de comunicación de redes, popularizado por Internet, que permiten la transmisión de información en redes de computadoras. El nombre TCP/IP proviene de dos protocolos importantes de la familia, el *Transmission Control Protocol (TCP)* y el *Internet Protocol (IP)*.

UML: Unified Modeling Language. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

Anexo 1: Descripción de los casos de uso del sistema

Caso de uso del sistema	Gestionar organismo
Actores del sistema	Registrador (inicia)
Propósito	Permitir al registrador poder gestionar un organismo en el sistema.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del organismo en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R1, R1.1, R1.2, R1.3, R1.4, R1.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El organismo fue gestionado.
Prioridad	Crítico

Tabla 2.7: Descripción textual del caso de uso «gestionar organismo»

Caso de uso del sistema	Gestionar empresa
Actores del sistema	Registrador (inicia)
Propósito	Permitir al registrador poder gestionar una empresa en el sistema de forma correcta.
Resumen	El caso de uso inicia cuando el registrador inserta los datos de la empresa en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R2, R2.1, R2.2, R2.3, R2.4, R2.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	La empresa fue gestionada.
Prioridad	Crítico

Tabla 2.8: Descripción textual del caso de uso «gestionar empresa»

Caso de uso del sistema	Gestionar contacto
--------------------------------	--------------------

Actores del sistema	Registrador (inicia)
Propósito	Conocer toda la información disponible de los contactos de cada proyecto en el que se está trabajando.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del contacto en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R4, R4.1, R4.2, R4.3, R4.4, R4.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El contacto fue gestionado.
Prioridad	Crítico

Tabla 2.9: Descripción textual del caso de uso «gestionar contacto»

Caso de uso del sistema	Gestionar producto
Actores del sistema	Registrador (inicia)
Propósito	Disponer de la información de un producto determinado en la base de datos de la aplicación.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del producto en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R5, R5.1, R5.2, R5.3, R5.4, R5.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El producto fue gestionado.
Prioridad	Crítico

Tabla 2.10: Descripción textual del caso de uso «gestionar producto»

Caso de uso del sistema	Gestionar implementador
Actores del sistema	Registrador (inicia)
Propósito	Conocer mediante la aplicación que implementador es el encargado de cada

	proyecto.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del implementador en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R6, R6.1, R6.2, R6.3, R6.4, R6.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El implementador fue gestionado.
Prioridad	Crítico

Tabla 2.11: Descripción textual del caso de uso «gestionar implementador»

Caso de uso del sistema	Gestionar avance de proyecto
Actores del sistema	Registrador (inicia)
Propósito	Conocer el estado de de un proyecto.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del avance de proyecto en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R8, R8.1, R8.2, R8.3, R8.4, R8.5
Precondiciones	El registrador debe haberse autenticado.
Poscondiciones	El avance del proyecto fue gestionado.
Prioridad	Crítico

Tabla 2.12: Descripción textual del caso de uso «gestionar avance de proyecto»

Caso de uso del sistema	Gestionar usuario
Actores del sistema	Administrador (inicia)
Propósito	Mantener el rol de cada usuario ya que esto se deriva en seguridad para el sistema.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del usuario en el sistema, donde luego podrá realizar otras funciones como modificar, listar,

	eliminar o descargar.
Referencia	R9, R9.1, R9.2, R9.3, R9.4, R9.5
Precondiciones	El administrador debe haberse autenticado y poseer los permisos adecuados.
Poscondiciones	El usuario fue gestionado.
Prioridad	Crítico

Tabla 2.13: Descripción textual del caso de uso «gestionar usuario»

Caso de uso del sistema	Gestionar municipio
Actores del sistema	Administrador (inicia)
Propósito	Que el sistema disponga de todos los municipios actualizados para la provincia de La Habana.
Resumen	El caso de uso inicia cuando el registrador inserta los datos del municipio en el sistema, donde luego podrá realizar otras funciones como modificar, listar, eliminar o descargar.
Referencia	R10, R10.1, R10.2, R10.3, R10.4, R10.5
Precondiciones	El administrador debe haberse autenticado y poseer los permisos adecuados.
Poscondiciones	El municipio fue gestionado.
Prioridad	Crítico

Tabla 2.14: Descripción textual del caso de uso «gestionar municipio»

Caso de uso del sistema	Descargar reporte de implementación
Actores del sistema	Usuario (inicia)
Propósito	Dar a conocer al implementador su reporte mensual
Resumen	El caso de uso inicia con la descarga del reporte de implementación y concluye junto a la descarga del reporte.
Referencia	R11
Precondiciones	El usuario debe haberse autenticado.
Poscondiciones	La descarga fue realizada con éxito.

Prioridad	Crítico
------------------	---------

Tabla 2.15: Descripción textual del caso de uso «descargar reporte de implantación»

Caso de uso del sistema	Descargar reporte de seguimiento
Actores del sistema	Usuario (inicia)
Propósito	Dar a conocer un reporte mensual de seguimiento para cada proyecto.
Resumen	El caso de uso inicia con la descarga del reporte de seguimiento y concluye junto a la descarga del reporte.
Referencia	R12
Precondiciones	El usuario debe haberse autenticado.
Poscondiciones	La descarga fue realizada con éxito.
Prioridad	Crítico

Tabla 2.16: Descripción textual del caso de uso «descargar reporte de seguimiento»

Anexo 2: Diagramas de clases del diseño

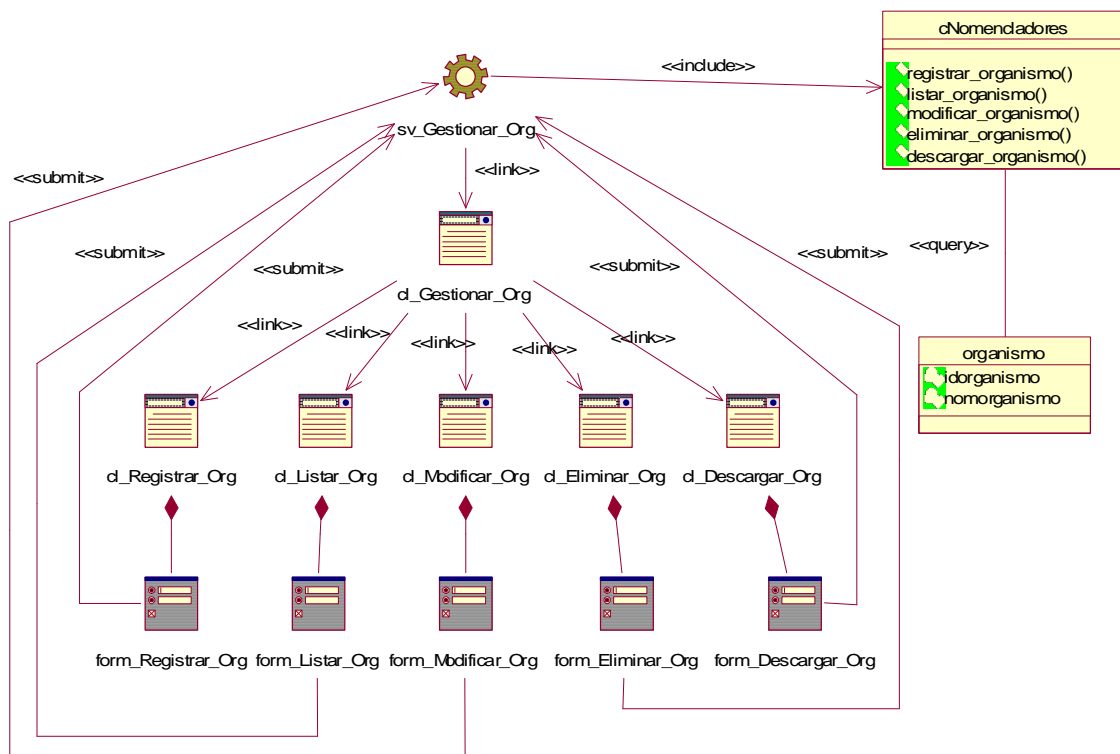


Figura 3.9. Diagrama de clases del diseño «Gestionar Organismo»

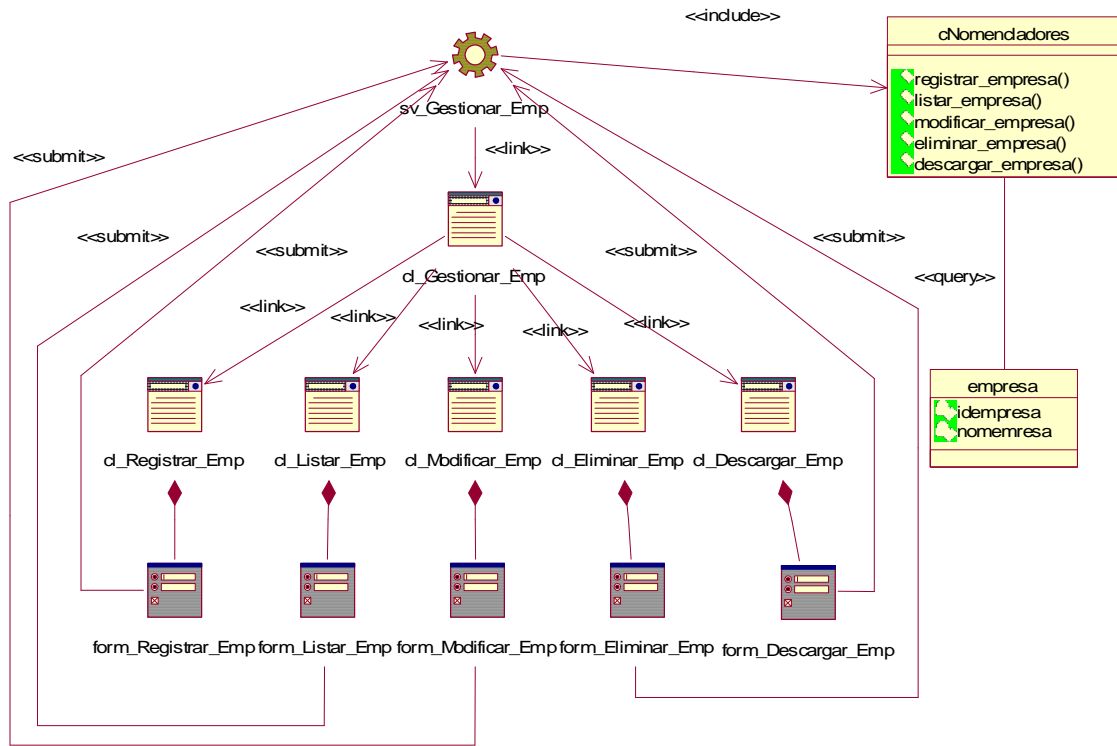


Figura 3.10. Diagrama de clases del diseño «Gestionar Empresa»

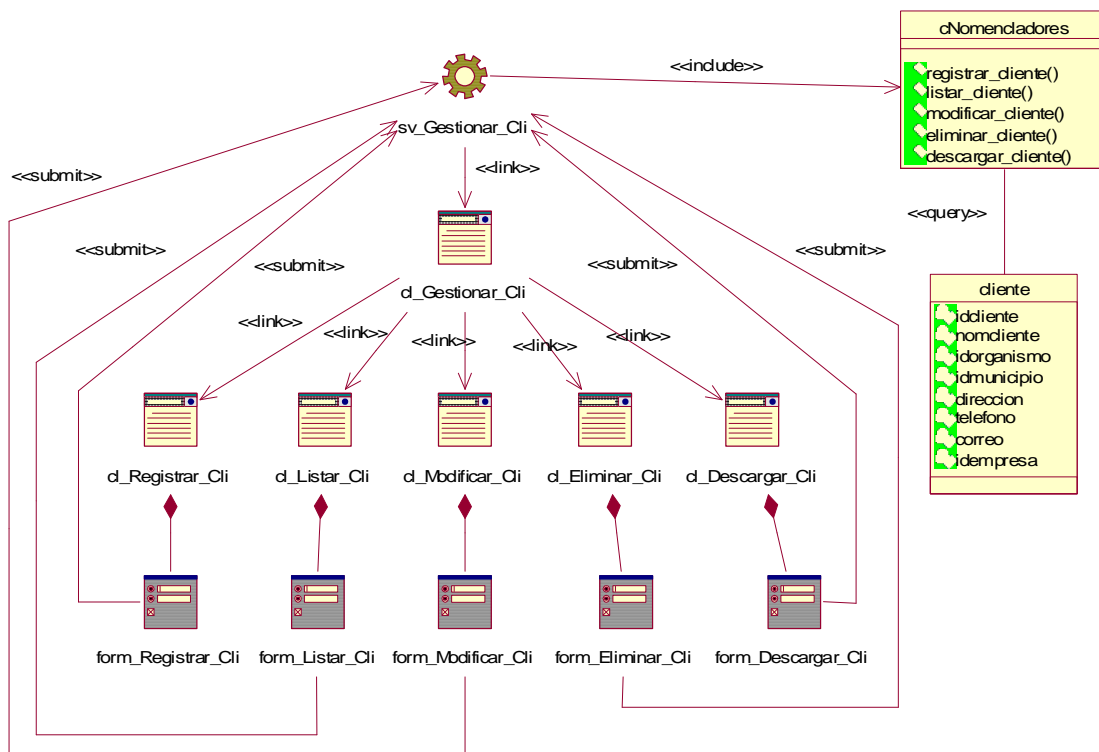


Figura 3.11. Diagrama de clases del diseño «Gestionar Cliente»

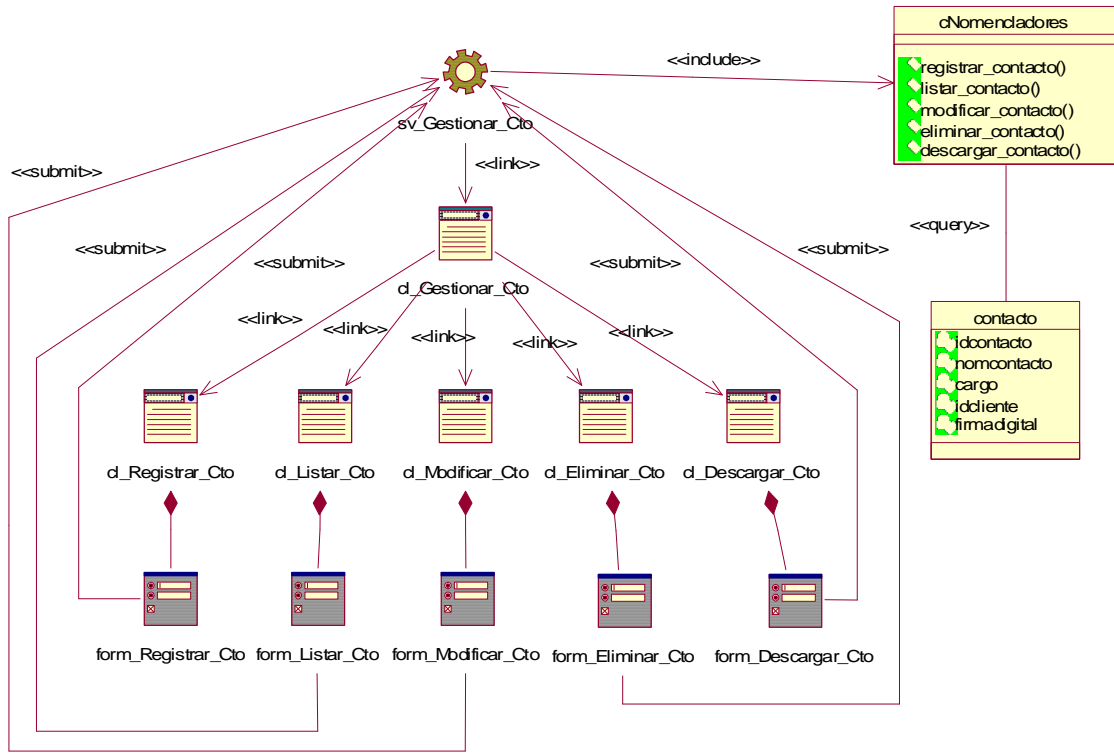


Figura 3.12. Diagrama de clases del diseño «Gestionar Contacto»

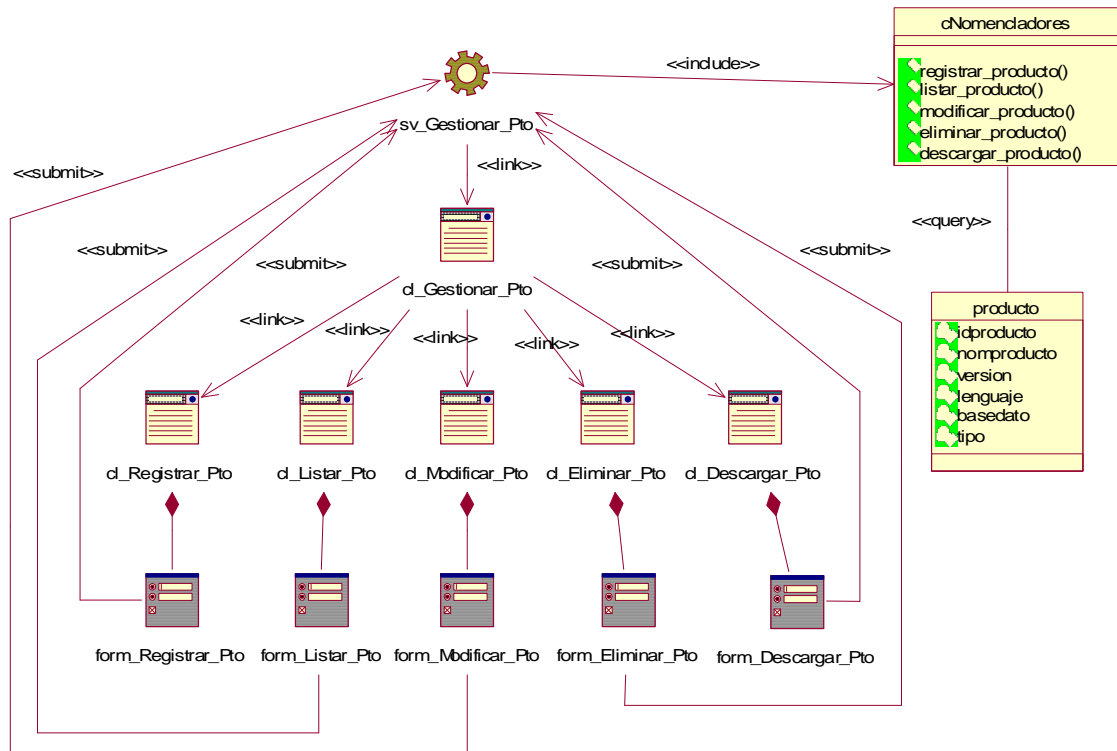


Figura 3.13. Diagrama de clases del diseño «Gestionar Producto»

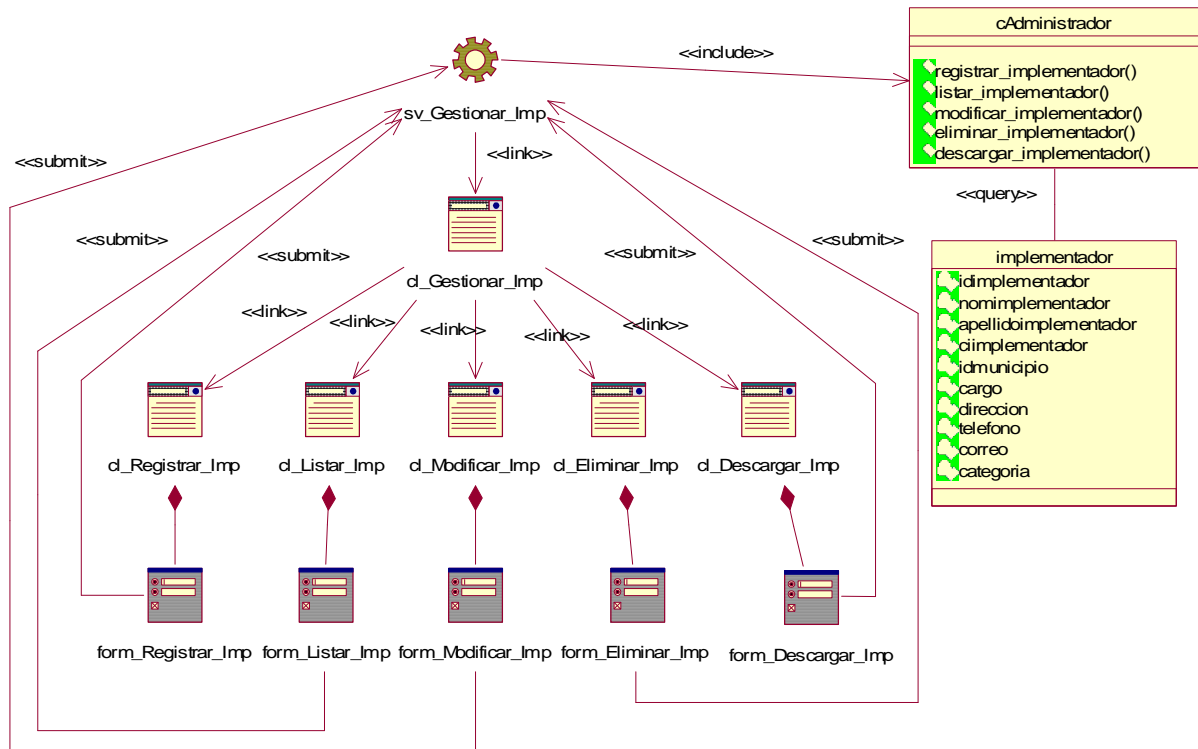


Figura 3.14. Diagrama de clases del diseño «Gestionar Implementador»

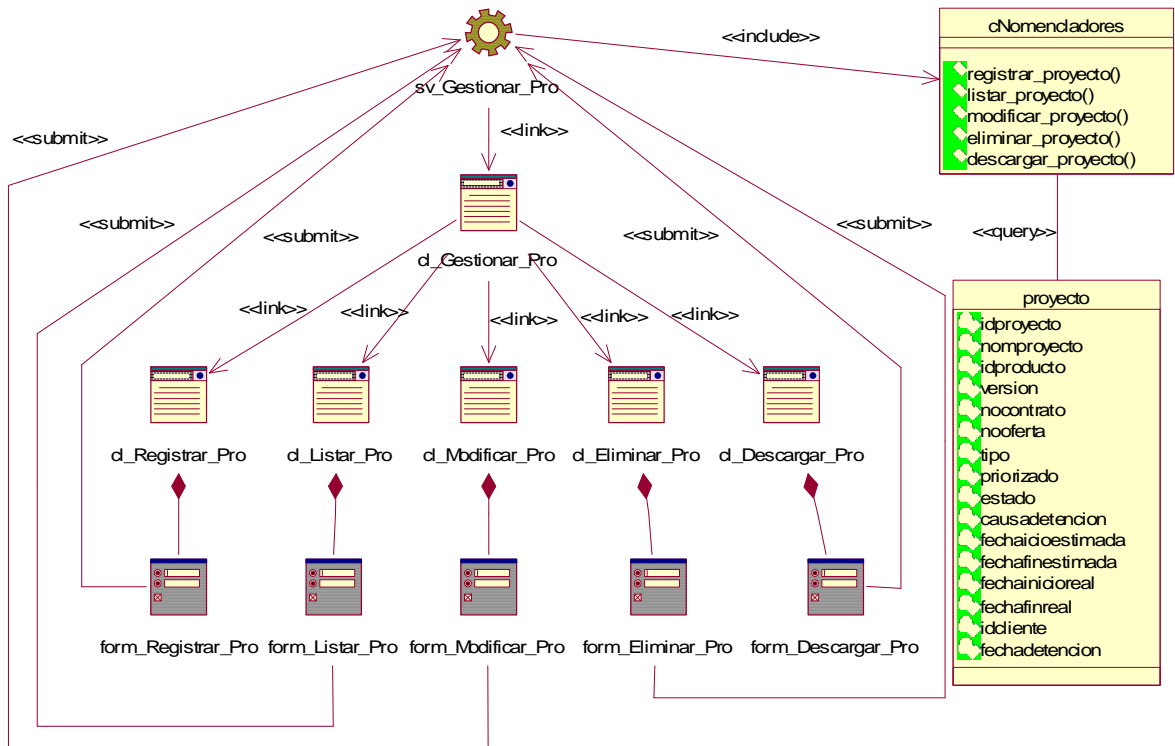


Figura 3.15. Diagrama de clases del diseño «Gestionar Proyecto»

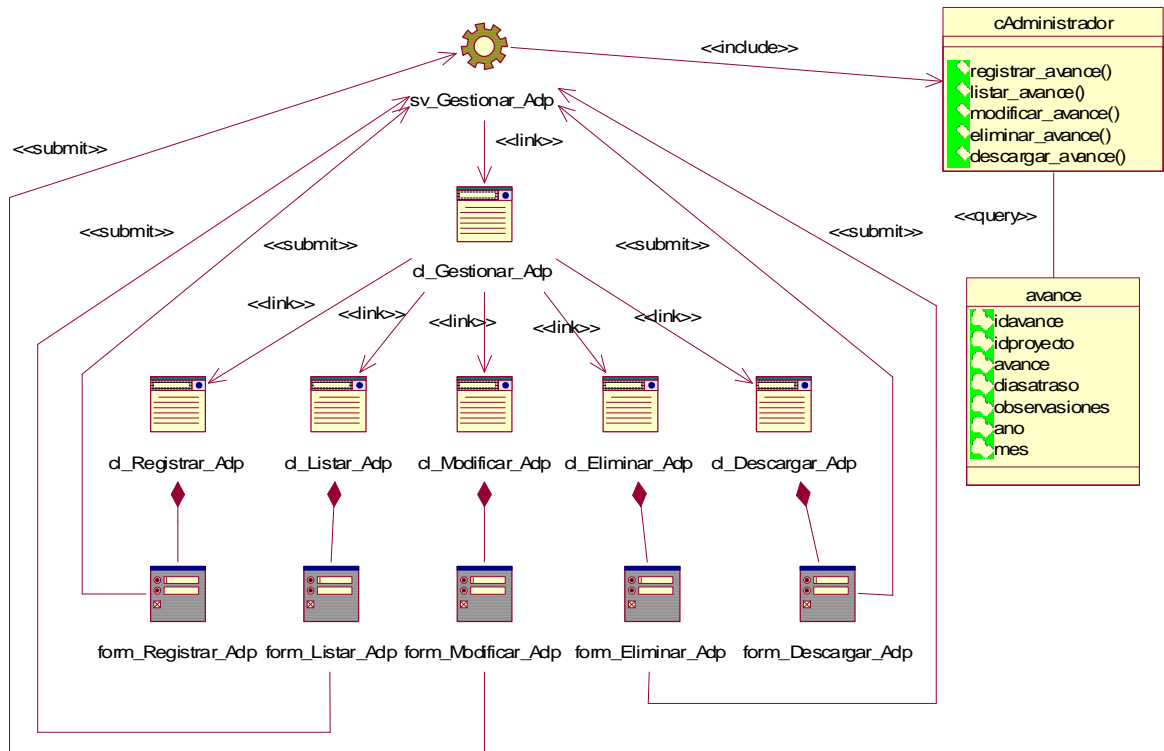


Figura 3.16. Diagrama de clases del diseño «Gestionar Avance del Proyecto»

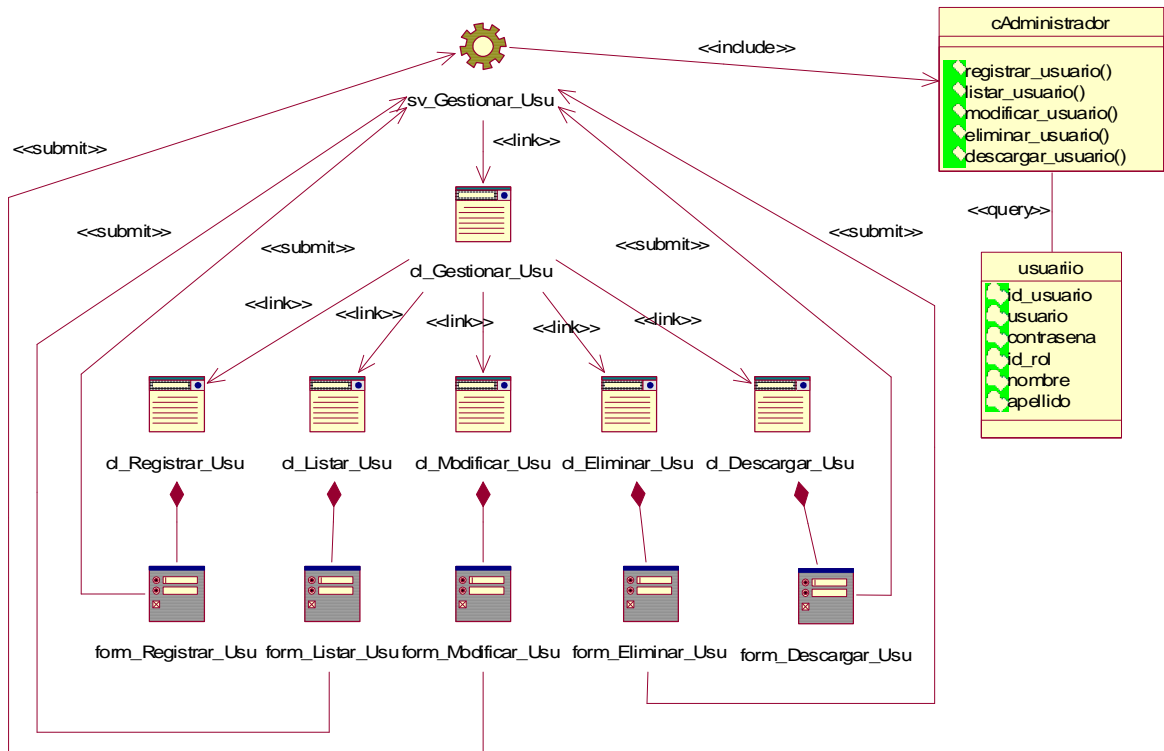


Figura 3.17. Diagrama de clases del diseño «Gestionar Usuario»

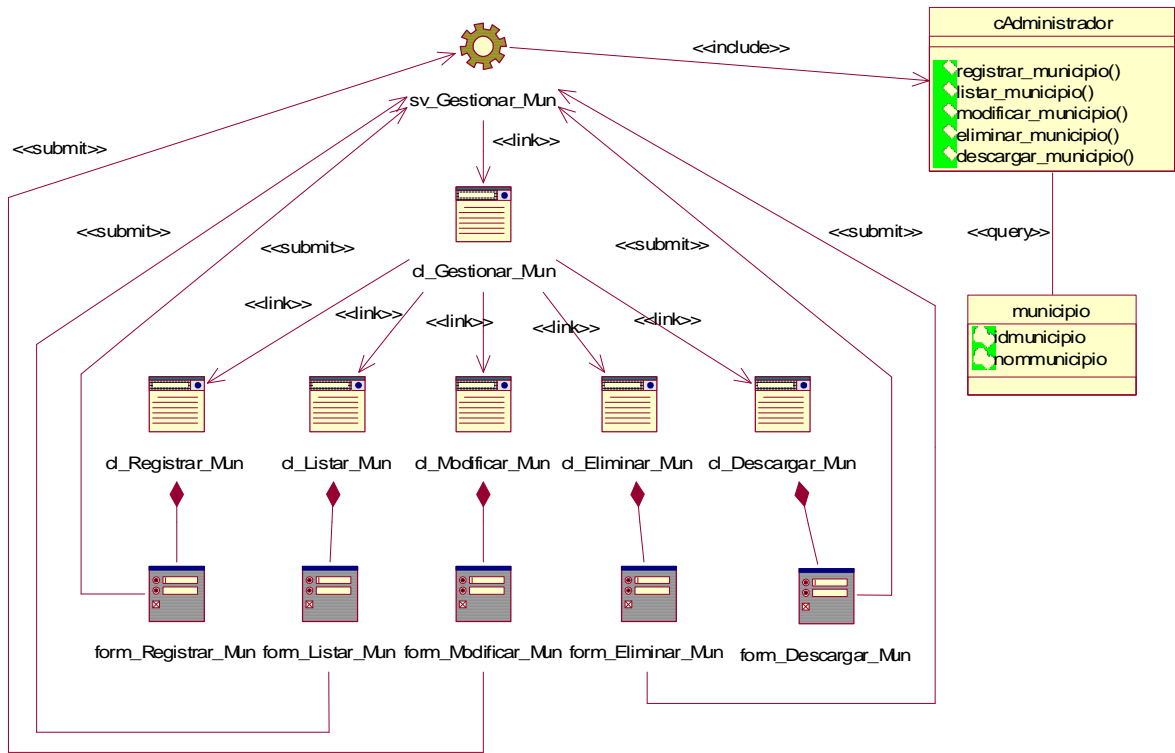


Figura 3.18. Diagrama de clases del diseño «Gestionar Municipio»

Anexo 3: Descripción textual de las clases persistentes

Nombre de la clase	avance	
Propósito	Guardar la información concerniente al avance de proyectos de implementación	
Atributo Descripción	Atributo	Descripción
Idproyecto	Identificador de la clase	
avance	Descripción de avance para un proyecto	
diasatraso	Tiempo de atraso para un proyecto	
observaciones	Descripción	
ano	Año en el que se registro el avance	
mes	Mes en el que se registro el avance	

Nombre de la clase	empresa	
Propósito	Guardar la información concerniente a cada empresa en el sistema	
Atributo Descripción	Atributo	Descripción
idempresa	Identificador de la clase	
nomempresa	Nombre de la empresa	

Nombre de la clase	proyecto	
Propósito	Guardar la información concerniente a cada proyecto en el sistema	
Atributo Descripción	Atributo	Descripción
idproyecto	Identificador de la clase	
nomproyecto	Nombre del proyecto	
idproducto	Identificador del producto	
version	Versión del proyecto	
nocontrato	Número de contrato para el proyecto	
nooferta	Número de oferta para el proyecto	
tipo	Tipo de proyecto	
priorizado	Indica si es priorizado o no	
estado	Estado del proyecto	

causadetencion	Descripción de la causa para la detención
fechainicioestimada	Fecha de inicio estimada del proyecto
fechafinestimada	Fecha fin estimada del proyecto
fechainicioreal	Fecha de inicio real del proyecto
fechafinreal	Fecha fin real del proyecto
idcliente	Identificador del cliente
fechadetencion	Fecha de la detención del proyecto

Nombre de la clase	producto	
Propósito	Guardar la información concerniente a cada producto en el sistema	
Atributo Descripción	Atributo	Descripción
	idproducto	Identificador de la clase
	nomproducto	Nombre del producto
	version	Versión del producto
	lenguaje	Lenguaje del producto
	basedato	Base de dato del producto
	tipo	Tipo de producto

Nombre de la clase	cliente	
Propósito	Guardar la información concerniente a cada cliente en el sistema	
Atributo Descripción	Atributo	Descripción
	idcliente	Identificador de la clase
	nomcliente	Nombre del cliente
	idorganismo	Identificador del organismo al que pertenece
	idmunicipio	Identificador del municipio del cliente
	direccion	Dirección del cliente
	telefono	Teléfono del cliente
	correo	Correo electrónico del cliente
	idempresa	Identificador de la empresa a la que pertenece

Nombre de la clase	municipio	
Propósito	Almacenar todos los municipio pertenecientes a la provincia en el sistema	
Atributo Descripción	Atributo Descripción	
idmunicipio	Identificador de la clase	
nommunicipio	Nombre del municipio	

Nombre de la clase	contacto	
Propósito	Guardar la información concerniente a cada contacto en el sistema	
Atributo Descripción	Atributo Descripción	
idcontacto	Identificador de la clase	
nomcontacto	Nombre del contacto	
cargo	Cargo del contacto	
idcliente	Identificador del cliente	
firmedigital	Firma digital para un contacto	

Nombre de la clase	rol	
Propósito	Registrar el rol de cada usuario que pertenece al sistema	
Atributo Descripción	Atributo Descripción	
id_rol	Identificador de la clase	
nombre	Nombre del rol	

Nombre de la clase	organismo	
Propósito	Guardar la información concerniente a cada organismo en el sistema	
Atributo Descripción	Atributo Descripción	
idorganismo	Identificador de la clase	
nomorganismo	Nombre del organismo	

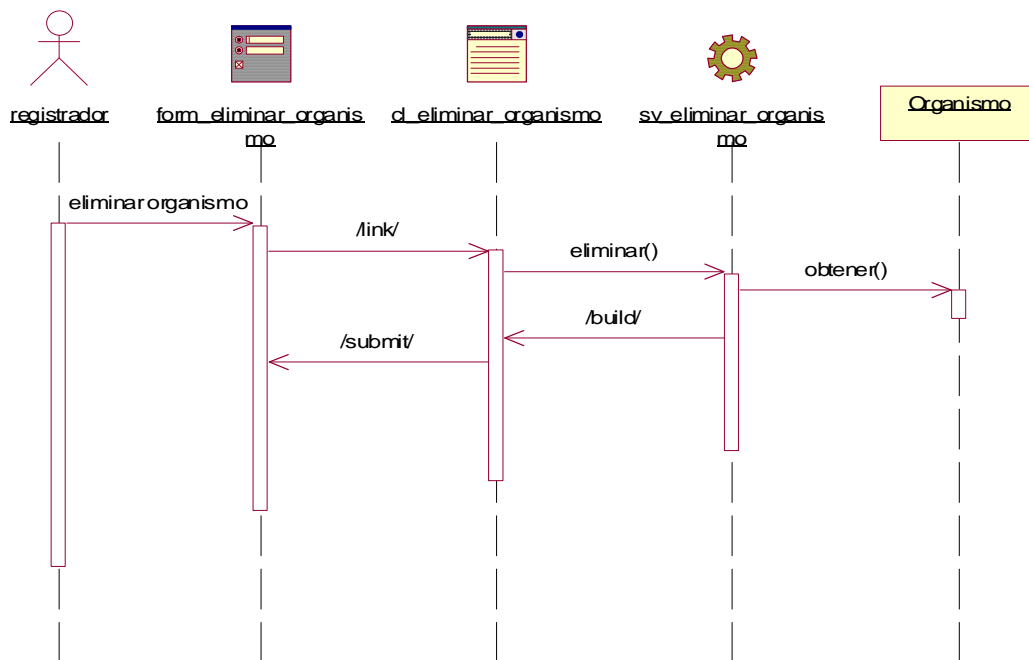
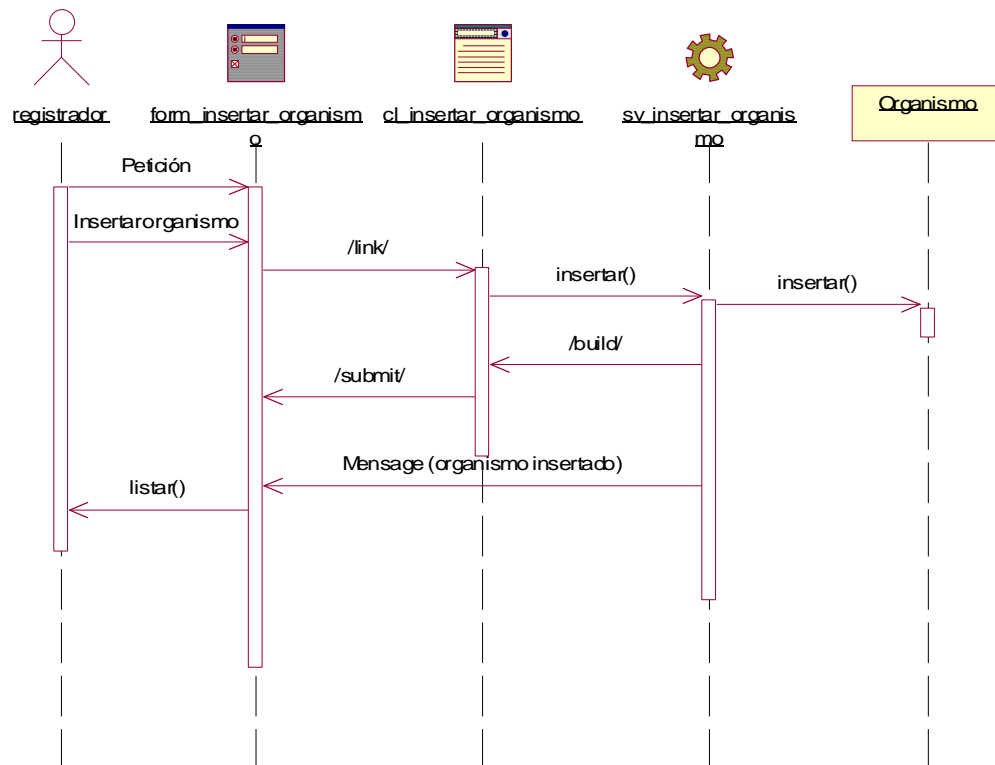
Nombre de la clase	implementador	
Propósito	Guardar la información concerniente a cada implementador en el sistema	

Atributo Descripción	Atributo Descripción
idimplementador	Identificador de la clase
nomimplementador	Nombre implementador
apellidoinplementador	Apellido implementador
ciimplementador	Número del carnet de identidad del implementador
idmunicipio	Identifica al municipio a que pertenece
cargo	Cargo implementador
direccion	Dirección implementador
telefono	Teléfono implementador
correo	Correo electrónico del implementador
categoria	Categoría del implementador

Nombre de la clase	Implementador_proyecto
Propósito	Guardar la información concerniente a cada proyecto en el sistema
Atributo Descripción	Atributo Descripción
idimpleproyecto	Identificador de la clase
idimplementador	Identificador del implementador
idproyecto	Identificador del proyecto

Nombre de la clase	usuario
Propósito	Registra los usuario que pertenecen al sistema
Atributo Descripción	Atributo Descripción
idusuario	Identificador de la clase
usuario	Nombre del usuario
contrasena	Contraseña del usuario
id_rol	Identifica el tipo de usuario
nombre	Nombre real del usuario
apellido	Apellido real del usuario

Anexo 4: Diagramas de secuencia



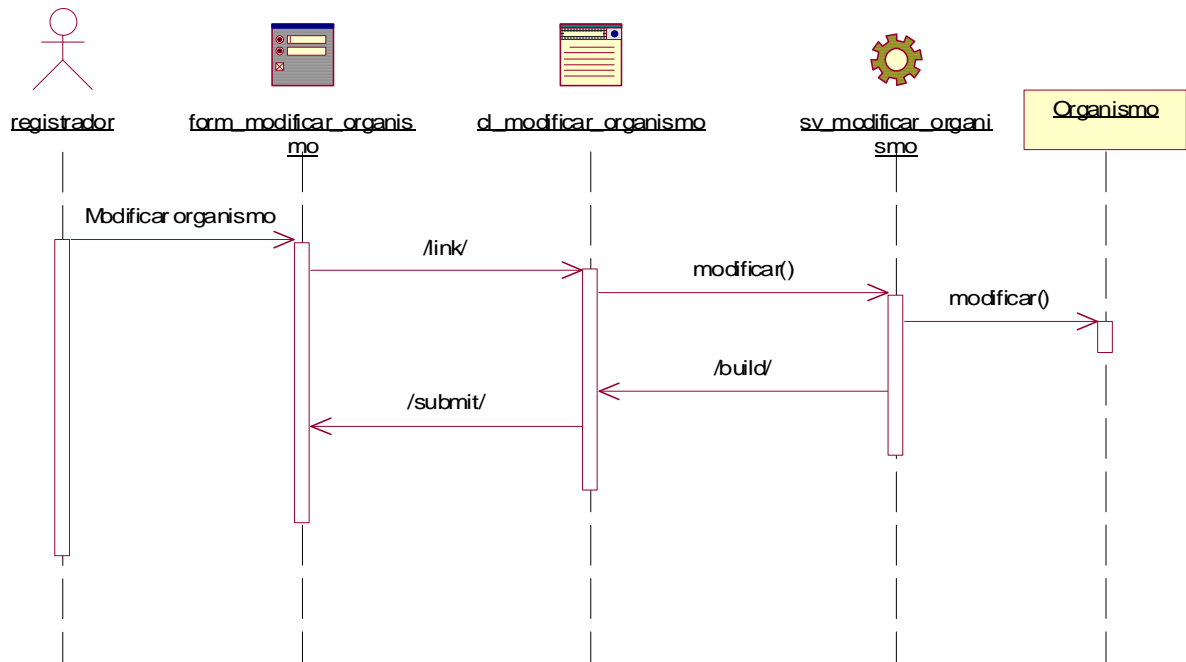
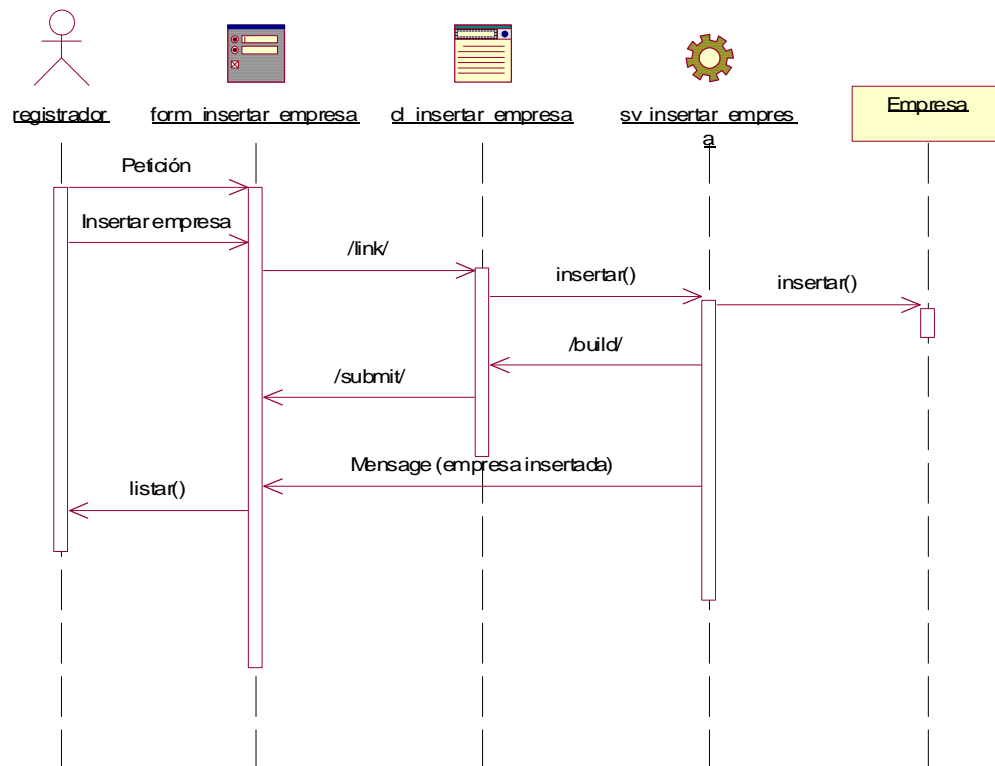


Figura 3.19. Diagrama de secuencia «gestionar organismo»



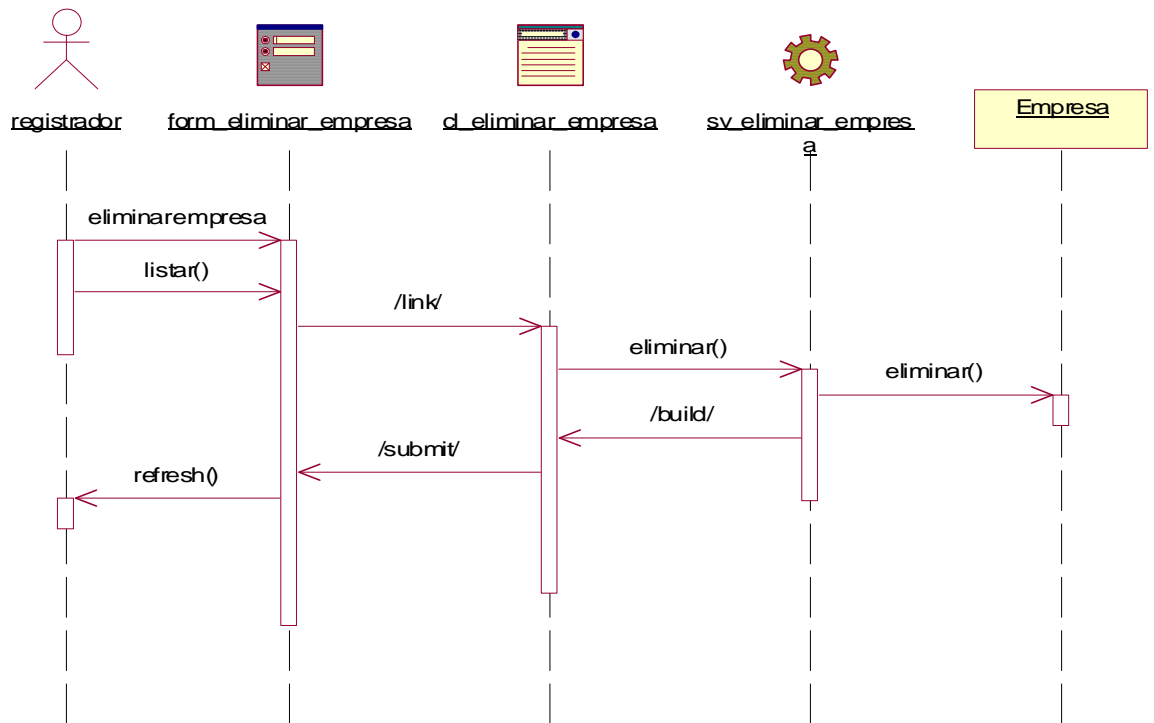
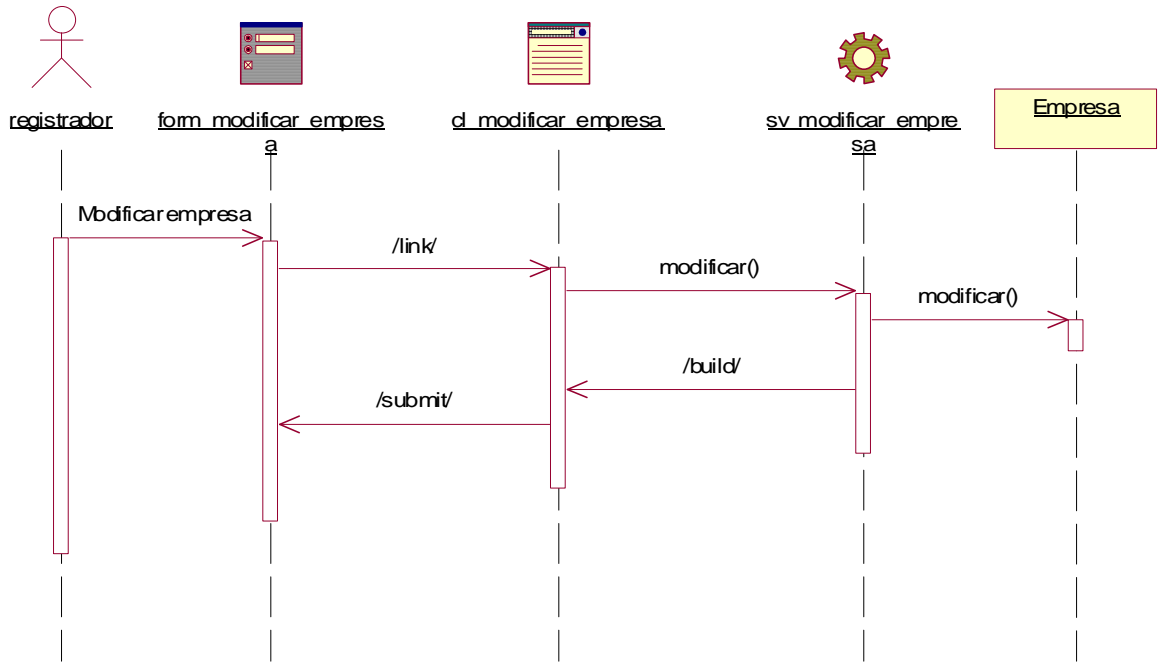
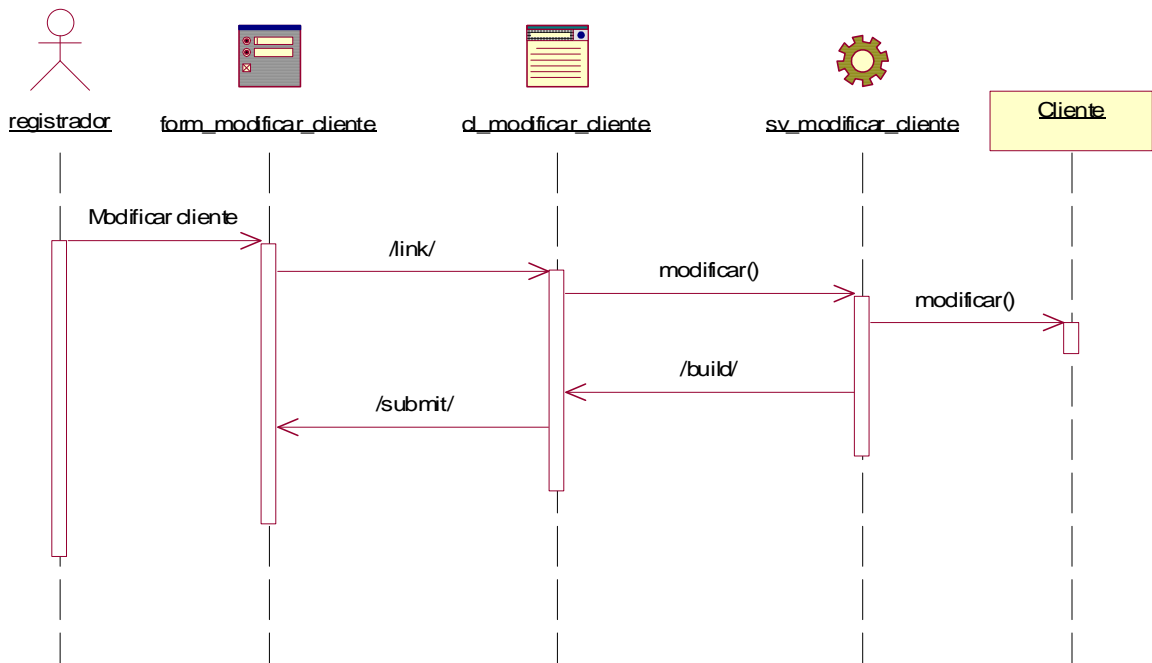
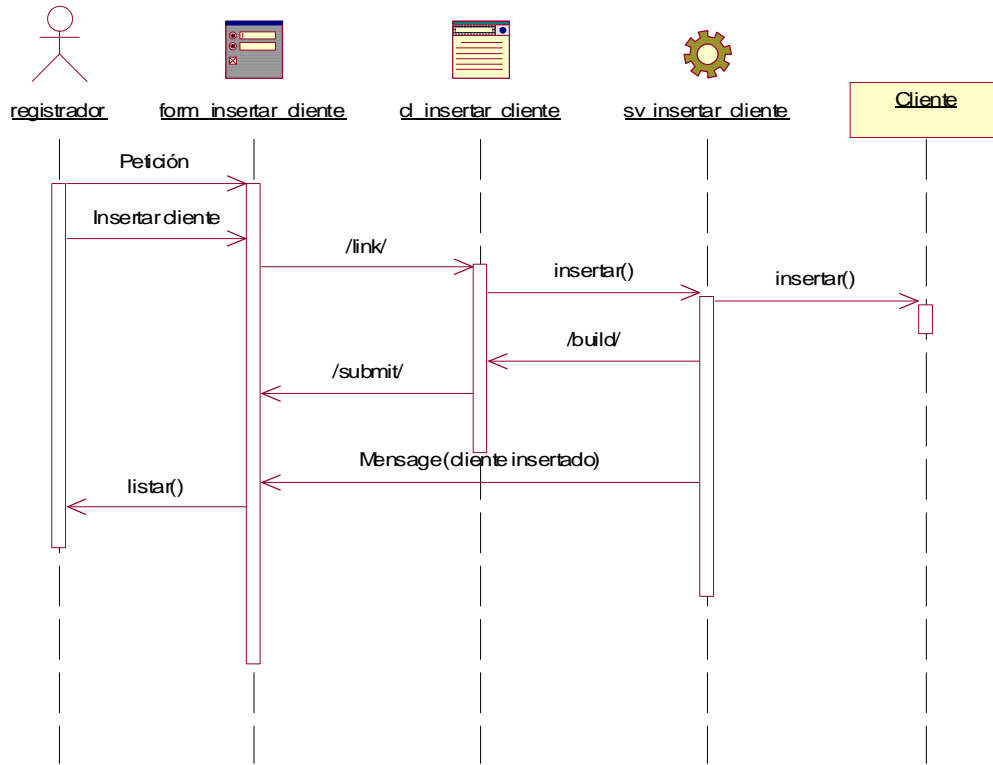


Figura 3.20. Diagrama de secuencia «gestionar empresa»



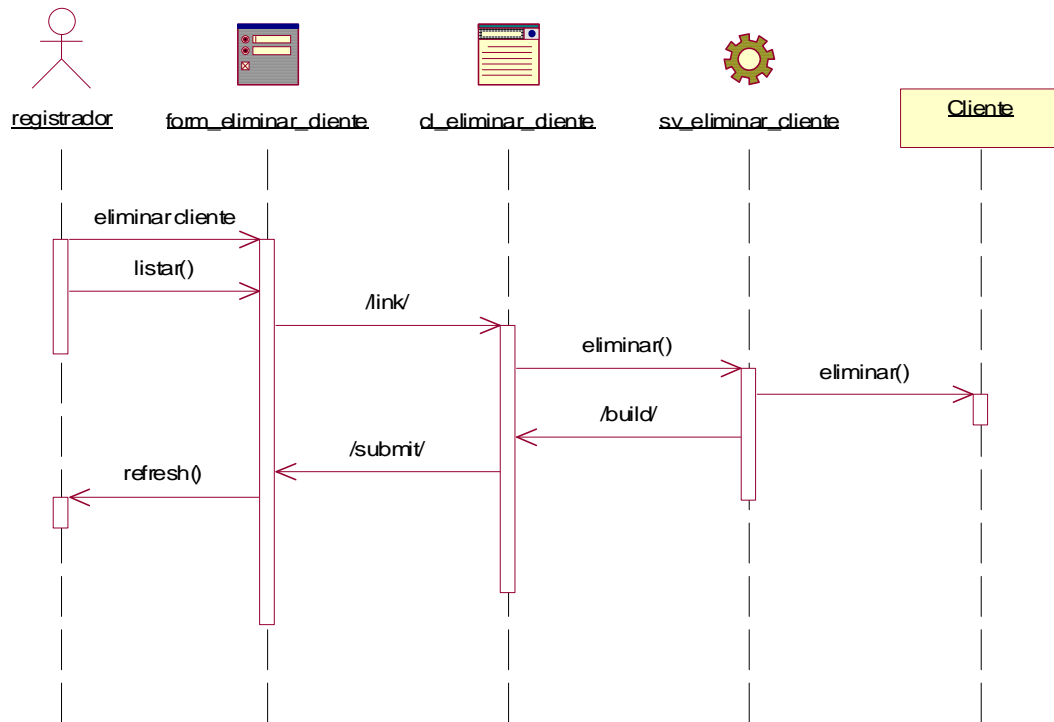
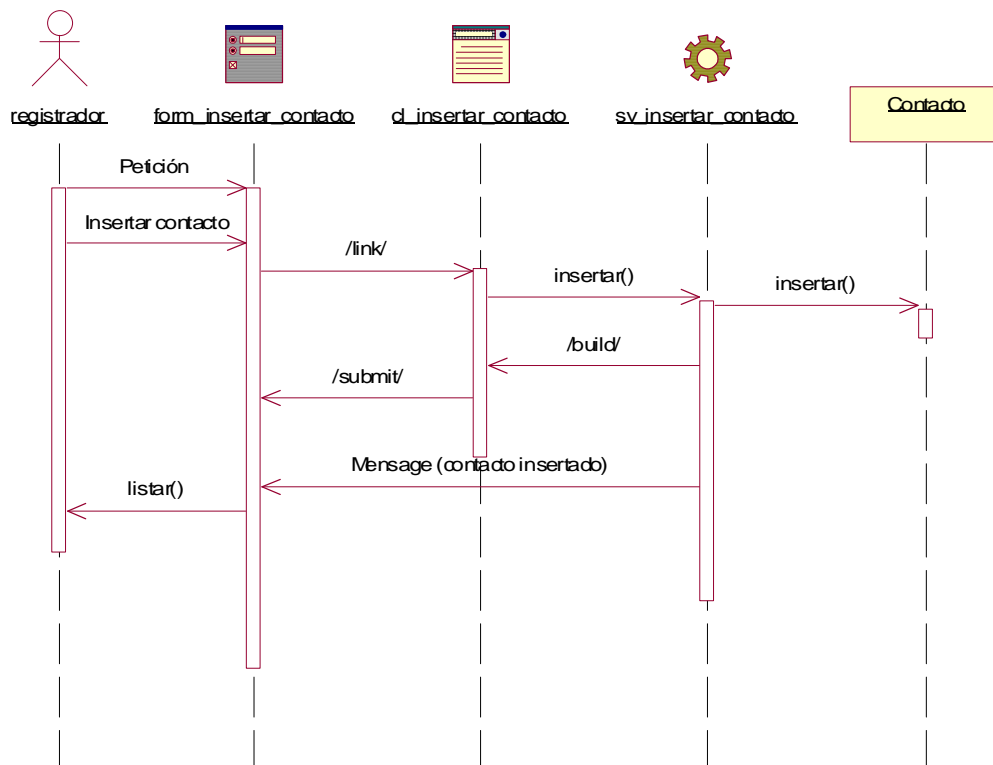


Figura 3.21. Diagrama de secuencia «gestionar cliente»



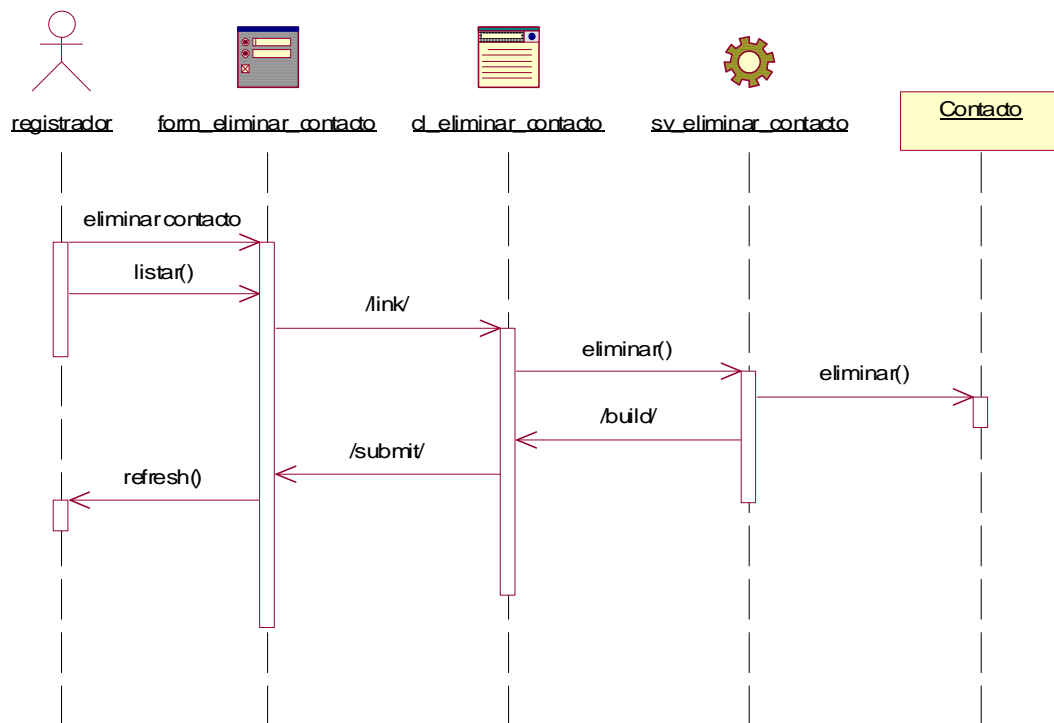
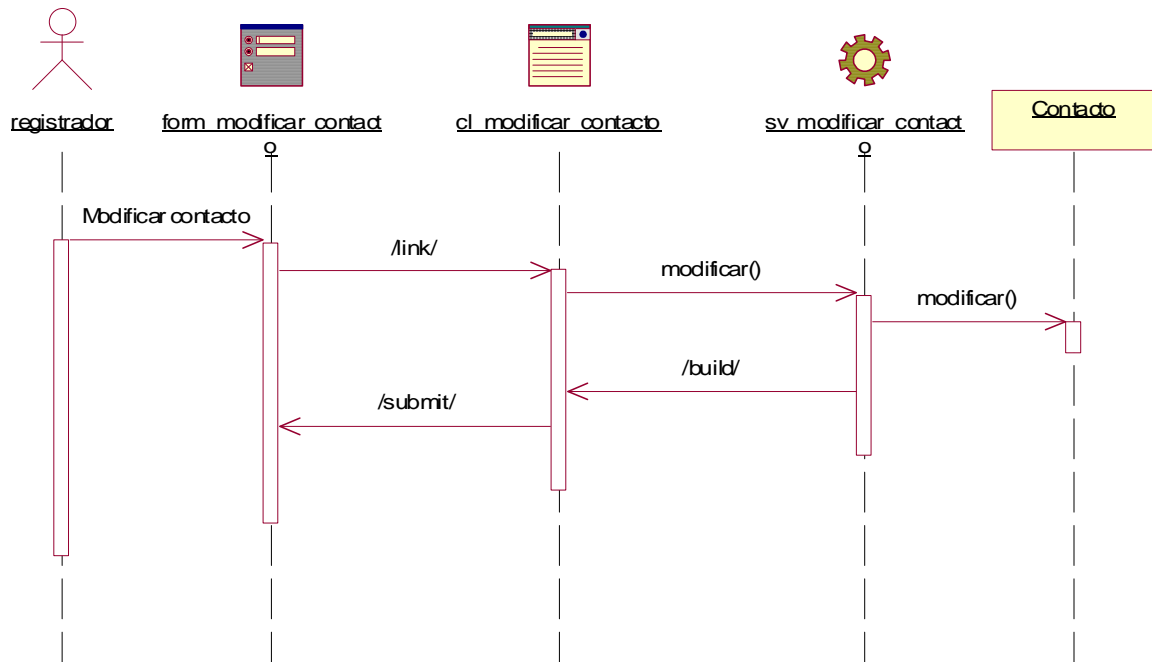
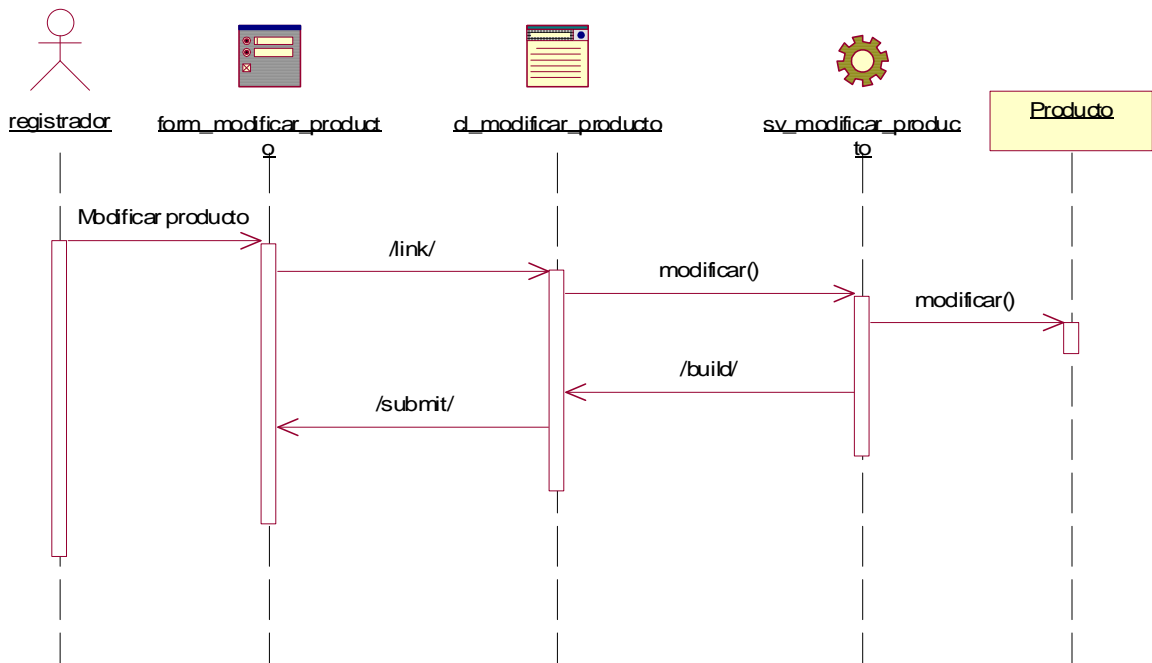
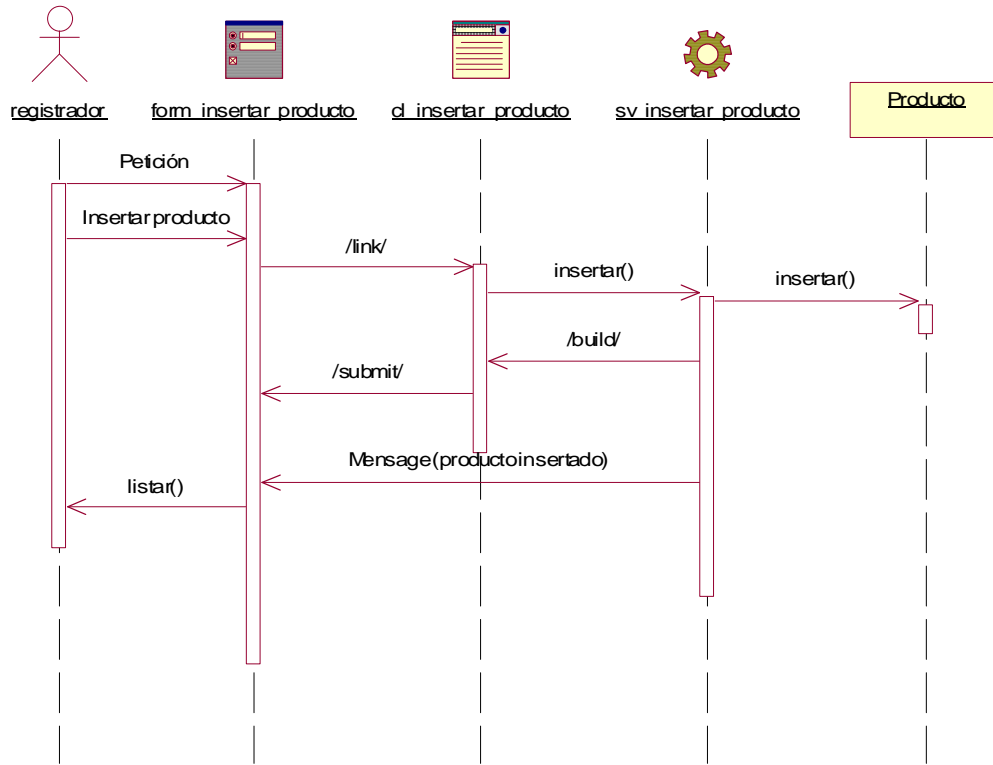


Figura 3.22. Diagrama de secuencia «gestionar contacto»



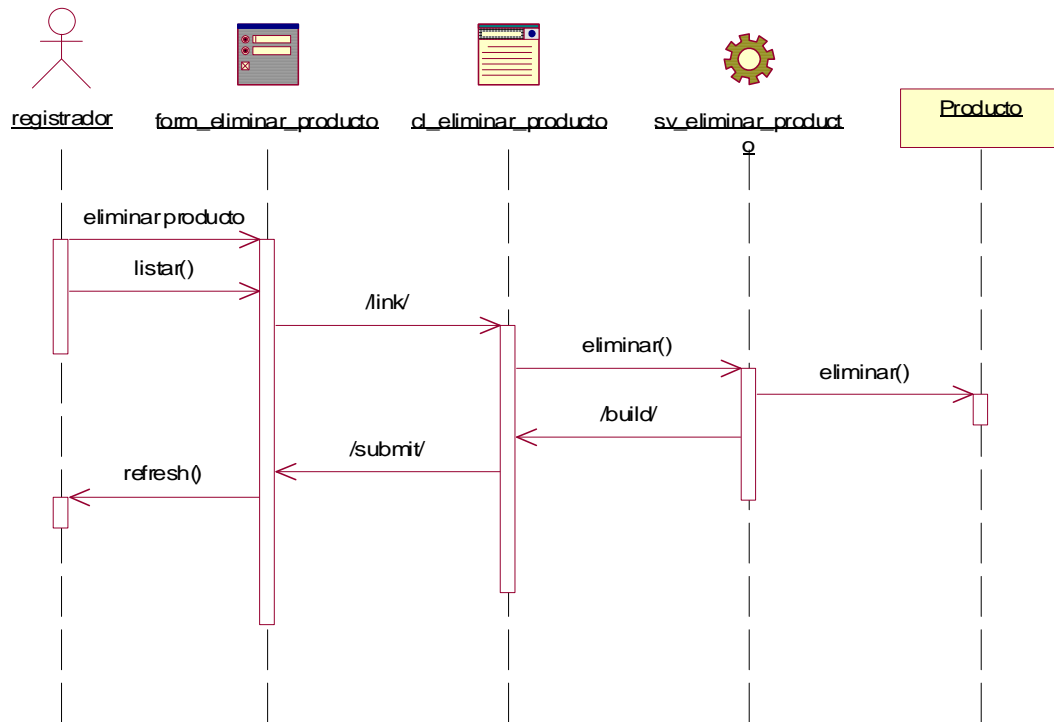
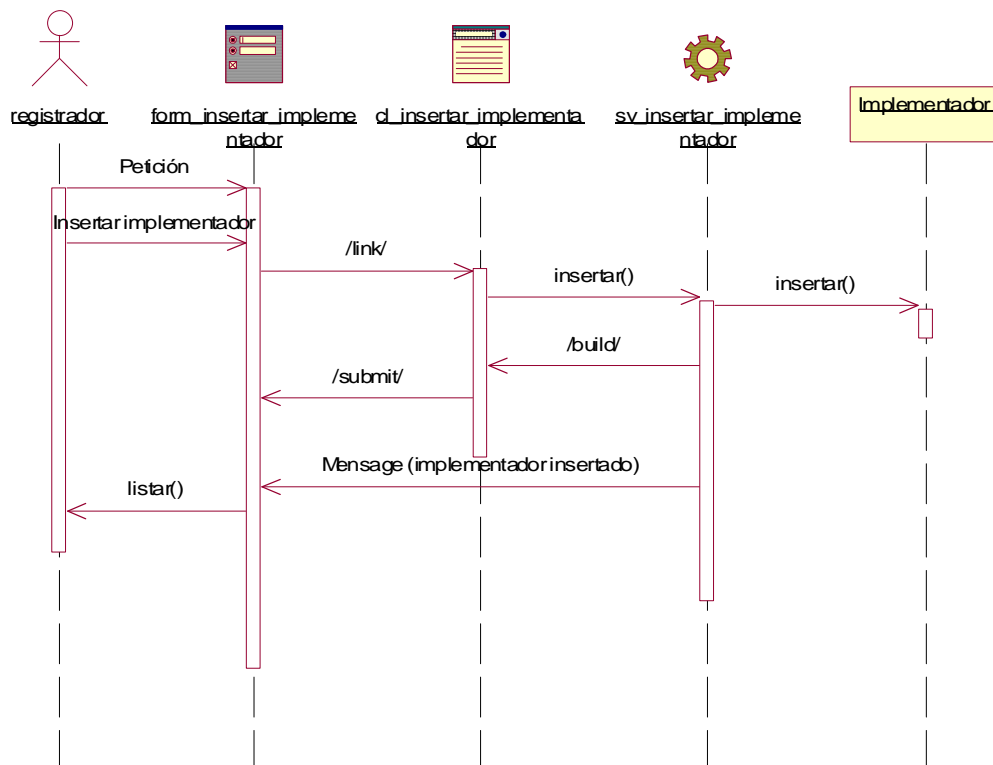


Figura 3.23. Diagrama de secuencia «gestionar producto»



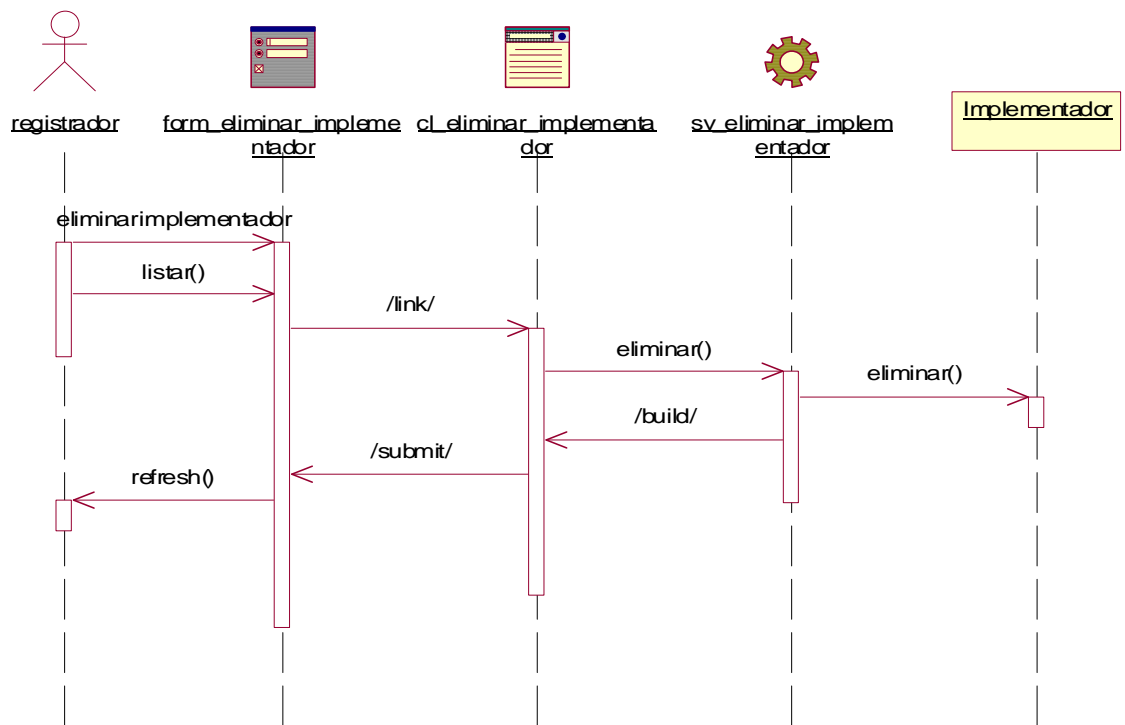
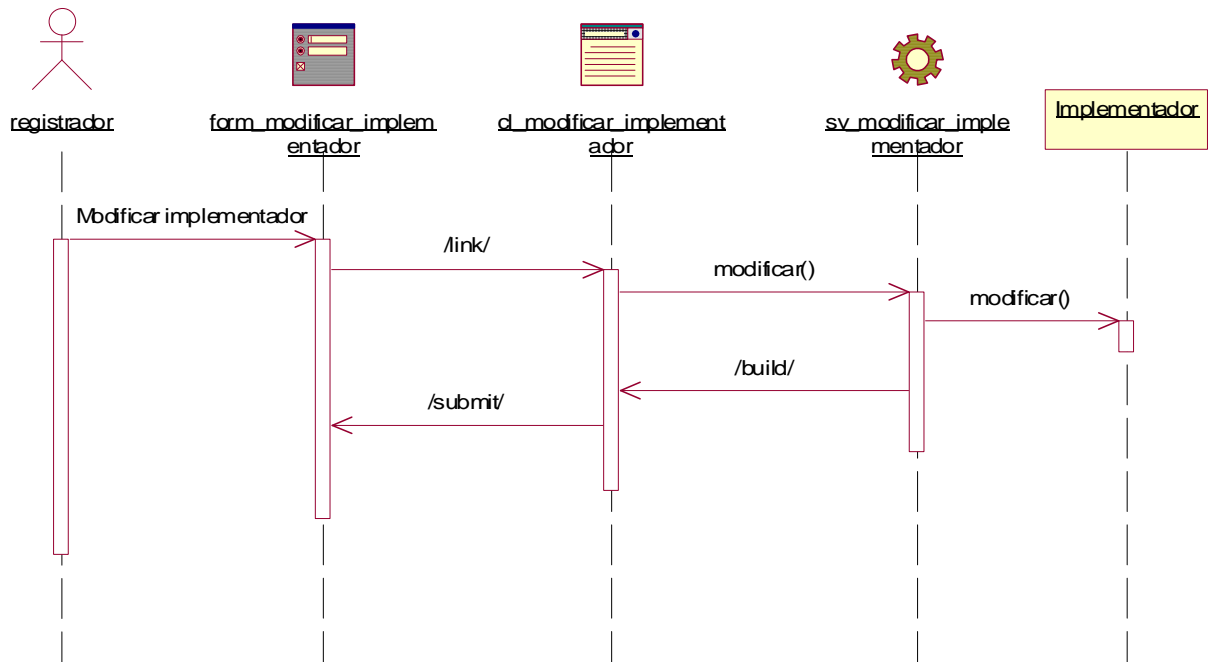
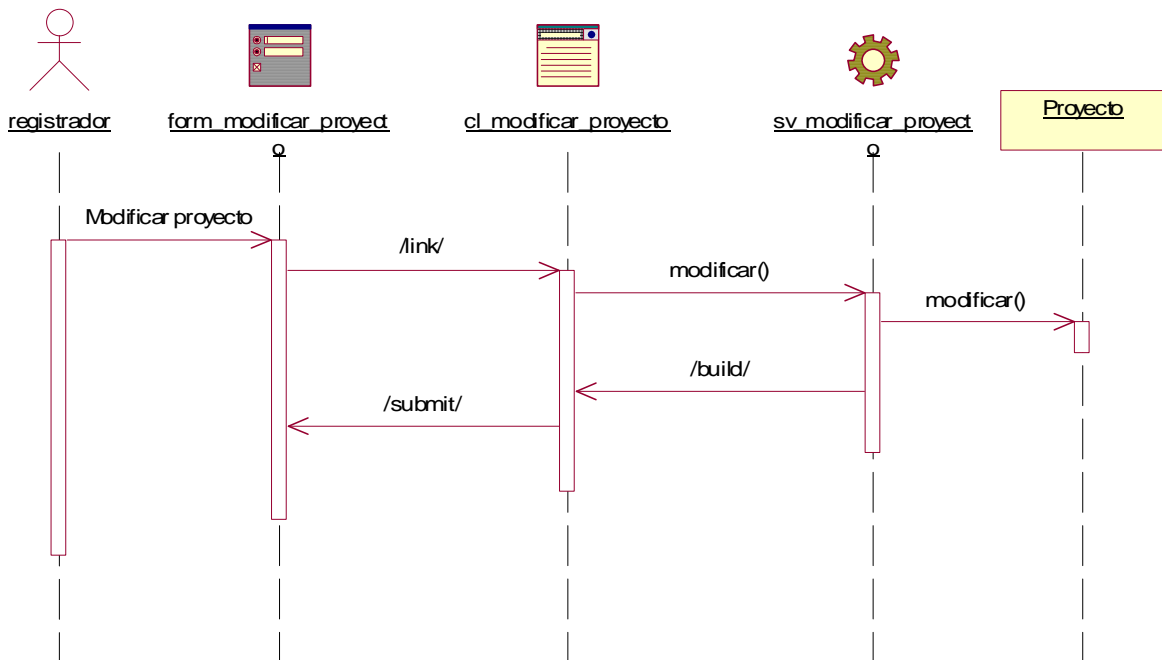
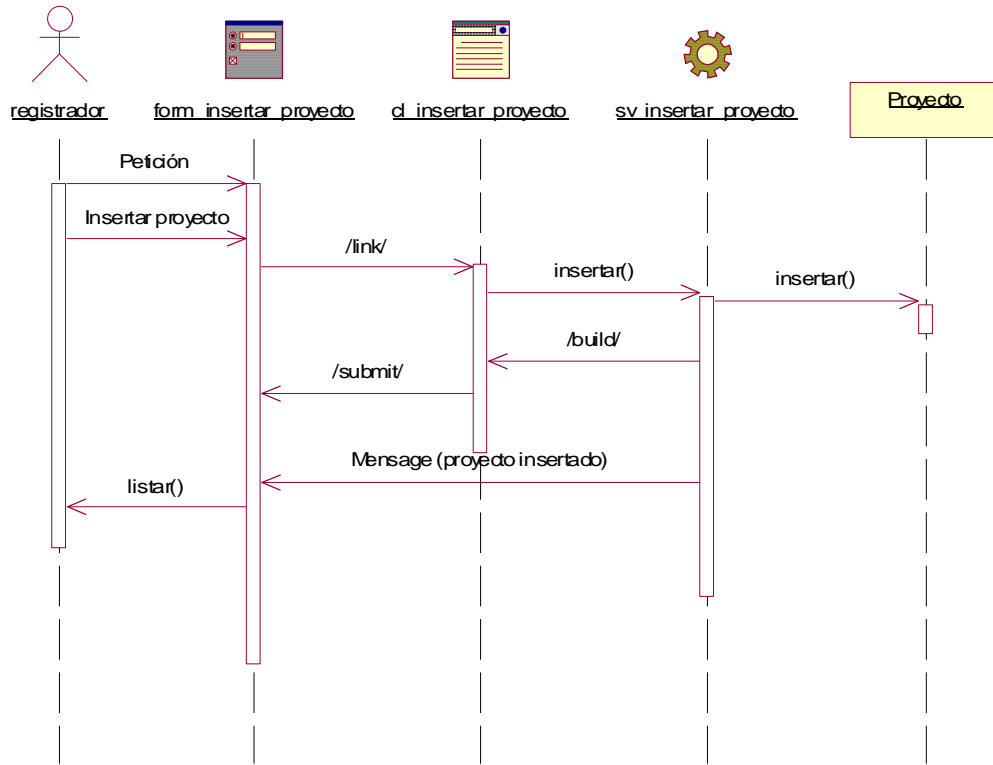


Figura 3.24. Diagrama de secuencia «gestionar implementador»



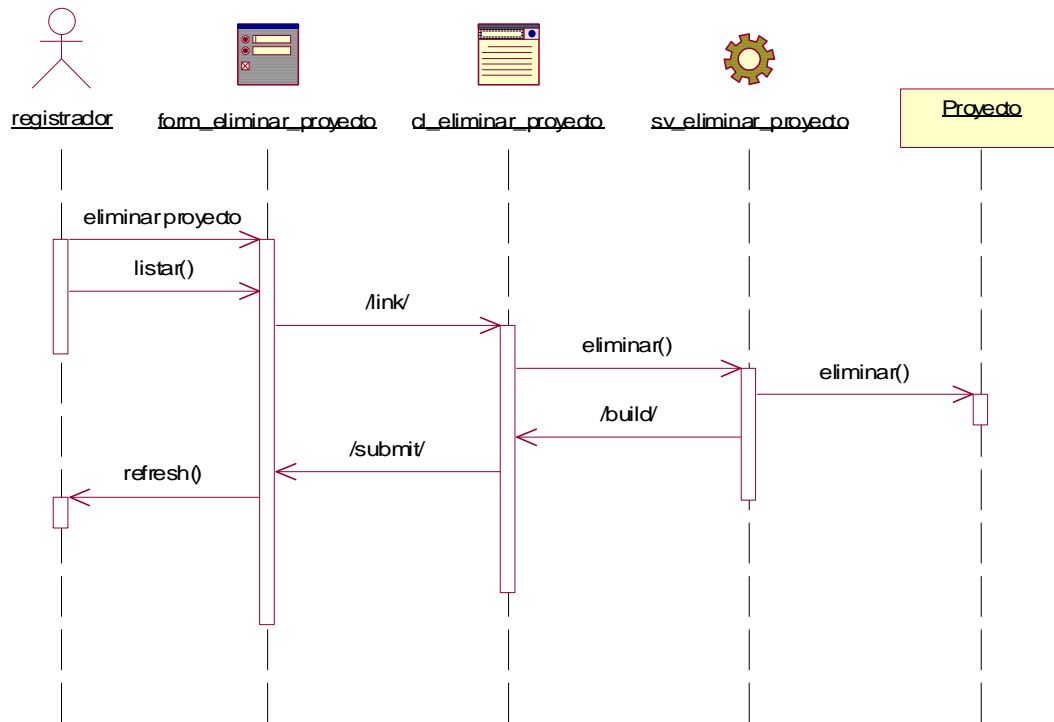
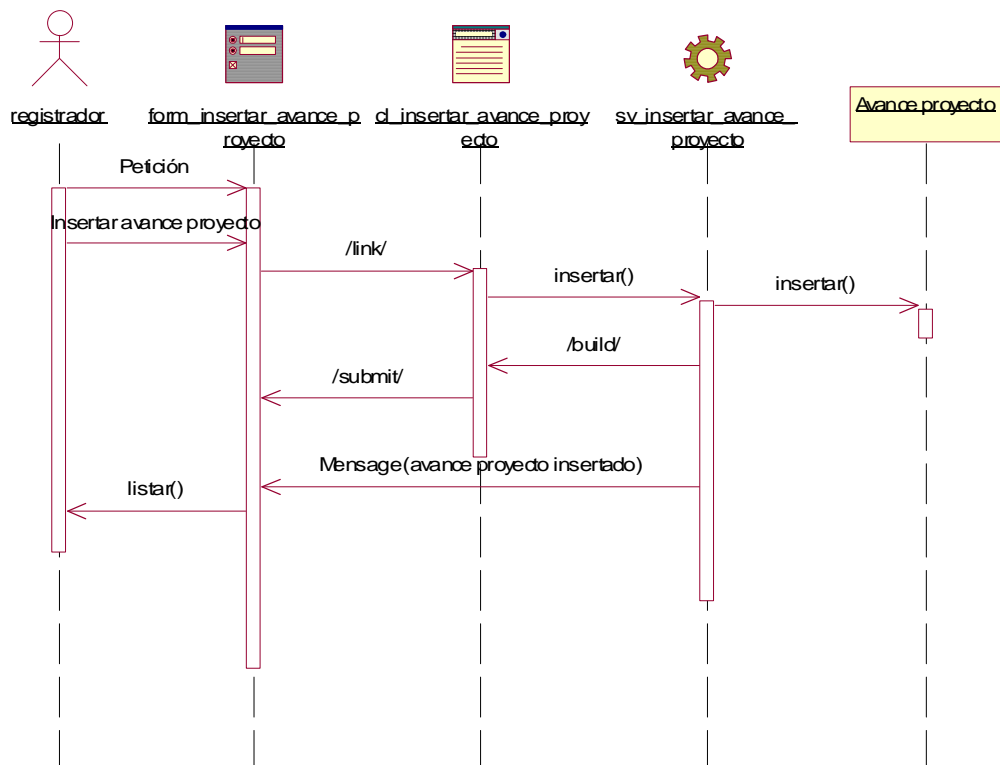


Figura 3.25. Diagrama de secuencia «gestionar proyecto»



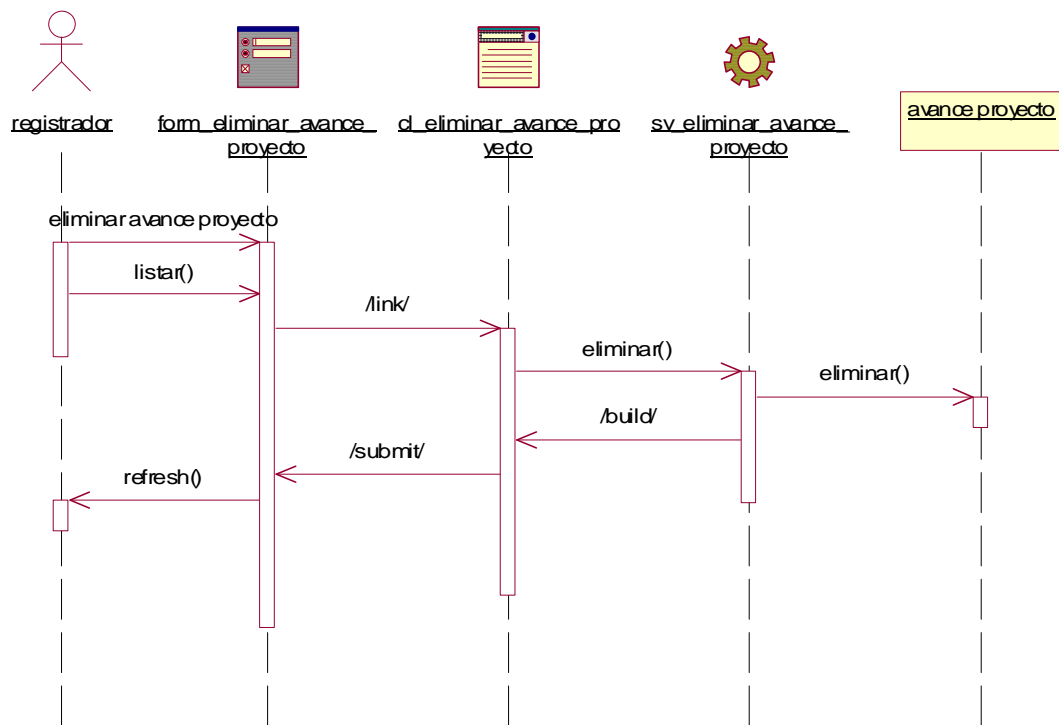
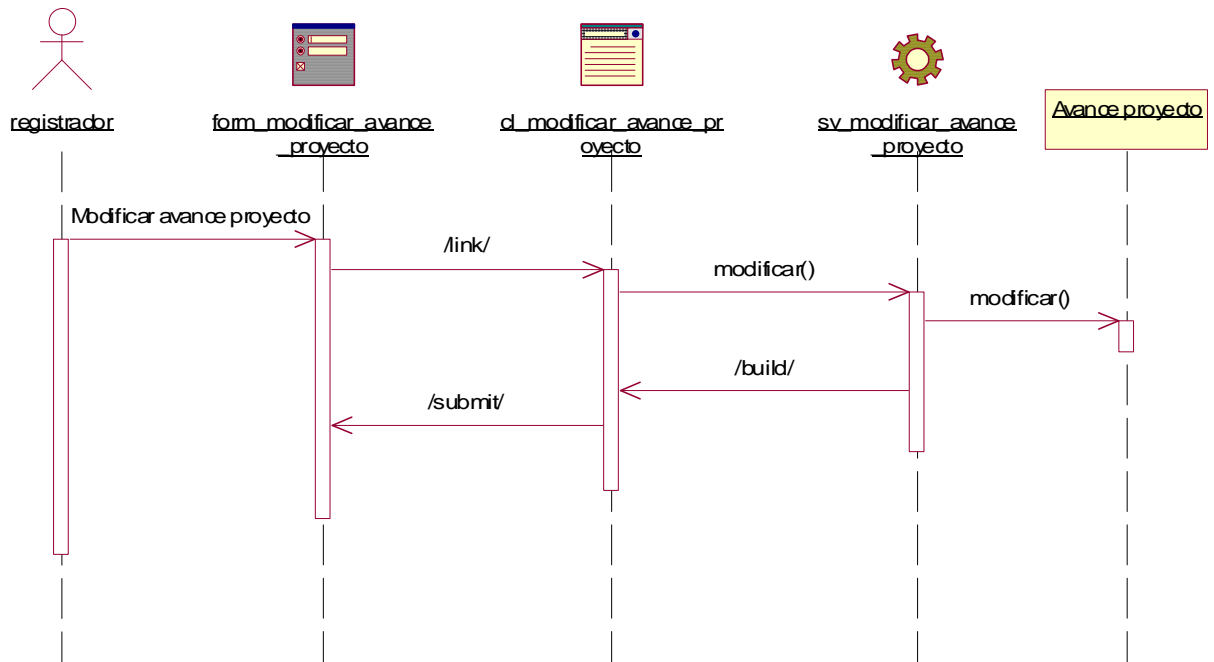
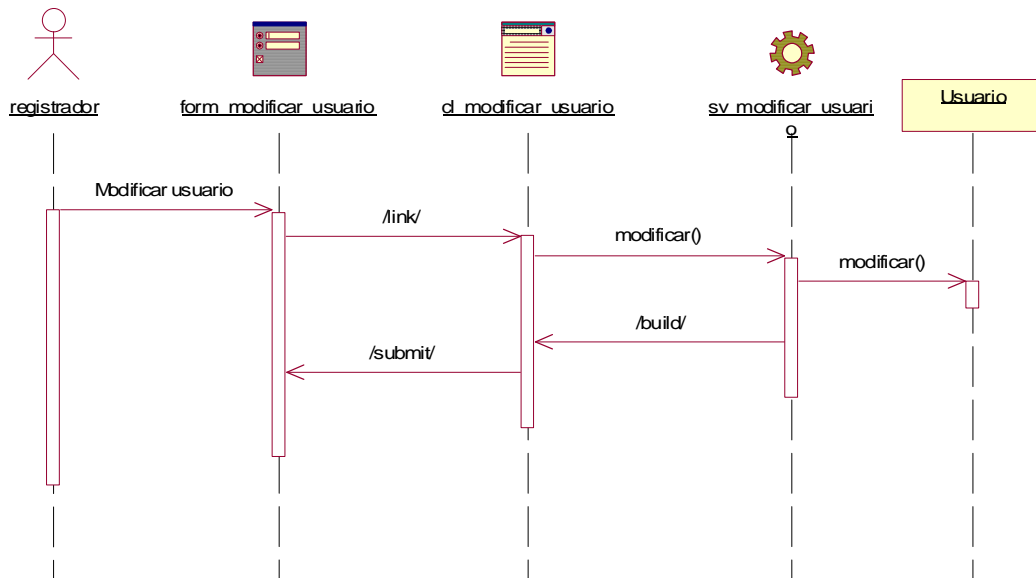
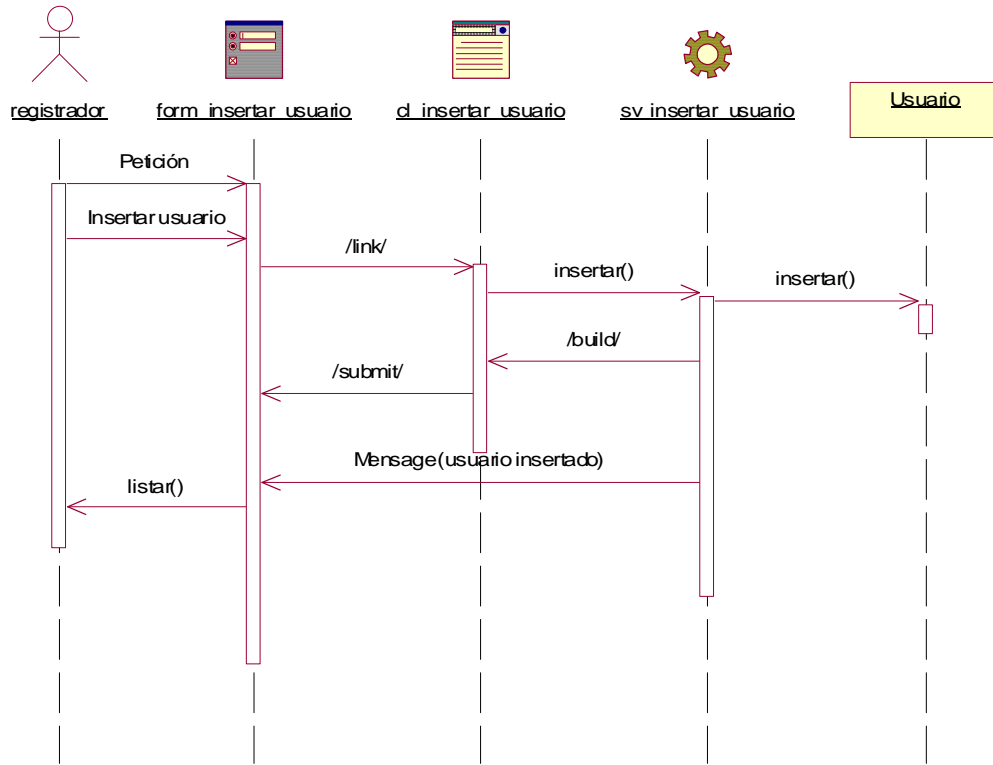


Figura 3.26. Diagrama de secuencia «gestionar avance de proyecto»



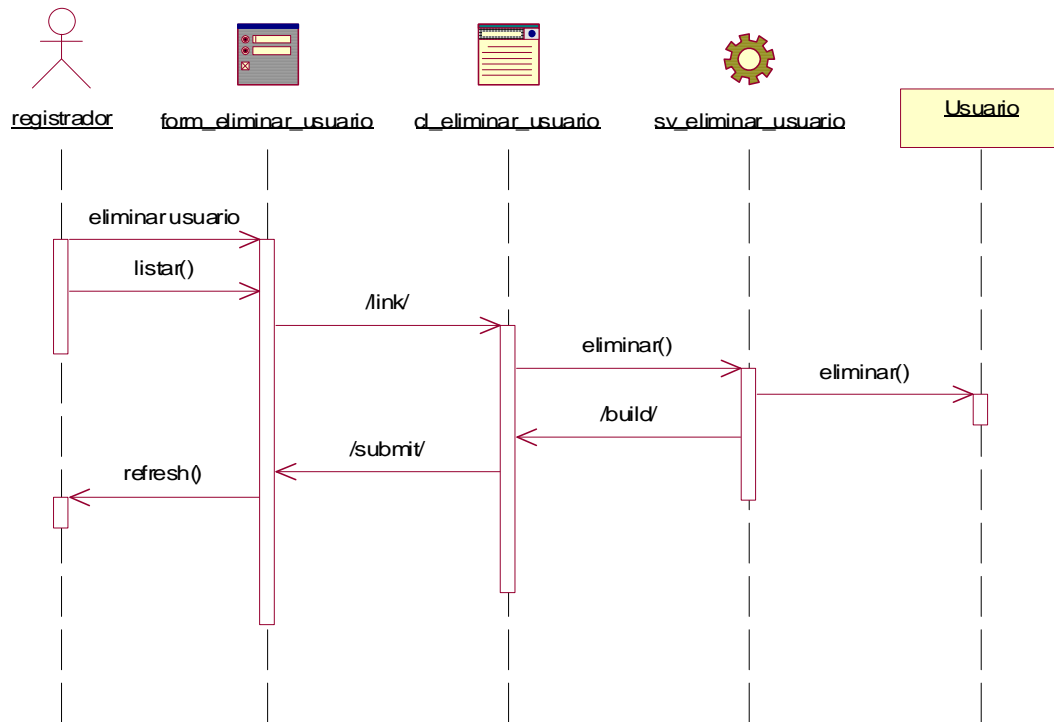
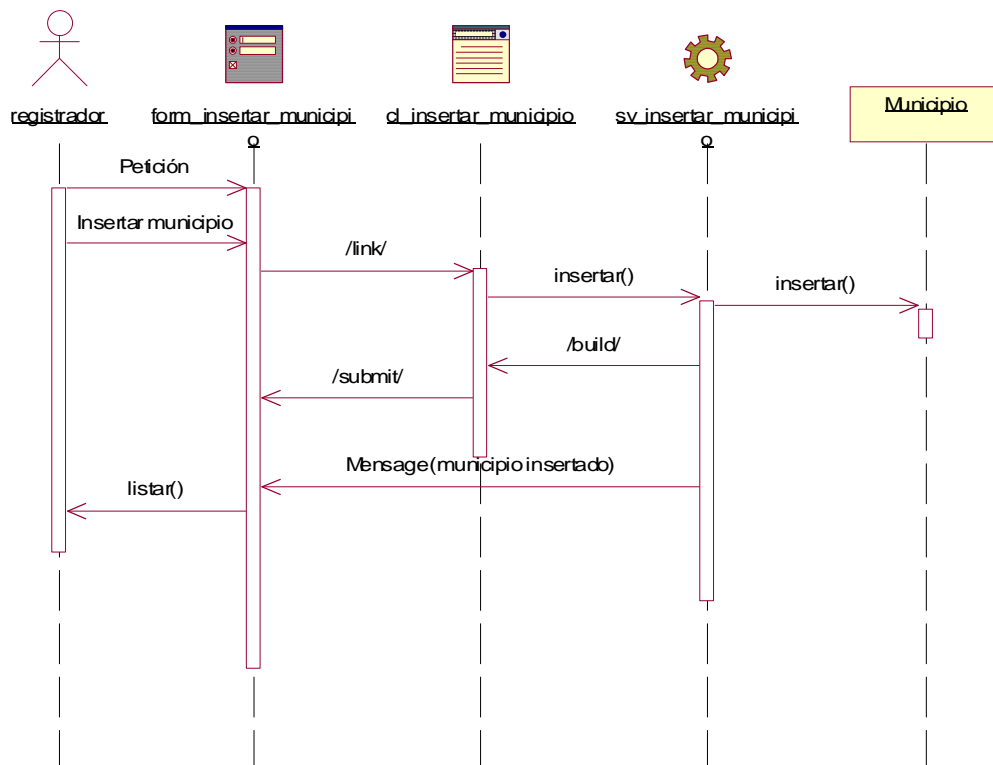


Figura 3.27. Diagrama de secuencia «gestionar usuario»



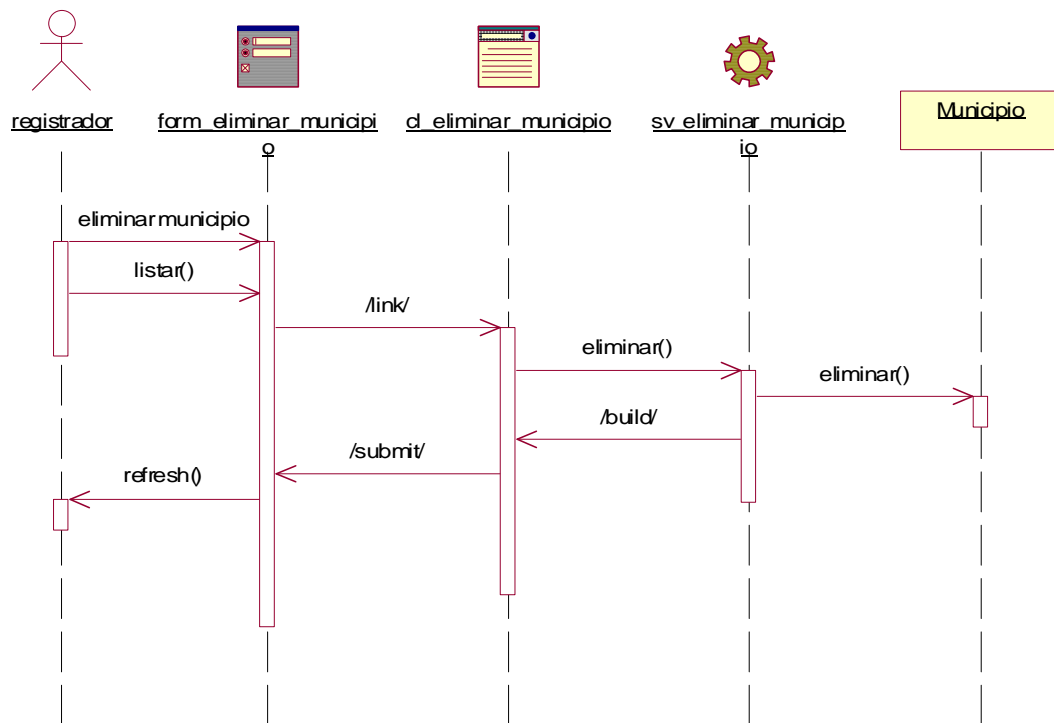
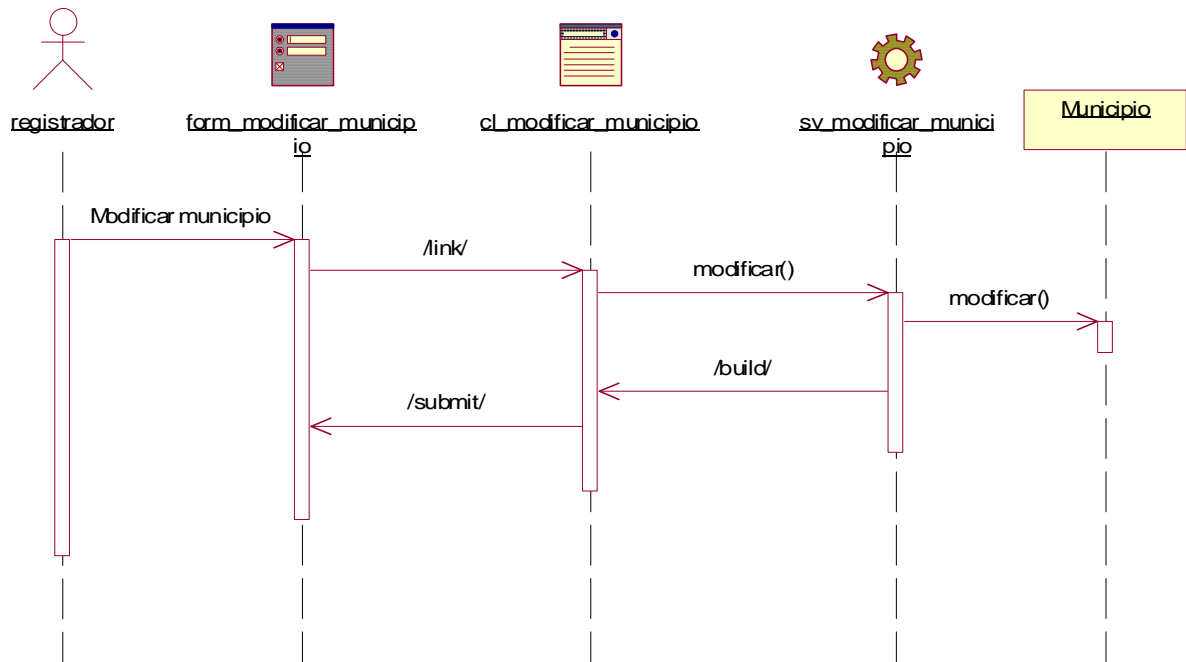


Figura 3.28. Diagrama de secuencia «gestionar municipio»

Anexo 5: Diagramas de componentes

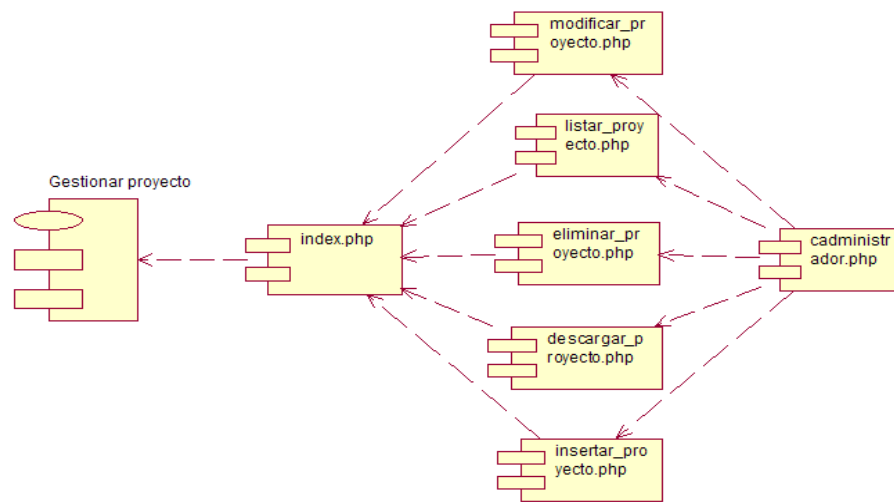


Figura 3.29. Diagrama de componentes «gestionar proyecto»

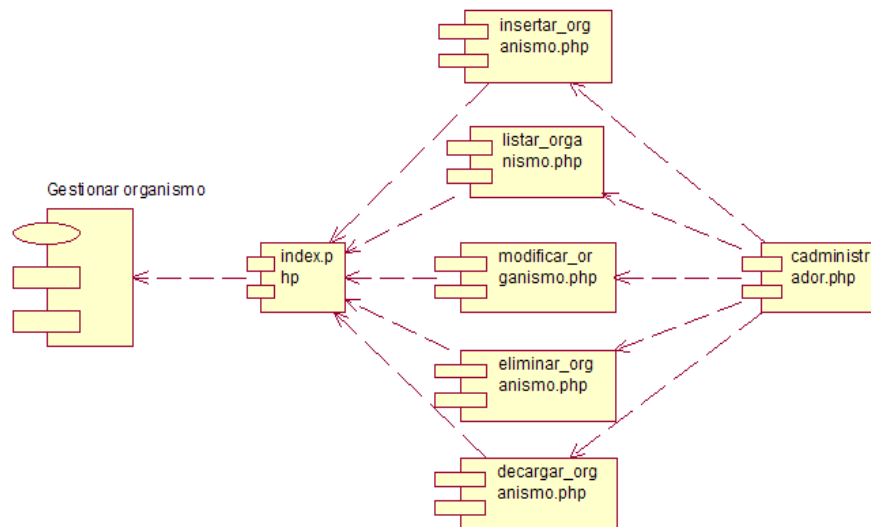


Figura 3.30. Diagrama de componentes «gestionar organismo»

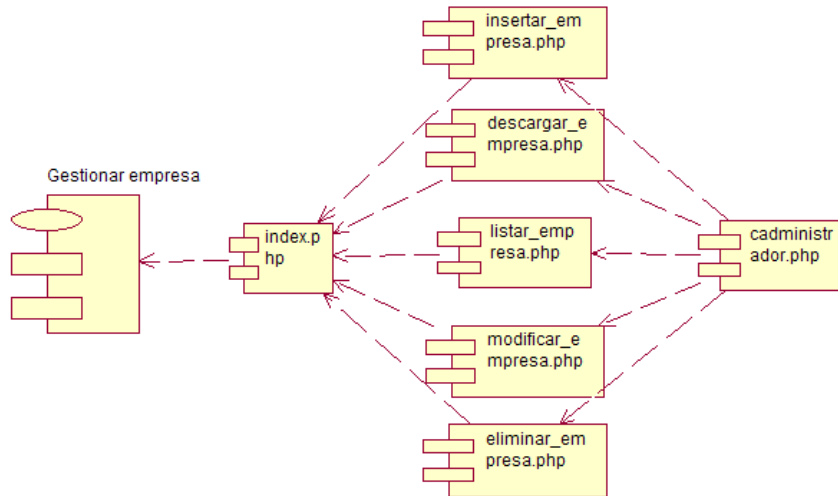


Figura 3.31. Diagrama de componentes «gestionar empresa»

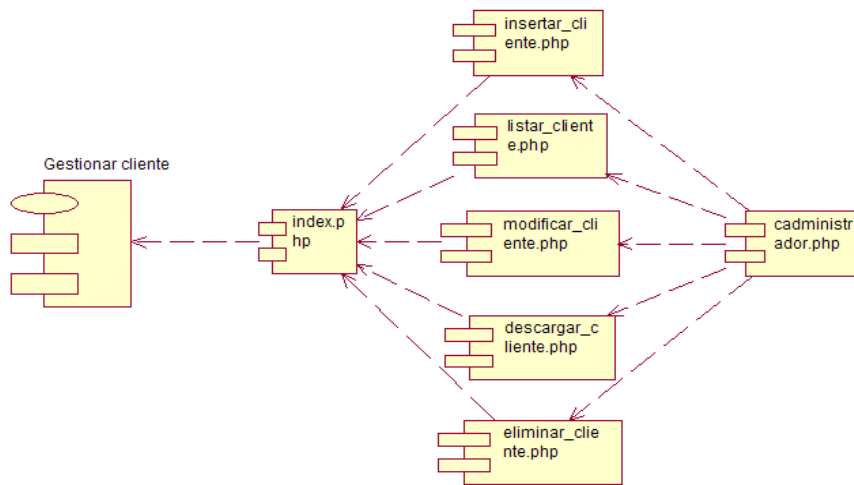


Figura 3.32. Diagrama de componentes «gestionar cliente»

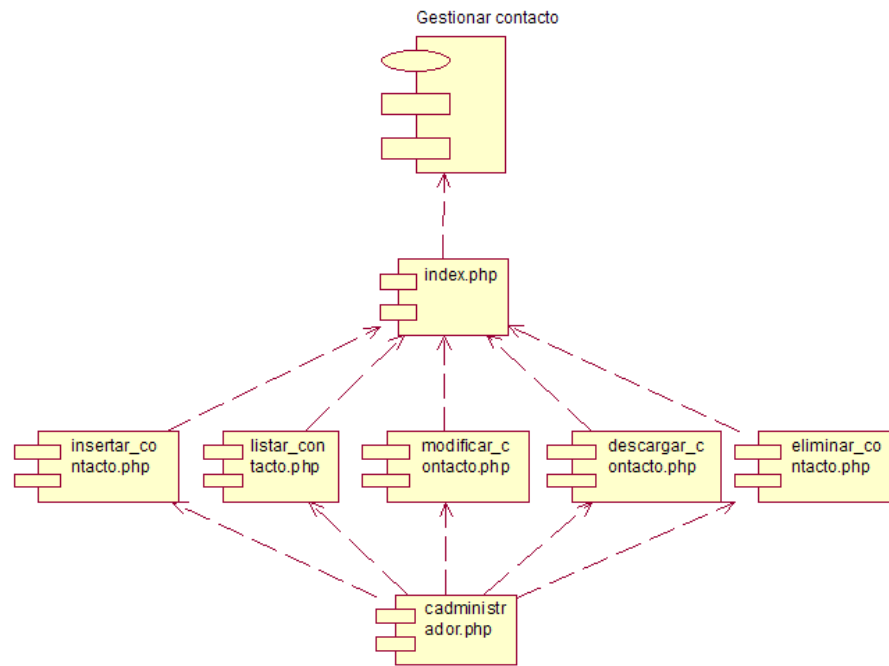


Figura 3.33. Diagrama de componentes «gestionar contacto»

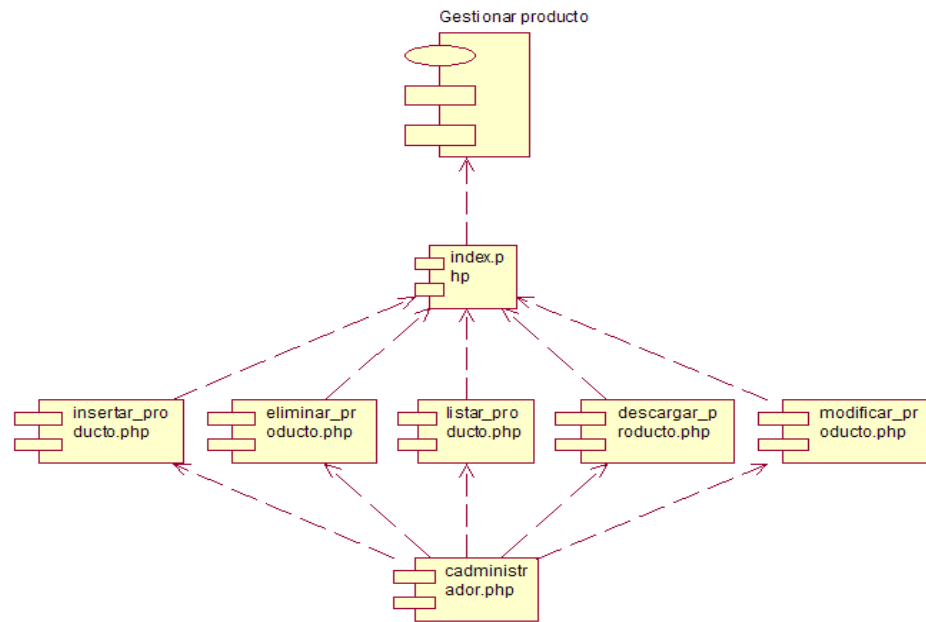


Figura 3.34. Diagrama de componentes «gestionar producto»

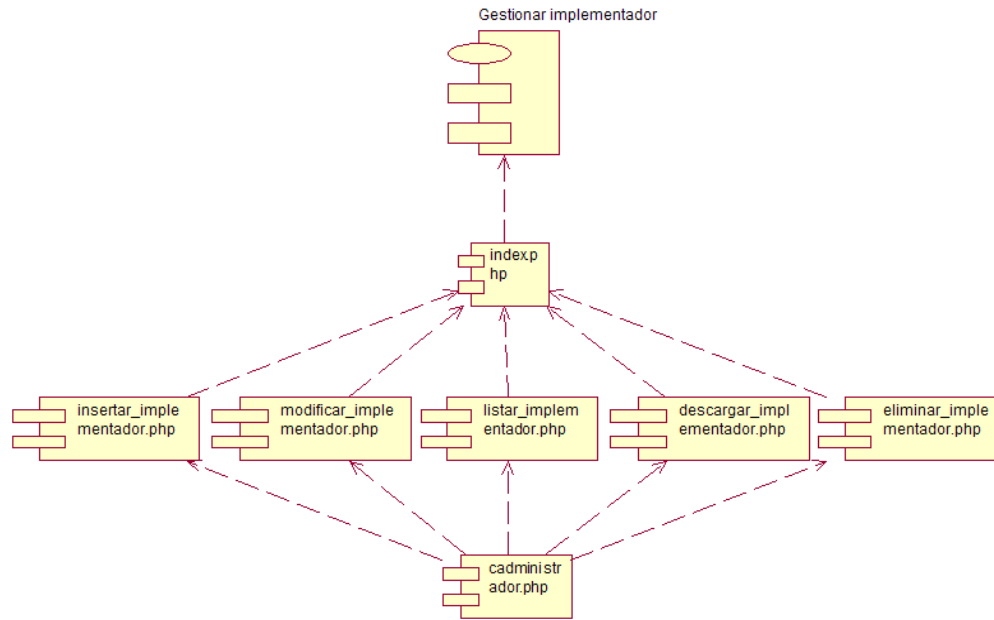


Figura 3.35. Diagrama de componentes «gestionar implementador»

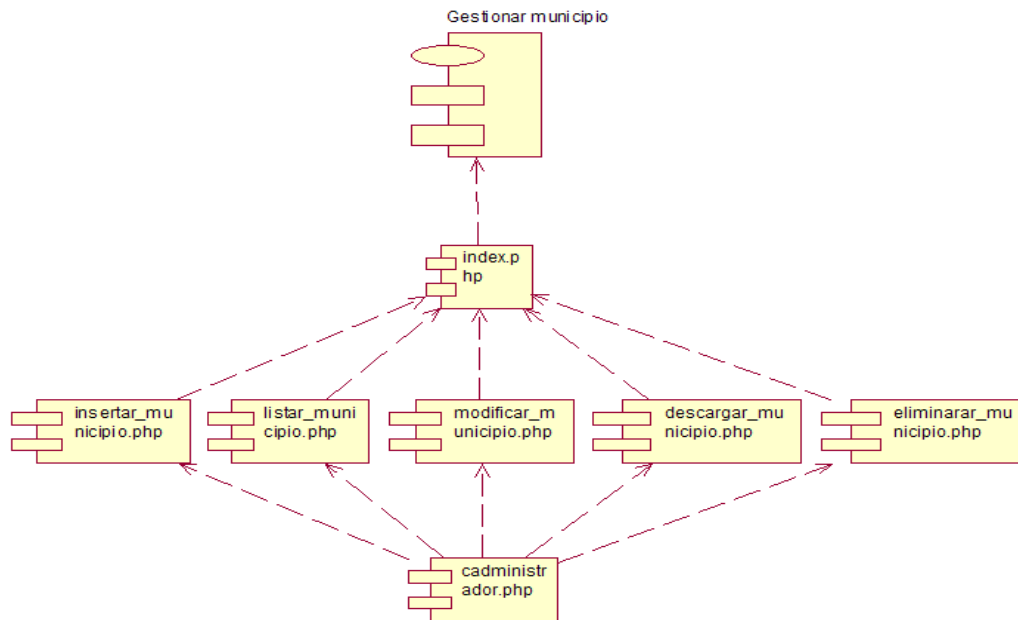


Figura 3.36. Diagrama de componentes «gestionar municipio»

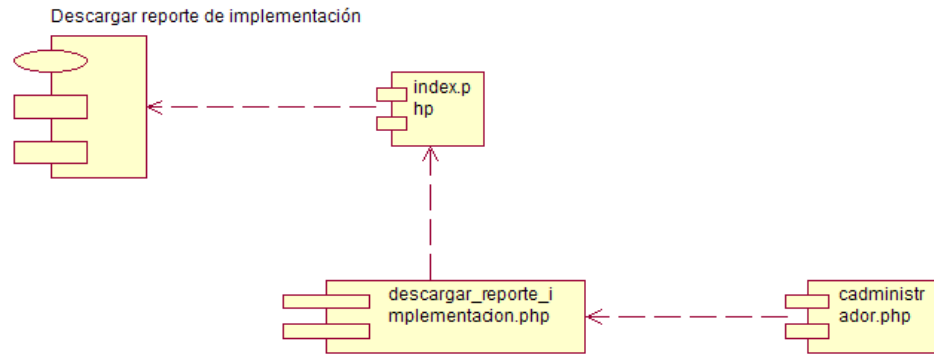


Figura 3.37. Diagrama de componentes «descargar reporte de implementación»

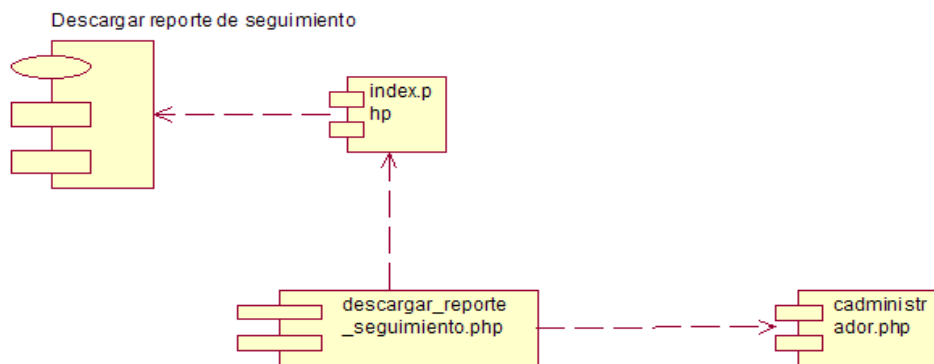


Figura 3.38. Diagrama de componentes «descargar reporte de seguimiento»