

**UNIVERSIDAD PRIVADA ANTENOR ORREGO
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN
Y SISTEMAS**



**TESIS PARA OBTENER TÍTULO DE INGENIERO DE COMPUTACIÓN Y
SISTEMAS**

**“IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO WEB
PARA LA GESTIÓN DE COMPRAS DE LA EMPRESA
CERTICOM S.A.C USANDO LA METODOLOGÍA ICONIX Y
FRAMEWORKS SPRING, HIBERNATE Y RICHFACES”**

AUTOR(ES):

Br. Wilson Carbajal, Dennis Martín

ASESOR:

Ing. Mendoza Puerta, Henry Antonio

TRUJILLO – PERÚ

2013

**“IMPLEMENTACIÓN DE UN SISTEMA INFORMÁTICO WEB
PARA LA GESTIÓN DE COMPRAS DE LA EMPRESA
CERTICOM S.A.C USANDO LA METODOLOGÍA ICONIX Y
FRAMEWORKS SPRING, HIBERNATE Y RICHFACES”**

Elaborado por:

Br. Wilson Carbajal, Dennis Martín

Aprobada por:

Ing. Patricia Vigo Pereyra
Presidente CIP: 070724

Ing. José Calderón Sedano
Secretario CIP: 139198

Ing. Freddy Infantes Quiroz
Vocal CIP: 139578

Ing. Henry Mendoza Puerta
Asesor CIP: 139568

PRESENTACION

Señores Miembros del Jurado:

De conformidad y en cumplimiento de los requisitos estipulados en el reglamento de grados y títulos de la Universidad Privada Antenor Orrego, ponemos a vuestra disposición la presente tesis titulado: **“IMPLEMENTACION DE UN SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE COMPRAS DE LA EMPRESA CERTICOM S.A.C USANDO LA METODOLOGÍA ICONIX Y FRAMEWORKS SPRING, HIBERNATE Y RICHFACES”** para obtener el título profesional de ingeniero de Computación y Sistemas mediante la presente Tesis de investigación.

El contenido del presente trabajo ha sido desarrollado tomando como marco de referencia los conocimientos adquiridos durante mi formación profesional, consulta de fuentes bibliográficas e información obtenida en la **Empresa CERTICOM S.A.C.**

El Autor.

DEDICATORIA

A Dios nuestro creador y hacedor de todas las cosas, quiénnos da aliento y fuerzas de seguir adelante.

A mis padres que con su constante apoyo me han permitido llegar lejos.

A mis hermanos, quienes siempre me dan aliento para seguir adelante.

Wilson Carbajal, Dennis Martín

AGRADECIMIENTOS

Agradezco a mi asesor, por brindarme su paciencia, experiencia, compromiso y excelente dirección en el presente trabajo. A nuestros docentes por impartirnos su conocimiento y profesionalismo.

Asimismo, agradezco a todo el personal de la Empresa, quienes me compartieron su conocimiento en el levantamiento de información, asimismo siempre se mostraron atentos ante nuestras inquietudes.

A mi familia por apoyarnos, por estar siempre presentes y brindarnos sus consejos, experiencia y ganas de superación.

Gracias a todos.

El Autor.

RESUMEN

“IMPLEMENTACION DE UN SISTEMA INFORMÁTICO WEB PARA LA GESTIÓN DE COMPRAS DE LA EMPRESA CERTICOM S.A.C USANDO LA METODOLOGÍA ICONIX Y FRAMEWORKS SPRING, HIBERNATE Y RICHFACES”

Por:

Br. Wilson Carbajal, Dennis Martín

La presente tesis que se desarrolla bajo el marco de la realidad problemática de la empresa CERTICOM S.A.C.; cuenta con un área de logística, desarrollándose de forma ineficiente, porque actualmente lo hacen de manera manual, generándose lentitud para aprobar órdenes de compras, lentitud para explotar la información de las órdenes de compra, así como errores en duplicidad de datos por llevar de manera manual la información, teniendo la necesidad de adoptar un sistema de información automatizado que agilice su proceso de compras.

Ante esto se plantea una solución tecnológica como es la implementación de un Sistema Informático Web la cual utiliza los siguientes frameworks Spring, Hibernate y Richfaces; como también el uso de la metodología ágil como Iconix y como herramienta de diseño Enterprise Architect.

Frente a esta problemática se planteó desarrollar un sistema informático web que permitirá aumentar la eficiencia y eficacia en los procesos de compras, reduciendo tiempo en el registro y procesamiento de la información, también tendrán un mejor control de la información que ayudara a la toma de decisiones; y es así, como CERTICOM S.A.C., estará a la vanguardia con la tecnología para satisfacer las necesidades de sus clientes.

ABSTRACT

“IMPLEMENTATION OF A COMPUTING SYSTEM FOR MANAGING WEB SHOPPING OF CERTICOM COMPANY, USING THE METHODOLOGY ICONIX SAC AND FRAMEWORKS SPRING, HIBERNATE AND RICHFACES”

By:

Br. Wilson Carbajal, Dennis Martin

This thesis is developed under the framework of the problematic reality CERTICOM SAC company, has an area of logistics, unfolding inefficiently, because currently do manually, generating slow to approve purchase orders, slow to exploit information on purchase orders and data duplication errors to bring the information manually, having the need to adopt an automated information system to streamline its procurement process.

Given this arises as a technological solution is the implementation of a Web Information System which uses the following frameworks Spring, Hibernate and Richfaces, as well as the use of agile methodology as Iconix and as a design tool Enterprise Architect.

Faced with this problematic arose web develop a computer system that will increase efficiency and effectiveness in the procurement process, saving time in the recording and processing of information, also have better control of information to assist decision making, and so, as CERTICOM SAC, will be at the forefront with technology to meet the needs of its customers.

ÍNDICE DE CONTENIDOS

DEDICATORIA.....	IV
AGRADECIMIENTOS.....	V
RESUMEN.....	VI
ABSTRACT.....	VII
ÍNDICE DE CONTENIDOS.....	VIII
ÍNDICE DE FIGURAS.....	X
INTRODUCCIÓN.....	XIII
CAPITULO I. FUNDAMENTO TEÓRICO.....	15
1.1. Sistema.....	15
1.1.1. Características.....	16
1.1.2. Parámetros.....	17
1.2. Sistema Informático.....	17
1.2.1. Estructura.....	18
1.3. Web.....	18
1.4. Sistema Informático Web.....	19
1.5. UML (Lenguaje de Modelamiento Unificado) 2.0.....	21
1.5.1. Estructura de modelado.....	22
1.5.2. Diagramas de Comportamiento.....	22
1.5.3. Diagramas de Interacción.....	22
1.6. Metodología Iconix.....	22
1.6.1. Metodología Ágil.....	22
1.6.2. Metodología Ágil ICONIX.....	23
1.6.3. Procesos de la Metodología Iconix.....	25
1.7. Java.....	27
1.8. Plataforma J2EE.....	27
1.9. Framework.....	28
1.10. Patrón MVC (Modelo Vista Controlador).....	29
1.11. Framework Spring.....	31
1.12. Richfaces.....	33
1.13. Hibernate.....	33
1.14. Oracle.....	35
1.14.1. Características principales de Oracle.....	35
1.14.2. Definición de base de datos e instancia de oracle.....	35
1.14.3. Estructura lógica de Oracle.....	35
1.14.4. Estructura física de Oracle.....	36
1.14.5. Seguridad para Oracle.....	37
1.14.6. Estructuras usadas para la recuperación en Oracle.....	38

1.15. Servidor Web Apache	38
CAPITULO II. DESARROLLO DE LA TESIS	41
2.1. Plan de desarrollo del proyecto	41
2.1.1. Estudio Preliminar del Proyecto	41
2.1.2. Propósito del proyecto	42
2.1.3. Hitos del proyecto	42
2.1.4. Requerimientos Funcionales y No funcionales del proyecto	42
2.1.5. Restricciones y Supuestos del proyecto	43
2.2. Análisis de Requisitos	44
2.2.1. Prototipo del sistema completo.	44
2.2.2. Diagrama de casos de uso.....	55
2.2.3. Modelo de Dominio.....	58
2.3. Análisis y Diseño Preliminar.....	59
2.3.1. Diagrama de Clases.	59
Fig. 36 Diagrama de Clases.....	59
2.3.2. Diagrama de Robustez.....	60
2.3.3. Especificación de los requerimientos	68
2.4. Diseño Detallado	74
2.4.1. Diseño de Interfaces.....	74
2.4.2. Diagramas de Secuencia	84
2.4.3. Modelo Lógico	100
2.4.4. Modelo Físico	101
2.5. Implementación	102
2.5.1. Diagrama de despliegue.....	102
2.5.2. Diagrama de Componentes	103
CAPÍTULO III: DISCUSIÓN	104
2.3. Contrastación de la Hipótesis.....	104
2.3.1. Identificación de Variables e Indicadores.....	104
2.3.2. Método de Análisis para la Prueba de los Indicadores Cuantitativos	104
2.3.2.1. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de Generacion de orden de compra.	105
2.3.2.2. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de obtención de reportes. 106	
2.3.2.3. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de aprobación de requerimiento.	108
2.4. Discusión de Resultados	110
CASOS DE PRUEBAS	112
CONCLUSIONES.....	113
RECOMENDACIONES	114
REFERENCIAS.....	115

ÍNDICE DE FIGURAS

Fig. 1 Esquema gráfico general de un sistema.....	16
Fig. 2 Proceso de un Sistema Informático Web.....	20
Fig. 3. Elementos de J2EE.....	28
Fig. 4. Patrón MVC.....	31
Fig. 5. Módulos de Spring.....	32
Fig. 6. Arquitectura de Hibernate.....	34
Fig. 7. Organigrama de la Empresa.....	41
Fig. 8. Prototipo Mantenimiento Categoría de Producto.....	44
Fig. 9. Prototipo Mantenimiento Sub-Categoría de Producto.....	44
Fig. 10. Prototipo Mantenimiento de Producto.....	45
Fig. 11. Prototipo Mantenimiento Proyecto Padre.....	45
Fig. 12 Prototipo Mantenimiento Departamento.....	46
Fig. 13 Prototipo Mantenimiento Proyecto.....	46
Fig. 14 Prototipo Mantenimiento Rubro.....	47
Fig. 15 Prototipo Mantenimiento Proveedor.....	47
Fig. 16 Prototipo Registro Usuario.....	48
Fig. 17 Prototipo Asignar Opciones a Usuario.....	48
Fig. 18 Prototipo Registrar Rol.....	49
Fig. 19 Prototipo Asignar Opciones a Rol.....	49
Fig. 20 Prototipo Registrar Requerimiento.....	50
Fig. 21 Prototipo Consultar Requerimientos.....	50
Fig. 22 Prototipo Aprobar Requerimiento.....	51
Fig. 23 Prototipo Consultar Requerimientos Aprobados.....	51
Fig. 24 Prototipo Cotizar Requerimiento.....	52
Fig. 25 Prototipo Consultar Requerimientos Cotizados.....	52
Fig. 26 Prototipo Confirmar Requerimiento.....	53
Fig. 27 Prototipo Consultar Requerimientos Confirmados.....	53
Fig. 28 Prototipo Aprobar Cotización.....	54
Fig. 29 Prototipo Consultar Requerimientos Aprobados.....	54
Fig. 30 Prototipo Orden de Compra.....	54
Fig. 31 Prototipo Reporte de Órdenes de Compra por Proyecto y Fecha.....	55
Fig. 32 Caso de Uso de Proceso del Negocio.....	55
Fig. 33 Caso de Uso de Proceso de Gestión de Compras.....	56
Fig. 34 Caso de Uso de Gestión de Mantenimiento de Información.....	57
Fig. 35 Modelo de Dominio.....	58
Fig. 36 Diagrama de Clases.....	59
Fig. 37 Diagrama de Robustez Mantenimiento Categoría de Producto.....	60
Fig. 38 Diagrama de Robustez Mantenimiento Sub-Categoría de Producto.....	60
Fig. 39 Diagrama de Robustez Mantenimiento de Producto.....	61
Fig. 40 Diagrama de Robustez Mantenimiento de Rubro.....	61
Fig. 41 Diagrama de Robustez de Mantenimiento de Proveedor.....	62
Fig. 42 Diagrama de Robustez de Mantenimiento de Proyecto Padre.....	62
Fig. 43 Diagrama de Robustez de Mantenimiento de Departamento.....	63
Fig. 44 Diagrama de Robustez de Mantenimiento de Proyecto.....	63
Fig. 45 Diagrama de Robustez de Mantenimiento de Usuario.....	64
Fig. 46 Diagrama de Robustez de Asignar Opción a Usuario.....	64
Fig. 47 Diagrama de Robustez de Mantenimiento Rol.....	65
Fig. 48 Diagrama de Robustez de Asignar Opción a Rol.....	65
Fig. 49 Diagrama de Robustez de Registrar Requerimiento.....	66
Fig. 50 Diagrama de Robustez de Aprobar Requerimiento.....	66
Fig. 51 Diagrama de Robustez de Cotizar Requerimiento.....	67
Fig. 52 Diagrama de Robustez de Confirmar Requerimiento Cotizado.....	67

Fig. 53 Diagrama de Robustez de Aprobar Requerimiento Confirmado y Generar Orden de Compra.....	68
Fig. 54 Diagrama de Robustez de Reportes de Orden de Compra por Proyecto y Fecha	68
Fig. 55 Página Registro de Usuarios	74
Fig. 56 Página Asignar Opción a Usuario.....	75
Fig. 57 Página Registro de Roles.....	75
Fig. 58 Página Asignar Opción a Rol.....	76
Fig. 59 Página de Mantenimiento de Categoría de Productos	76
Fig. 60 Página Mantenimiento de Sub-Categoría de Productos	76
Fig. 61 Página Mantenimiento de Productos.....	77
Fig. 62 Página Mantenimiento de Proyecto Padre	77
Fig. 63 Página Mantenimiento de Departamentos	78
Fig. 64 Página Mantenimiento de Proyectos.....	78
Fig. 65 Página Mantenimiento de Rubros.....	79
Fig. 66 Página Mantenimiento de Proveedores.....	79
Fig. 67 Página Registrar Requerimiento	80
Fig. 68 Página Consultar y Aprobar Requerimiento.....	80
Fig. 69 Página Aprobar Requerimiento	80
Fig. 70 Página Consultar Requerimientos Aprobados	80
Fig. 71 Página Cotizar Requerimiento.....	81
Fig. 72 Página Consultar Requerimientos Cotizados	81
Fig. 73 Página Confirmar Requerimiento.....	82
Fig. 74 Página Consultar Requerimientos Confirmados.....	82
Fig. 75 Página Aprobar Cotización y Generar Orden de Compra.....	83
Fig. 76 Consultar Órdenes de Compra	83
Fig. 77 Página Orden de Compra	83
Fig. 78 Página Reporte de Órdenes de Compra por Proyecto y Fecha.....	84
Fig. 79 Diagrama de Secuencia de Mantenimiento Categoría	84
Fig. 80 Diagrama de Secuencia de Mantenimiento Sub-Categoría.....	85
Fig. 81 Diagrama de Secuencia de Mantenimiento Producto.....	86
Fig. 82 Diagrama de Secuencia de Mantenimiento Rubro.....	87
Fig. 83 Diagrama de Secuencia de Mantenimiento Proveedor	88
Fig. 84 Diagrama de Secuencia de Mantenimiento Proyecto Padre	89
Fig. 85 Diagrama de Secuencia de Mantenimiento Departamento	90
Fig. 86 Diagrama de Secuencia de Mantenimiento Proyecto.....	91
Fig. 87 Diagrama de Secuencia de Mantenimiento Usuario.....	92
Fig. 88 Diagrama de Secuencia de Asignar Opción a Usuario	93
Fig. 89 Diagrama de Secuencia de Mantenimiento Rol	94
Fig. 90 Diagrama de Secuencia de Asignar Opción a Rol.....	95
Fig. 91 Diagrama de Secuencia de Registrar Requerimiento.....	95
Fig. 92 Diagrama de Secuencia de Aprobar Requerimiento.....	96
Fig. 93 Diagrama de Secuencia de Cotizar Requerimiento	97
Fig. 94 Diagrama de Secuencia de Confirmar Requerimiento Cotizado	97
Fig. 95 Diagrama de Secuencia de Aprobar Requerimiento Confirmado y Generar Orden de Compra.....	98
Fig. 96 Diagrama de Secuencia de Reportes de Orden de Compra por Proyecto y Fecha	99
Fig. 97 Diagrama Lógico de Base de Datos.....	100
Fig. 98 Diagrama Físico de Base de Datos	101
Fig. 99 Diagrama de Despliegue.....	102
Fig. 100 Diagrama de Componentes.....	103
Fig. 101 Región de Aceptación y Rechazo para la prueba de Hipótesis	106
Fig. 102 Región de Aceptación y Rechazo para la prueba de Hipótesis	108
Fig. 103 Región de Aceptación y Rechazo para la prueba de Hipótesis	109

ÍNDICE DE TABLAS

Tabla 1 Cuadro del Proceso de Iconix	26
Tabla 2 Hitos del Proyecto	42
Tabla 3 Especificación de Mantenimiento de Categoría de Producto.....	69
Tabla 4 Especificación de Mantenimiento de Sub-Categoría de Producto	69
Tabla 5 Especificación de Mantenimiento de Producto.....	69
Tabla 6 Especificación de Mantenimiento del Rubro	70
Tabla 7 Especificación de Mantenimiento del Proveedor.....	70
Tabla 8 Especificación de Mantenimiento del Proyecto Padre.....	70
Tabla 9 Especificación de Mantenimiento del Departamento.....	71
Tabla 10 Especificación de Mantenimiento del Proyecto	71
Tabla 11 Especificación de Mantenimiento del Usuario	71
Tabla 12 Especificación de Asignación de Opción a Usuario.....	72
Tabla 13 Especificación de Mantenimiento del Rol.....	72
Tabla 14 Especificación de Asignación de Opción a Rol	72
Tabla 15 Especificación de Registro de Requerimiento y Detalle de Requerimiento	72
Tabla 16 Especificación de Aprobación de Requerimientos.....	73
Tabla 17 Especificación de Cotización de Requerimientos	73
Tabla 18 Especificación de Confirmación Requerimiento Cotizado	73
Tabla 19 Especificación de Aprobación Requerimiento Confirmado.....	74
Tabla 20 Especificación de Requerimiento Reporte de Órdenes de Compra por proyecto y fecha	74
Tabla 21: Indicadores.....	104
Tabla 22: Resultado para el indicador Minimizar el Tiempo de Generar Órdenes de Compra .	110
Tabla 23: Resultado para el indicador Minimizar el Tiempo de Obtención de Reportes.....	110
Tabla 24: Resultado para el indicador Minimizar el Tiempo de Aprobar Requerimientos.....	110

INTRODUCCIÓN

En la modernidad, el uso de tecnología en los sistemas informáticos juegan un papel muy importante en el desarrollo de las organizaciones y en el control de las operaciones administrativas y financieras entre otras, la obtención de una información inmediata y eficaz dentro de la estructura organizativa determina el éxito y el alcance de los objetivos con mayor eficacia dentro de la empresa, haciéndose indispensables para ser competitivas en este mercado globalizado.

Estos sistemas informáticos se han desarrollado en una gran cantidad de aplicaciones; como aplicaciones de escritorio, web y móvil, pero solo los sistemas web han proporcionado soluciones fáciles y eficientes en su desarrollo y por lo cual las empresas hacen más uso de estos, por su disponibilidad y usabilidad.

En vista de esto los sistemas web se han visto obligados a evolucionar su forma de desarrollo e implementación, por tal motivo existen muchas metodologías ágiles y tecnologías libres; como tenemos en modelamiento a RUP, XP, SCRUM y ICONIX, siendo esta una de las que más cumple con estándares necesarios para el desarrollo de aplicaciones web.

En el uso de tecnologías web se han desarrollado diferentes formas de construcción, pero los que más destacan son los frameworks de java que facilitan el uso e implementación de aplicaciones web proporcionando una estructura definida la cual ayuda a crear aplicaciones con mayor rapidez.

Por ello el área temática de la presente tesis está relacionada con sistemas de tecnología de información de tipo web usando metodologías ágiles y frameworks de java de implementación.

La empresa CERTICOM S.A.C., presenta la siguiente realidad problemática: las actividades realizadas en la empresa son de forma manual en el proceso de gestión de compras, la cual genera lentitud para aprobar órdenes de compras, lentitud para explotar la información de las órdenes de compra, así como errores en duplicidad de datos. Es así que definimos el siguiente **problema: ¿De qué manera se puede mejorar la Gestión de Compras de la empresa Certicom S.A.C utilizando Tecnologías de Información?**

La hipótesis planteada es la siguiente: Implementando un sistema informático web y utilizando la metodología Iconix y Frameworks Spring, Hibernate y Richfaces podrá automatizar la Gestión de Compras de la Empresa Certicom S.A.C

Teniendo como objetivo general: **”Implementar un sistema informático web para la Gestión de compras de la Empresa Certicom S.A.C, utilizando la metodología Iconix y Frameworks Spring ,Hibernate y Richfaces.”**

Teniendo como objetivos específicos:

- Realizar la investigación bibliográfica sobre la metodología Iconix orientada al desarrollo de aplicaciones web.
- Recopilar información propia de la empresa por medio de entrevistas o reuniones, que permita la posterior identificación de procesos y requerimientos de usuarios.
- Realizar el análisis y diseño a partir de la información recopilada aplicando las fases de la metodología Iconix.
- Implementar el software del sistema informático web utilizando los frameworks: Spring, Hibernate y Richfaces.
- Realizar las pruebas al sistema informático web por cada requerimiento implementado

Finalmente el presente trabajo comprende los siguientes capítulos:

En el Capítulo I Fundamento teórico como parte conceptual de algunas terminologías y temas que involucren este trabajo. El Capítulo II Desarrollo de la Tesis, se detalla la metodología ICONIX, descrita en todas sus etapas, así como las conclusiones y recomendaciones a las que se llegó tras la realización de la tesis y como resultado final tenemos las conclusiones y recomendaciones, expuestas a los interesados en el tema.

CAPITULO I. FUNDAMENTO TEÓRICO

1.1. Sistema

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software).

Cada sistema existe dentro de otro más grande, por lo tanto un sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un súper sistema.

Los sistemas tienen límites o fronteras, que los diferencian del ambiente. Ese límite puede ser físico (el gabinete de una computadora) o conceptual. Si hay algún intercambio entre el sistema y el ambiente a través de ese límite, el sistema es abierto, de lo contrario, el sistema es cerrado.

El ambiente es el medio externo que envuelve física o conceptualmente a un sistema. El sistema tiene interacción con el ambiente, del cual recibe entradas y al cual se le devuelven salidas. El ambiente también puede ser una amenaza para el sistema.

Un grupo de elementos no constituye un sistema si no hay una relación e interacción, que de la idea de un "todo" con un propósito, esto puede ser apreciado en la Figura N° 1

En general, podemos señalar que, ante la palabra "sistemas", todos los que la han definido están de acuerdo en que es un conjunto de partes coordinadas en interacción para alcanzar un conjunto de objetivos.

El ser humano, por ejemplo, es un sistema (podríamos añadir un sistema maravillosamente construido y diseñado) con muchas partes diferentes que contribuyen de distinta forma a mantener su vida, su reproducción y su acción.

Otra definición, que agrega algunas características adicionales, señala que un sistema es un grupo de partes y objetos que interactúan y que forman un todo o que se encuentran bajo la influencia de fuerzas en alguna relación definida.



Fig. 1 Esquema gráfico general de un sistema
(Richard Dios, 2009).

1.1.1. Características

Todo sistema tiene algunos propósitos u objetivos. Los elementos (u objetos), como también las relaciones, definen una distribución que trata siempre de alcanzar un objetivo.

Globalismo o totalidad es un cambio en una de las unidades del sistema, con probabilidad producirá cambios en las otras. El efecto total se presenta como un ajuste a todo el sistema. Hay una relación de causa/efecto. De estos cambios y ajustes, se derivan dos fenómenos: entropía y homeostasia.

Entropía es la tendencia de los sistemas a desgastarse, a desintegrarse, para el relajamiento de los estándares y un aumento de la aleatoriedad. La entropía aumenta con el correr del tiempo. Si aumenta la información, disminuye la entropía, pues la información es la base de la configuración y del orden.

Homeostasia es el equilibrio dinámico entre las partes del sistema. Los sistemas tienen una tendencia a adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno.

1.1.2. Parámetros

Los parámetros de los sistemas son:

- *Entrada o insumo o impulso (input)*: es la fuerza de arranque del sistema, que provee el material o la energía para la operación del sistema.
- *Salida o producto o resultado (output)*: es la finalidad para la cual se reunieron elementos y relaciones del sistema. Los resultados de un proceso son las salidas, las cuales deben ser coherentes con el objetivo del sistema. Los resultados de los sistemas son finales, mientras que los resultados de los subsistemas son intermedios.
- *Procesamiento o procesador o transformador (throughput)*: es el fenómeno que produce cambios, es el mecanismo de conversión de las entradas en salidas o resultados. Generalmente es representado como la caja negra, en la que entra los insumos y salen cosas diferentes, que son los productos.
- *Retroacción o retroalimentación o retroinformación (feedback)*: es la función de retorno del sistema que tiende a comparar la salida con un criterio preestablecido, manteniéndola controlada dentro de aquel estándar o criterio.
- *Ambiente*: es el medio que envuelve externamente el sistema. Está en constante interacción con el sistema, ya que éste recibe entradas, las procesa y efectúa salidas. La supervivencia de un sistema depende de su capacidad de adaptarse, cambiar y responder a las exigencias y demandas del ambiente externo. Aunque el ambiente puede ser un recurso para el sistema, también puede ser una amenaza (Monografías, 2006).

1.2. Sistema Informático

Un sistema informático como todo sistema, es el conjunto de partes interrelacionadas, hardware, software y de recurso humano (humanware).

Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos.

La computadora personal o PC, junto con la persona que lo maneja y los periféricos que los envuelven, resultan de por sí un ejemplo de un sistema informático.

El hardware incluye una o varias CPU, memoria, sistemas de almacenamiento externo, etc. El software incluye al sistema operativo, firmware y aplicaciones, siendo especialmente importante los sistemas de gestión de bases de datos.

Por último el soporte humano incluye al personal técnico (analistas, programadores, operarios, etc.) que crean y/o mantienen el sistema y a los usuarios que lo utilizan (Alfonso, 2010).

1.2.1. Estructura

Los sistemas informáticos suelen estructurarse en subsistemas.

Subsistema físico: asociado al hardware. Incluye entre otros elementos la CPU, memoria principal, la placa base, etc.

Subsistema lógico: asociado al software y la arquitectura. Incluye al sistema operativo, el firmware, las aplicaciones y las bases de datos.

Recursos humanos: hace referencia al personal que está relacionado con el sistema. Especialmente usuarios y técnicos (analistas, diseñadores, programadores, operarios, mantenedores, etc) (Wikipedia, 2013).

1.3. Web

La palabra web (del inglés: red, malla, telaraña¹). La Web es una idea que se construyó sobre la internet. Las conexiones físicas son sobre la internet, pero introduce una serie de ideas nuevas, heredando las ya existentes.

Empezó a principios de 1990, en Suiza en el centro de investigación CERN (centro de Estudios para la Investigación Nuclear) y la idea fue de Tim Berners-Lee, que se gestó observando una libreta que él usaba para añadir y mantener referencias de cómo funcionaban los ordenadores en el CERN.

Antes de la Web, la manera de obtener los datos por la Internet era caótica; había un sinnúmero de maneras posibles y con ello había que conocer múltiples programas y sistemas operativos. La Web introduce un concepto fundamental; la posibilidad de lectura universal, que consiste en que una vez que la información esté disponible, se pueda acceder a ella desde cualquier ordenador, desde cualquier país, por cualquier persona autorizada, usando un único y simple

programa. Para que esto fuese posible, se utilizan una serie de conceptos, el más conocido es el hipertexto.

World Wide Web (también conocida como "la Web"), el sistema de documentos (o páginas web) interconectados por enlaces de hipertexto, disponibles en Internet.

WorldWideWeb, el primer navegador web, más tarde renombrado a Nexus.

Una página web: documento o fuente de información, generalmente en formato HTML y que puede contener hiperenlaces a otras páginas web. Dicha página web, podrá ser accesible desde un dispositivo físico, una intranet, o Internet.

Un sitio web, que es un conjunto de páginas web, típicamente comunes a un dominio o subdominio en la World Wide Web.

Un servidor web, un programa que implementa el protocolo HTTP para transferir lo que llamamos hipertextos, páginas web o páginas HTML. También se le da este nombre, al ordenador que ejecuta este programa (Instituto tecnológico Tijuana, 2009).

Web 2.0, término acuñado por Tim en 2004 para referirse a una segunda generación de Web basada en comunidades de usuarios y una gama especial de servicios web, como las redes sociales, los blogs, los wikis, que fomentan la colaboración y el intercambio ágil de información entre los usuarios.

Web 3.0: El término Web 3.0 apareció por primera vez en 2006 en un artículo de Jeffrey Zeldman, crítico de la Web 2.0 y asociado a tecnologías como AJAX. Actualmente existe un debate considerable en torno a lo que significa Web 3.0, y cuál es la definición acertada (Azhlee, 2009).

1.4. Sistema Informático Web

Los sistemas informáticos web son relativamente nuevos en el mundo de la computación, son un nuevo reto para los ingenieros del software. Las aplicaciones web son cada vez más complejas, como el software, al principio no se modelaba, pronto surgen metodologías que intentan solucionar el problema.

Los sistemas Web fomentan un entorno de requisitos muy cambiantes, gran número de usuarios y/o requisitos (mundial); el equipo de desarrolladores suele ser pequeño.

Los modelos son abstracciones que simplifican nuestra comprensión de los sistemas, como lenguaje de modelado ya existente deberíamos considerar a UML.

JimConallen recomienda modelar webs extendiendo UML y aplicando un patrón de diseño llamado MVC (modelo-vista-controlador), El servidor web ofrece páginas web y recursos (css, js, imagenes, flash, etc.) identificándose de forma única mediante URL o URI.

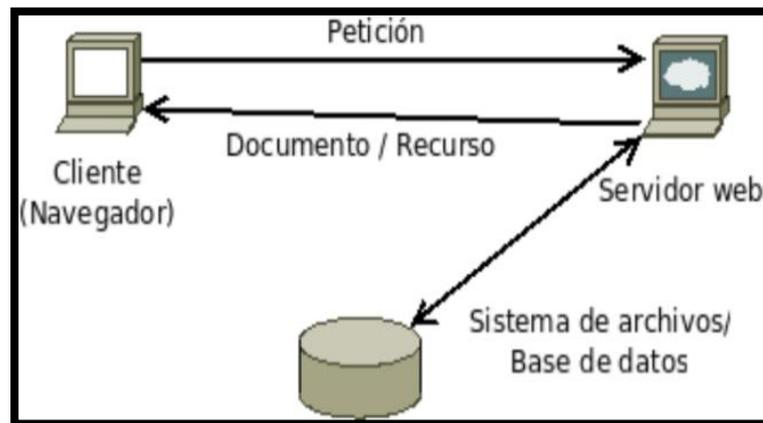


Fig. 2 Proceso de un Sistema Informático Web.

Los sistemas informáticos web conforman una comunicación entre cliente y servidor utilizando el protocolo HTTP. No mantiene conexión tras una petición. Eso genera, que sea necesario recurrir a cookies para conocer el estado del cliente. (Sesiones).

Una aplicación web genera una página web para un cliente en función de N variables.

Una aplicación web es un sistema informático Web que nos ofrece la lógica de negocio. (Interfaces, formularios).

- **Lenguajes en la parte del cliente**
 - Lenguajes de script como javascript (estándar ECMA), y Visual Basic Script(Microsoft). Pueden usarse para complementar la lógica de negocio.
 - La web es síncrona pero la tendencia es la Web asíncrona gracias a un conjunto de tecnologías denominadas como AJAX.
 - Para el renderizado Web se usa HTML, XHTML o XML. Complementados con CSS (hojas de estilo en cascada).
 - Flash como lenguaje de presentación, aporta multimedia a la web. Applet, java.

- **Lenguajes en la parte del servidor**
 - Los más conocidos son PHP (software libre), JSP (Sun Microsystems) y ASP/ASP.NET(Microsoft).
 - Las primeras versiones de PHP y ASP no separaban bien las capas. Pudiendo llegar a tener mezcladas las tres capas: presentación (XHTML), lógica de negocio (PHP) y modelo de datos(SQL). Procedimentales.
 - La separación de capas es difícil ya que tradicionalmente la lógica de negocio se encarga de generar la presentación dinámicamente.
 - En aplicaciones grandes, es preferible por usar lenguajes que implementan MVC (Ricardo Marmolejo, 2008).

1.5. UML (Lenguaje de Modelamiento Unificado) 2.0

Lenguaje Unificado de Modelado (LUM) o (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

1.5.1. Estructura de modelado

- ✓ Diagrama de clases
- ✓ Diagrama de componentes
- ✓ Diagrama de objetos
- ✓ Diagrama de estructura compuesta (UML 2.0)
- ✓ Diagrama de despliegue
- ✓ Diagrama de paquetes

1.5.2. Diagramas de Comportamiento

- ✓ Diagrama de actividades
- ✓ Diagrama de casos de uso
- ✓ Diagrama de estados

1.5.3. Diagramas de Interacción

- ✓ Diagrama de secuencia
- ✓ Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x)
- ✓ Diagrama de tiempos (UML 2.0)
- ✓ Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0) (Wikipedia, 2013)

1.6. Metodología Iconix

1.6.1. Metodología Ágil

Es un lenguaje de modelamiento de notación gráfica (incluye diferentes tipos de diagramas), que define el proceso, quién debe hacer, qué, cuándo y cómo alcanzar un objetivo.

El objetivo principal de un método ágil es minimizar la documentación de desarrollo empleándola fundamentalmente como vehículo de comprensión de problemas dentro del grupo de trabajo y de comunicación con los usuarios.

Los desarrolladores: necesitamos obtener aplicaciones en menor tiempo, más vistosas y de menor costo.

Los usuarios: exigen calidad, sistemas fáciles de mantener, extender y modificar.

1.6.2. Metodología Ágil ICONIX

Es un proceso simplificado en comparación con otros procesos más tradicionales, que unifica un conjunto de métodos de orientación a objetos con el objetivo de abarcar todo el ciclo de vida de un proyecto.

Presenta claramente las actividades de cada etapa y exhibe una secuencia de pasos que deben ser seguidos.

Los desarrollos de aplicaciones van cambiando por innovaciones tecnológicas, estrategias de mercado y otros avatares de la industria de la informática, esta lleva a los desarrolladores de aplicaciones a evolucionar para obtener aplicaciones en medio tiempo, más vistosas y de menos costo.

Los usuarios exigen calidad frente a los requisitos y los desarrollos de aplicaciones deben contar con técnicas y herramientas logrando satisfacer las necesidades de los usuarios y obteniendo sistemas fáciles de mantener, extender y modificar.

Claro está, que es indispensable, el uso de una metodología para el desarrollo de sistemas, logrando un sistema sano, que cumpla con los requerimientos de los usuarios.

Está entre la complejidad del RationalUnifiedProcesses y la simplicidad de Extreme Programming.

a. Características de ICONIX

- **Iterativo e incremental:** varias iteraciones ocurren entre el desarrollo del modelo del dominio y la identificación de los casos de uso. El modelo estático es incrementalmente refinado por los modelos dinámicos.
- **Trazabilidad:** cada paso está referenciado por algún requisito. Se define trazabilidad como la capacidad de seguir una relación entre los diferentes “artefactos de software” producidos.
- **Dinámica del UML:** La metodología ofrece un uso “dinámico” del UML por que utiliza algunos diagramas del UML, sin exigir la utilización de todos, como en el caso de RUP.

b. Tareas de ICONIX

- **Análisis de Requisitos**

Se realiza un levantamiento de todos los requisitos del sistema a desarrollar.

- **Modelo de Dominio:**

Con los requisitos se construye el modelo estático del sistema.

- **Prototipación Rápida:**

Se usa para simular el diseño del sistema.

Se espera que los usuarios lo evalúen como si fuera el sistema final.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo.

El proceso se repite y finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando es evidente que no se obtendrá mayor beneficio con una iteración adicional.

- **Modelo de Casos de Uso:**

El modelo de los casos de uso comprende los actores, el sistema y los propios casos de uso.

Los casos de uso permiten a los usuarios estructurar y articular sus deseos; les obligan a definir la manera como querrían interactuar con el sistema, a precisar qué informaciones quieren intercambiar y a describir lo que debe hacerse para obtener el resultado esperado.

- **Análisis y Diseño Preliminar**

- **Descripción de Casos de Uso:**

Los Casos de Uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario; permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.

- **Diagrama de Robustez:**

Ilustra gráficamente las interacciones entre los objetos participantes de un caso de uso. Los que pueden ser:

- Objetos de interfaz. (Pantallas)
 - Objetos entidad. (Almacenamientos)
 - Objetos de control. (Gestores)
-
- **Diseño**
 - **Diagrama de Secuencia:**

Es el núcleo del modelo dinámico y muestra todos los cursos alternos que pueden tomar los casos de uso.

Especifica el comportamiento. La representación se concentra sobre la expresión de las interacciones.

Se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos.
-
- **Implementación**
 - **Escribir /Generar el Código:**

La importancia de la interactividad, accesibilidad y navegación en el software harán que el usuario se sienta seguro y cómodo al poder hacer uso de la aplicación sin inconvenientes.

Pero además debemos tener en cuenta factores como:

 - *La Reusabilidad:* que es la posibilidad de hacer uso de los componentes en diferentes aplicaciones.
 - *La Extensibilidad:* que consiste en modificar con facilidad el software.
 - *La Confiabilidad:* realización de sistemas descartando las posibilidades de error.

Realizar pruebas. Test de casos, datos y resultados. Test de integración con los usuarios para verificar la aceptación de los resultados.

1.6.3. Procesos de la Metodología Iconix

El proceso de Iconix es un proceso abierto, libre de usar. El proceso se centra en los casos de uso y el código.

El proceso de Iconix se divide en los flujos de trabajo dinámicos y estáticos, que son altamente repetitivos es posible que vaya a través de una repetición de todo el proceso para un lote pequeño de casos de uso (tal vez un par de paquetes de valor,

que no es una cantidad enorme, dado que cada caso de uso es sólo un par de párrafos), hasta llegar al código fuente y pruebas unitarias.

Por esta razón, el Proceso de Iconix está bien adaptado a los proyectos ágiles, donde se necesita una rápida retroalimentación de factores tales como los requisitos, el diseño, y las estimaciones.

Tabla 1 Cuadro del Proceso de Iconix

FASES	ACTIVIDADES	RESULTADOS
Análisis de Requisitos	<ul style="list-style-type: none"> • Requisitos funcionales (Casos de Uso): Definir lo que el sistema debe ser capaz de hacer. • Modelado de Dominio: Definir el espacio del problema en términos inequívocos. • Requisitos de comportamiento (Prototipación Rápida): Definir cómo el usuario y el sistema va a interactuar. Empezar con un prototipo de interfaz gráfica de usuario e identificar todos los casos el uso que se van a desarrollar. 	<ul style="list-style-type: none"> • Revisión de los Requisitos
Análisis y Diseño Preliminar	<ul style="list-style-type: none"> • Análisis de robustez: Dibujar un diagrama de robustez (un "cuadro objeto" de los pasos de un caso de uso). • Describir las acciones y limitaciones de los casos de uso detalladamente. • Actualizar el modelo de dominio: aquí se puede descubrir las clases que faltan, las ambigüedades correcta y agregar atributos y operaciones a los objetos de dominio. 	<ul style="list-style-type: none"> • Revisión de Diseño Preliminar.
Diseño Detallado	<ul style="list-style-type: none"> • Diagrama de Secuencia: dibujar un diagrama de secuencia (un diagrama de secuencia por cada caso de uso) para mostrar en detalle cómo se va a poner en práctica el caso de uso. • Actualizar el modelo de dominio: los objetos de dominio son realmente las clases de dominio, o entidades. 	<ul style="list-style-type: none"> • Revisión Crítica del Diseño.
Implementación	<ul style="list-style-type: none"> • Diagrama de Despliegue: dibujar un diagrama de despliegue para mostrar en detalle cómo va distribuirse el funcionamiento del sistema. • Diagrama de Componentes: describir los elementos físicos del sistema y sus 	<ul style="list-style-type: none"> • Ejecución de la Aplicación y entrega del sistema.

	relaciones.	
	• Codificar y hacer pruebas unitarias: Escribir el código y las pruebas unitarias.	
	• Realizar una revisión y actualización del Código (Carla de San Martín, 2009).	

1.7. Java

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.¹²

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en el 1995. A partir de mayo del 2007, en cumplimiento con las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath (Wikipedia, 2013).

1.8. Plataforma J2EE

Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4; traducido informalmente como Java Empresarial), es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en

componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma Java EE está definida por una especificación. Similar a otras especificaciones del Java CommunityProcess, Java EE es también considerada informalmente como un estándar debido a que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE; estandarizado por The Java CommunityProcess / JCP.

Java EE tiene varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans, servlets, portlets (siguiendo la especificación de Portlets Java), JavaServerPages y varias tecnologías de servicios web. Ello permite al desarrollador crear una aplicación de empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel (Wikipedia, 2013).

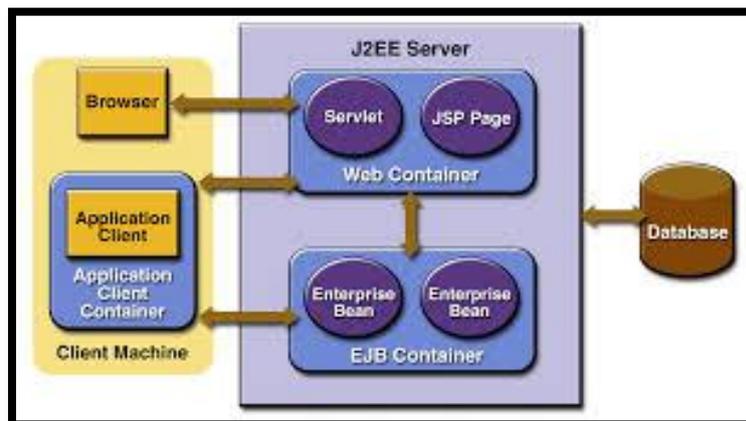


Fig. 3. Elementos de J2EE

(<http://docs.oracle.com/javaee/1.3/tutorial/doc/Overview4.html>)

1.9. Framework

La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio (Wikipedia, 2013).

1.10. Patrón MVC (Modelo Vista Controlador)

El patrón de desarrollo de software MVC (Model -View - Controller) define la organización independiente de los Objetos de Negocio (Model), la interfaz con el usuario u otro sistema (View) y el controlador del workflow de la aplicación (Controller) quien expresará: "si estoy aquí y me piden esto entonces hacer tal cosa, si sale bien mostrar esto y si no lo aquello otro").

Modelo

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe reflejar. Es la parte encargada de representar la lógica de negocio de una aplicación. Así, por ejemplo, un sistema de administración de datos geográficos tendrá un modelo que representara la altura, coordenadas de posición, distancia, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

Vista

Las vistas son las encargadas de la representación de los datos, contenidos en el modelo, al usuario. La relación entre las vistas y el modelo son de muchas a uno, es decir cada vista se asocia a un modelo, pero pueden existir muchas vistas asociadas al mismo modelo. De esta manera, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

La vista solo necesita la información requerida del modelo para realizar un despliegue. Cada vez que se realiza una actuación, que implica una modificación del modelo de dominio, la vista cambia a través de notificaciones generadas por el modelo de la

aplicación. Sencillamente, es la representación visual del modelo que redibuja las partes necesarias cuando se produce una modificación del mismo.

Controlador

El controlador es el encargado de interpretar y dar sentido a las instrucciones que realiza el usuario, realizando actuaciones sobre el modelo. Si se realiza algún cambio, comienza a actuar, tanto si la modificación se produce en una vista o en el modelo. Interactúa con el modelo a través de una referencia al propio modelo.

¿Cómo funciona MVC en aplicaciones Web?

1. Captura de la petición en el controlador: La aplicación recibe peticiones que son centralizadas en el Controlador. Éste es el encargado de interpretar, a partir de la URL de la solicitud, el tipo de operación que hay que realizar. Normalmente, esto se hace analizando el valor de algún parámetro que se envía anexando a la URL de la petición y que se utiliza con esta finalidad.

2. Procesamiento de la petición: Una vez que el Controlador determine la operación a realizar, procede a ejecutar las acciones pertinentes, invocando para ello a los diferentes métodos expuestos por el Modelo.

Dependiendo de las acciones a realizar (por ejemplo, un alta de un usuario en el sistema), el Modelo necesitará manejar los datos enviados por el cliente en la petición, datos que le serán proporcionados por el controlador. De la misma manera, los resultados generados por el Modelo (por ejemplo la información resultante de una búsqueda serán entregados directamente al controlador).

3. Generación de respuestas: Los resultados devueltos por el Modelo al Controlador son depositados por éste en una variable de petición, sesión o aplicación, según el alcance que deban tener. A continuación, el Controlador invoca a la página JSP que debe encargarse de generar la vista correspondiente, esta página accederá a la variable de ámbito donde estén depositados los resultados y los utilizará para generar dinámicamente la respuesta XHTML que será enviada al cliente (<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>).

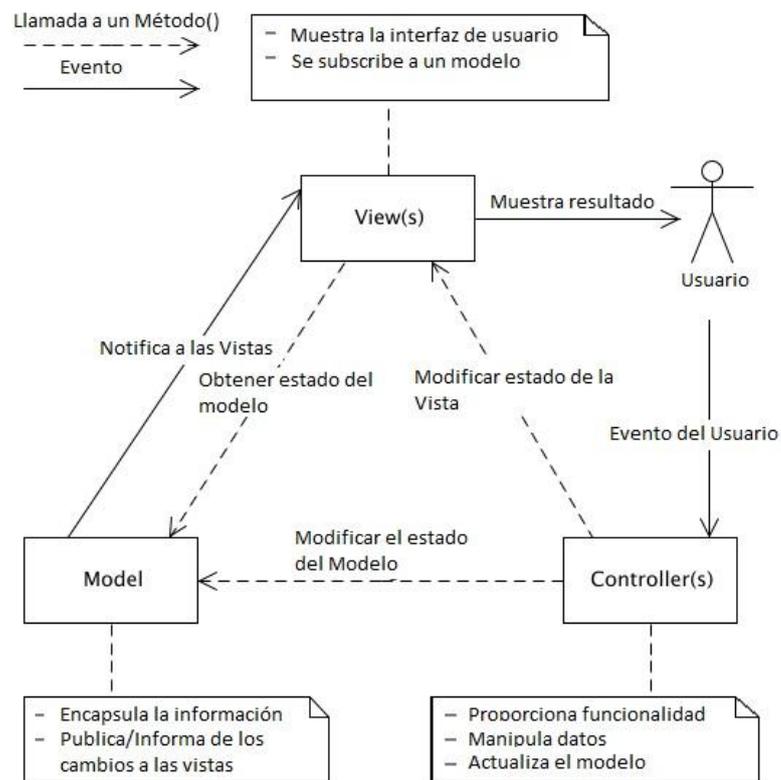


Fig. 4. Patrón MVC
 (<http://lbcnet.wordpress.com>)

1.11. Framework Spring

Spring es un framework liviano y no intrusivo: generalmente los objetos que programamos no tienen dependencias en clases específicas de Spring. Sus características principales son inyección de dependencias y programación orientada a aspectos.

Módulos de Spring

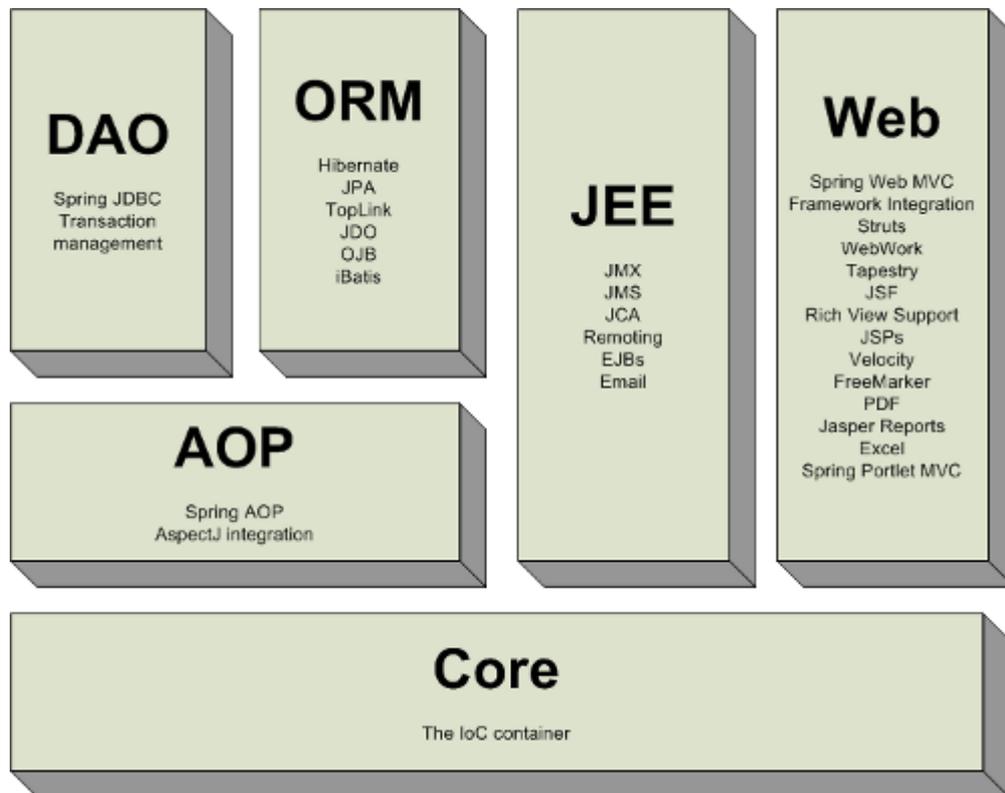


Fig. 5. Módulos de Spring

El diagrama muestra los módulos con los que cuenta Spring (hasta la versión 2.5). En su núcleo (Core) se encuentra el BeanFactory – el contenedor fundamental de Spring y quien se encarga de la inyección de dependencias.

El contenedor ApplicationContext se basa en BeanFactory y extiende su funcionalidad con soporte para i18n, eventos de ciclo de vida, validación y mejor integración con AOP.

- **AOP** – provee la implementación de AOP, permitiéndonos desarrollar interceptores de método y puntos de corte para desacoplar el código de las funcionalidades transversales.
- **DAO** - Provee una capa de abstracción sobre JDBC, abstrae el código de acceso a datos de una manera simple y limpia. Tiene una capa de excepciones sobre los mensajes de error provistos por cada servidor específico de base de datos. Además cuenta con manejo de transacciones a través de AOP.
- **ORM** – Provee la integración para las distintas APIs de mapeo objeto-relacional incluyendo JPA, JDO, Hibernate e iBatis.

- **JEE** – Provee integración con aplicaciones Java Enterprise Edition así como servicios JMX, JMS, EJB, etc.
- **Web** – Módulo que aporta clases especiales orientadas al desarrollo web e integración con tecnologías como Struts y JSF. Cuenta con el paquete Spring MVC, una implementación del conocido patrón de diseño aplicando los principios de Spring (Fernando Briano, 2010).

1.12. Richfaces

RichFaces es una librería de componentes visuales para JSF, escrita en su origen por Exadel y adquirida por Jboss. Además, RichFaces posee un framework avanzado para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF.

Son características de RichFaces las siguientes:

- Se integra perfectamente en el ciclo de vida de JSF,
- Incluye funcionalidades Ajax, de modo que nunca vemos el JavaScript y tiene un contenedor Ajax propio,
- Contiene un set de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas nuestras necesidades,
- Soporta facelets,
- Soporta css themes o skins,
- Es un proyecto open source, activo y con una comunidad también activa.
- Usando Ajax4JSF tenemos que indicar qué parte de la pantalla tiene que repintarse. No es tan simple como ICEfaces, pero implica tener más control sobre los eventos que se producen en la interfaz de usuario.
- En las últimas versiones siempre se les cuela alguna "peora", que merma la funcionalidad de algún componente y donde, por ejemplo, funcionaba la subida de ficheros mediante un componente JSF con barra de progreso en Internet Explorer, ahora solo funciona en Firefox. Aunque también es cierto que se detecta y soluciona en la siguiente versión (José Sánchez, 2010).

1.13. Hibernate

Hibernate es una herramienta de Mapeo Objeto-Relacional (ORM) para la plataforma Java y disponible también para .Net con el nombre de NHibernate que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una

aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (HibernateQueryLanguage), al mismo tiempo que una API para construir las consultas programáticamente. Hibernate fue una iniciativa de un grupo de desarrolladores dispersos alrededor del mundo conducidos por Gavin King. Tiempo después, JBoss Inc. contrató a los principales desarrolladores de Hibernate y trabajó con ellos en brindar soporte al proyecto (Wikipedia, 2013).

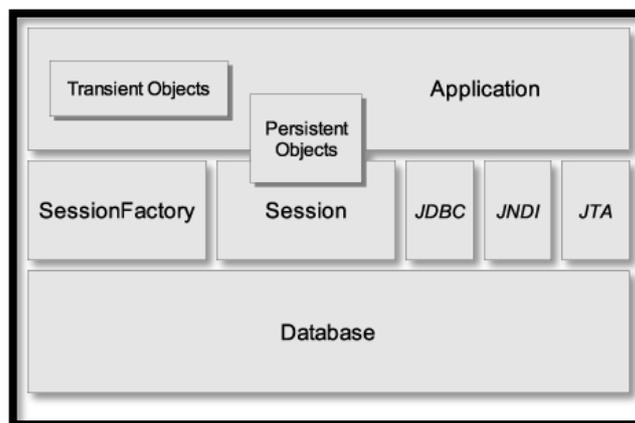


Fig. 6. Arquitectura de Hibernate
(<http://joeljil.wordpress.com>)

1.14. Oracle

1.14.1. Características principales de Oracle.

Oracle es un Sistema Gestor de Bases de Datos con características objeto-relacionales, que pertenece al modelo evolutivo de SGBD. Sus características principales son las siguientes:

- Entorno cliente/servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Disponibilidad controlada de los datos de las aplicaciones.
- Adaptación a estándares de la industria, como SQL-92.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Conectabilidad.
- Replicación de entornos.

1.14.2. Definición de base de datos e instancia de oracle.

Una base de datos Oracle es una colección de datos tratada como una unidad. El propósito general es almacenar y recuperar información relacionada.

Una instancia Oracle consta de una estructura de memoria, llamada Área Global del Sistema (SGA), y de unos procesos background utilizados por el servidor Oracle para manejar una base de datos. Cada instancia Oracle puede abrir y utilizar sólo una base datos en cualquier punto y momento.

1.14.3. Estructura lógica de Oracle.

La estructura lógica de Oracle viene determinada por:

- Uno o más tablespaces (área lógica de almacenamiento).
- Los objetos de los distintos esquemas (colecciones de objetos).

Las estructuras de almacenamiento lógico, incluyendo tablespaces, segmentos y extensiones, dictan la forma de utilización del espacio físico. Los objetos del esquema y las relaciones entre ellos forman el diseño relacional de la base de datos.

Tablespaces

Una base de datos se divide en unidades de almacenamiento lógico llamadas tablespaces. Un tablespace se usa para agrupar estructuras lógicas relacionadas.

Esquemas y objetos del esquema

Un esquema es una colección de objetos de la base de datos. Los objetos del esquema son estructuras lógicas que hacen referencia directa a datos de la base de datos (tablas, vistas, secuencias, procedimientos almacenados, sinónimos, índices, clusters y enlaces con otras bases de datos).

Bloques de datos, extensiones y segmentos

Los bloques de datos Oracle son las estructuras lógicas de más bajo nivel. Una base de datos se almacena en bloques de datos, que se corresponden con un número específico de bytes de espacio en disco. Una extensión es un número específico de bloques de datos contiguos, que se emplea para almacenar un tipo de información específico. Un segmento es un conjunto de extensiones reservadas para una determinada estructura lógica (por ejemplo, segmentos de datos o segmentos de índices).

1.14.4. Estructura física de Oracle.

La estructura física está formada por los ficheros del sistema operativo que constituyen la base de datos. Todas las base de datos Oracle constan de tres tipos de ficheros: uno o más ficheros de datos, uno o más ficheros de redo log y uno o más ficheros de control. Proveen el almacenamiento físico de la información de la base de datos.

Ficheros de datos

Los ficheros de datos contienen los datos de la base de datos. Tienen las siguientes características:

- Un fichero sólo puede estar asociado con una base de datos.

- Los ficheros de datos tienen atributos que permiten reservar automáticamente para ellos extensiones cuando se acaba el espacio.
- Uno o más ficheros de datos forman una unidad lógica de almacenamiento llamada tablespace.

Ficheros de redo log

El conjunto de ficheros de redo log se denomina Redo Log de la base de datos. La función principal de Redo Log es guardar todos los cambios hechos a los datos. Son críticos en la protección de la base de datos ante fallos; se utilizan en caso de fallo para recuperar la base de datos.

Ficheros de control

Un fichero de control contiene entradas que especifican la estructura física de la base de datos. En particular contiene:

- El nombre de la base de datos.
- Los nombres y localización de los ficheros de la base de datos y los ficheros de redo log.
- Fecha de creación de la base de datos.
- Información de sincronización para el proceso de recuperación de la base de datos.

1.14.5. Seguridad para Oracle.

El servidor Oracle provee de control de accesos discrecional, es decir, acceso restringido a la información basado en privilegios.

Oracle gestiona la seguridad de la base de datos usando:

- Usuarios y esquemas de la base de datos.
- Privilegios.
- Roles.
- Ajustes de rendimiento y cuotas.
- Límites sobre los recursos.
- Auditoría.

Cada usuario tiene un dominio de seguridad, que determina cosas como:

- Acciones (privilegios y roles) disponibles para el usuario.
- Cuotas sobre tablespaces.
- Límites en los recursos del sistema.

1.14.6. Estructuras usadas para la recuperación en Oracle.

Oracle posee varias estructuras y mecanismos software para proveer:

- Recuperación de la base de datos ante distintos tipos de fallos.
- Operaciones de recuperación flexibles.
- Disponibilidad de los datos durante las operaciones de backup y recovery.

Oracle usa varias estructuras para proveer la recuperación completa de la instancia:

- el Redo Log,
- los segmentos de rollback,
- un fichero de control,
- las copias necesarias de la base de datos (Roberto Velasco, 2013).

1.15. Servidor Web Apache

El servidor http apache es un servidor web http de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo http/1.1² y la noción de sitio virtual.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. Apache es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphereapplication server. Mac OS X integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

Apache es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. Un usuario que tiene Apache instalado en su escritorio puede colocar arbitrariamente archivos en la raíz de documentos de Apache, desde donde pueden ser compartidos.

Los programadores de aplicaciones web a veces utilizan una versión local de Apache con el fin de previsualizar y probar código mientras éste es desarrollado.

Microsoft Internet InformationServices (IIS) es el principal competidor de Apache, así como Sun Java System Web Server de Sun Microsystems y un anfitrión de otras aplicaciones como Zeus Web Server. Algunos de los más grandes sitios web del mundo están ejecutándose sobre Apache. La capa frontal (frontend) del motor de búsqueda Google está basada en una versión modificada de Apache, denominada Google Web Server (GWS). Muchos proyectos de Wikimedia también se ejecutan sobre servidores web Apache.

a. Ventajas

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)

b. Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web.

Algunos de estos módulos son:

- mod_ssl - Comunicaciones Seguras vía TLS.
- mod_rewrite - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así

engañar a los navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).

- `mod_dav` - Soporte del protocolo WebDAV (RFC 2518).
- `mod_deflate` - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- `mod_auth_ldap` - Permite autenticar usuarios contra un servidor LDAP.
- `mod_proxy_ajp` - Conector para enlazar con el servidor JakartaTomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- `mod_cband` - Control de tráfico y limitador de ancho de banda.
- `mod_perl` - Páginas dinámicas en Perl.
- `mod_php` - Páginas dinámicas en PHP.
- `mod_python` - Páginas dinámicas en Python.
- `mod_rexx` - Páginas dinámicas en REXX y Object REXX.
- `mod_ruby` - Páginas dinámicas en Ruby.
- `mod_aspdotnet` - Páginas dinámicas en .NET de Microsoft (Módulo retirado).
- `mod_mono` - Páginas dinámicas en Mono
- `mod_security` - Filtrado a nivel de aplicación, para seguridad (Wikipedia, 2013).

CAPITULO II. DESARROLLO DE LA TESIS

2.1. Plan de desarrollo del proyecto

2.1.1. Estudio Preliminar del Proyecto

A. Descripción de la Empresa

Certicom, una empresa ubicada en *Av. Canaval y Moreyra N°452, 5 piso*, con más de 15 años de experiencia especializada en apoyar a sus clientes a crecer, tiene como misión proveer servicios de tercerización en la gestión y operación de procesos creando valor a través del uso intensivo de las tecnologías, la ingeniería y la innovación, orientado al mejoramiento de los tiempos y costos y optimizando la eficiencia.

B. Estructura de la Empresa

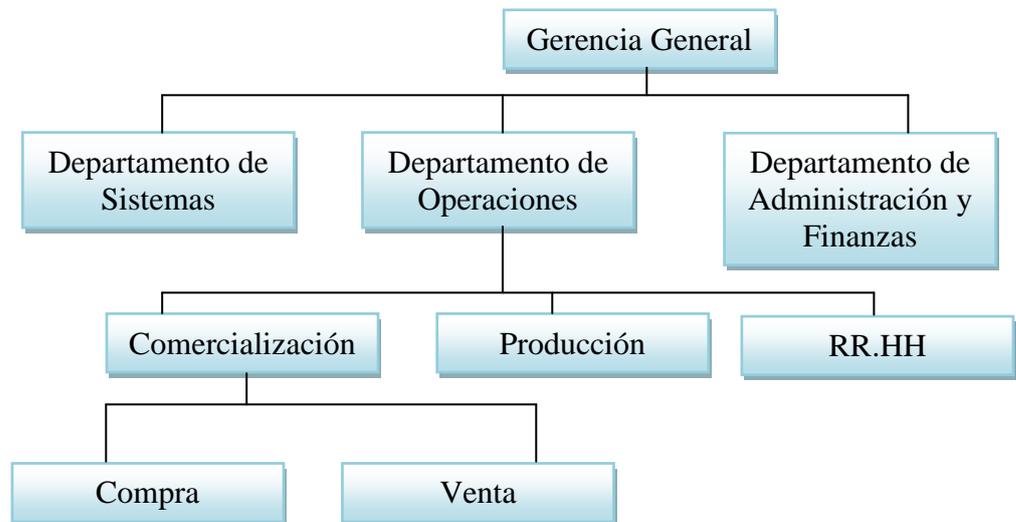


Fig. 7. Organigrama de la Empresa

2.1.2. Propósito del proyecto

La solución informática a desarrollar va a permitir manejar adecuadamente la Gestión de Compras de la empresa CERTICOM S.A.C. Este sistema informático web debe también permitir registrar todos los procesos de las operaciones por parte de logística y empleados de la empresa, realizar procesos de reportes y consultas que exploren la información oficial que existen en la empresa que ayuden a la toma de decisiones para la empresa.

2.1.3. Hitos del proyecto

Tabla 2 Hitos del Proyecto

Componentes	
Fases	Hitos
Fase1 : Gestión de Proyectos	Plan de Trabajo a realizar. Estudio Preliminar del trabajo a realizar. Desarrollo del informe del trabajo a realizar.
Fase2: Análisis de Requisitos	Prototipo del sistema completo a realizar. Diagrama de casos de uso a realizar. Modelo de Dominio a realizar.
Fase2: Análisis y Diseño Preliminar	Diagrama de Robustez a realizar. Especificación de los requerimientos a realizar. Actualización del diagrama de dominio a realizar.
Fase2: Diseño Detallado	Diagrama de Secuencia a realizar. Actualización del diagrama de dominio a realizar.
Fase2: Implementación	Diagrama de Despliegue a realizar. Diagrama de Componentes a realizar. Codificación y realización de pruebas a realizar.

2.1.4. Requerimientos Funcionales y No funcionales del proyecto

- **Requerimientos Funcionales**
 - Registrar categoría de productos, también se permitirá actualizarlos

- Registrar sub-categoría de productos, también se permitirá actualizarlos
 - Registrar productos, también se permitirá actualizarlos
 - Registrar rubro, también se permitirá actualizarlos.
 - Registrar proveedor, también se permitirá actualizarlos.
 - Registrar proyecto Padre, también se permitirá actualizarlos.
 - Registrar proyecto, también se permitirá actualizarlos.
 - Registrar departamento, también se permitirá actualizarlos.
 - Registrar usuario, también se permitirá actualizarlos.
 - Asignar opciones a usuario.
 - Registrar rol, también se permitirá actualizarlos.
 - Asignar opciones a rol.
 - Registrar requerimiento y su detalle.
 - Aprobar requerimientos generados por los usuarios finales.
 - Cotizar los requerimientos aprobados.
 - Seleccionar y confirmar la cotización ganadora por requerimiento.
 - Aprobar la cotización ganadora y generar la orden de compra.
 - Generar reporte de compra de productos por proyecto y fecha.
- **Requerimientos No Funcionales**
 - Contar con un manual de usuario.
 - El tiempo de respuesta del sistema informático no mayor a 2 segundos.
 - Contar con cuadros de diálogo para la mejor manipulación de los procesos.
 - Navegación fácil vía teclado y/o mouse.
 - Correrá bajo el browser Mozilla Firefox.
 - Multiplataforma (Windows XP / vista)
 - La disponibilidad del sistema debe de ser las 24 horas del día durante los 365 días del año.

2.1.5. Restricciones y Supuestos del proyecto

- **Restricciones**
 - El software será desarrollado utilizando la integración de los frameworks Spring, Richface e Hibernate.
 - El gestor de la base de datos será Oracle.
 - El desarrollo del sistema utilizará la metodología Iconix.

- **Supuestos**
 - Se dispone de los recursos económicos necesarios para ejecutar el proyecto.
 - Existe el compromiso del asesor y miembros del equipo de trabajo en desarrollar el sistema informático web.

2.2. Análisis de Requisitos

2.2.1. Prototipo del sistema completo.

Mantenimiento Categoría

Nombre

Estado ▼

Nombre	Estado	Acciones
Equipos de computo	H	Editar Eliminar
Suministros	H	Editar Eliminar
Servicios	H	Editar Eliminar

Fig. 8. Prototipo Mantenimiento Categoría de Producto

Mantenimiento Sub-Categoría

Nombre

Categoría ▼

Estado ▼

Nombre	Categoría	Estado	Acciones
Papelería	Suministros	H	Editar Eliminar
Oficina	Suministros	H	Editar Eliminar
Productos de Limpieza	Suministros	H	Editar Eliminar

Fig. 9. Prototipo Mantenimiento Sub-Categoría de Producto

Mantenimiento Producto

Nombre

Sub-Categoría

Tipo Producto

Precio Coste

Estado

Nombre	Sub-Categoría	Tipo Producto	Precio Coste	Estado	Acciones
Papelote	Papelería	Consumible	20.0	H	Editar Eliminar
Papel Bond A4 75gr Chamex	Papelería	Consumible	18.0	H	Editar Eliminar
Papel a color	Papelería	Consumible	16.0	H	Editar Eliminar

Fig. 10. Prototipo Mantenimiento de Producto

Mantenimiento Proyecto Padre

Nombre

Estado

Nombre	Estado	Acciones
Gastos de Administración	H	Editar Eliminar
Scotiabank	H	Editar Eliminar
Kursa	H	Editar Eliminar

Fig. 11. Prototipo Mantenimiento Proyecto Padre

Mantenimiento Departamento

Nombre

Departamento Superior

Estado

Nombre	Dep. Superior	Estado	Acciones
Logística	Gerencia General	H	Editar Eliminar
Sistemas	Gerencia General	H	Editar Eliminar
Tesorería	Contabilidad	H	Editar Eliminar

Fig. 12 Prototipo Mantenimiento Departamento

Mantenimiento Proyecto

Nombre

Proyecto Padre

Departamento

Estado

Nombre	Proyecto Padre	Departamento	Estado	Acciones
BBVA	Gastos de Administración	Sistemas	H	Editar Eliminar
Sistemas	Gastos de Administración	Sistemas	H	Editar Eliminar
Kursa	Gastos de Administración	Sistemas	H	Editar Eliminar

Fig. 13 Prototipo Mantenimiento Proyecto

Mantenimiento Rubro

Nombre

Estado

Nombre	Estado	Acciones
Banca	H	Editar Eliminar
Telefonia	H	Editar Eliminar
Suministros	H	Editar Eliminar

Fig. 14 Prototipo Mantenimiento Rubro

Mantenimiento Proveedor

Es una empresa?

Nombre

Tipo documento Num. Documento

Rubro

Dirección

Correo

Teléfono

Ciudad

Estado

Nombre	Estado	Acciones
Acovalco	H	Editar Eliminar
Tailoy	H	Editar Eliminar
Telefonica	H	Editar Eliminar

Fig. 15 Prototipo Mantenimiento Proveedor

Registrar Usuario

Nombre

Apellido Paterno

Apellido Materno

Perfil

Tipo Documento Usuario

Correo Teléfono

Departamento

Es Jefe?

Fecha Ingreso Fecha Nacimiento

Estado

Código	Apellido Paterno	Apellido Materno	Nombres	Estado	Acciones	Opciones de Menú
12345678	Falcetta	Saladino	Giuseppe	H	Editar Eliminar	Asignar Opciones
46090495	Wilson	Carbajal	Dennis Martín	H	Editar Eliminar	Asignar Opciones
ADMINGC	Díaz	Flores	Lázaro	H	Editar Eliminar	Asignar Opciones

Fig. 16 Prototipo Registro Usuario

Opciones por Usuario

Documento 46090495

Nombre Wilson Carbajal, Dennis

Perfil

Disponibles Seleccionadas

»»

»

«

««

Fig. 17 Prototipo Asignar Opciones a Usuario

Mantenimiento Rol

Nombre

Descripción

Estado

Nombre	Estado	Acciones	Columna 1
Logística	H	Editar Eliminar	Asignar Opciones
Jefe de Proyecto	H	Editar Eliminar	Asignar Opciones
Usuario Final	H	Editar Eliminar	Asignar Opciones

Fig. 18 Prototipo Registrar Rol

Opciones de Menú por Perfil

Perfil

Descripción

Disponibles

Seleccionadas

»»
»
«
««

Fig. 19 Prototipo Asignar Opciones a Rol

Registrar Requerimiento

Datos de Requerimiento

Número: 1

Responsable: Dennis Wilson Fecha:

Proyecto: Jefe: Giuseppe Falcetta

Detalle de Producto

Producto	Precio	Seleccionar
<input type="text"/>		
Cloro	10.0	<input type="radio"/>
Detergente	15.0	<input checked="" type="radio"/>
Limpia Vidrios	8.0	<input type="radio"/>

Cantidad:

Detalle Requerimiento

Producto	Precio	Cantidad	Acción
Cloro	10.0	10	Eliminar
Detergente	15.0	10	Eliminar
Limpia Vidrios	8.0	14	Eliminar

Fig. 20 Prototipo Registrar Requerimiento

Consultar Requerimientos

Listado de Requerimientos

#	N° Requerimiento	Responsable	Fecha	Jefe Proyecto	Estado	Ir Requerimiento
1	1	Dennis Wilson	2013-07-20	G. Falcetta	Nuevo	Ir Requerimiento

Fig. 21 Prototipo Consultar Requerimientos

Requerimiento por Aprobar

Datos de Requerimiento

Número 1

Responsable Dennis Wilson Fecha

Proyecto Jefe Giuseppe Falcetta

Detalle Requerimiento

Producto	Precio	Cantidad
Cloro	10.0	10
Detergente	15.0	10
Limpia Vidrios	8.0	14

Fig. 22 Prototipo Aprobar Requerimiento

Consultar Requerimientos Aprobados

Listado de Requerimientos

#	N° Requerimiento	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Ir Requerimiento
1	1	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Aprobado	Ir Requerimiento

Fig. 23 Prototipo Consultar Requerimientos Aprobados

Requerimiento por Cotizar

Datos de Requerimiento

Número: 1

Responsable: Dennis Wilson Fecha: 15/07/13

Proyecto: Sistemas Jefe: Giuseppe Falcetta

Detalle Requerimiento

Producto	Precio	Cantidad
Papel Bond A4 75Gr Chamex	10.0	15
Papelote	15.0	10

Nueva Cotización

Proveedor: Tailoy Fecha: 24/08/13

Listado de Detalle Requerimientos

#	Producto	Precio	Cantidad	IGV	Subtotal
1	Papel Bond A4 75Gr Chamex	16.0	15	18%	
2	Papelote	20.0	10	18%	

Subtotal:
 IGV:
 Total:

Fig. 24 Prototipo Cotizar Requerimiento

Consultar Requerimientos Cotizados

Listado de Requerimientos

#	N° Requerimiento	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Ir Requerimiento
1	1	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Cotizado	Ir Requerimiento

Fig. 25 Prototipo Consultar Requerimientos Cotizados

Requerimiento por Confirmar

Datos de Requerimiento

Número: 1

Responsable: Dennis Wilson Fecha: 15/07/13

Proyecto: Sistemas Jefe: Giuseppe Falcetta

Cotización

Proveedor	Total	Detalle	Seleccionar
Acovalco	50.0	Ir a detalle cot.	<input type="radio"/>
Tailoy	65.5	Ir a detalle cot.	<input type="radio"/>
Refricorp	48.2	Ir a detalle cot.	<input checked="" type="radio"/>

Detalle Cotización

Proveedor: Tailoy Fecha: 24/08/13

Listado de Detalle Requerimientos

#	Producto	Precio	Cantidad	IGV	Subtotal
1	Papel Bond A4 75Gr Chamex	16.0	15	18%	
2	Papelote	20.0	10	18%	

Subtotal:
IGV:
Total:

Fig. 26 Prototipo Confirmar Requerimiento

Consultar Requerimientos Confirmados

Listado de Cotizaciones

#	N° Requerimiento	Proveedor	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Ir Requerimiento
1	1	Refricorp	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Confirmado	Ir a Cotización
2	1	Acovalco	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Cancelado	Ir a Cotización
3	1	Tailoy	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Cancelado	Ir a Cotización

Fig. 27 Prototipo Consultar Requerimientos Confirmados

Cotización Ganadora

Proveedor Fecha

Listado de Detalle Requerimientos

#	Producto	Precio	Cantidad	IGV	Subtotal
1	Papel Bond A4 75Gr Chamex	16.0	15	18%	
2	Papelote	20.0	10	18%	

Subtotal:
IGV:
Total:

Fig. 28 Prototipo Aprobar Cotización

Consultar Requerimientos Aprobados

Listado de Cotizaciones

#	N° Requerimiento	N° Orden Compra	Proveedor	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Acción
1	1	OC0001	Refricorp	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Aprobado	Ir a Orden Compra
2	1	OC0002	Acovalco	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Cancelado	Ir a Orden Compra
3	1	OC0003	Talloy	Dennis Wilson	2013-07-20	G. Falcetta	Sistemas	Cancelado	Ir a Orden Compra

Fig. 29 Prototipo Consultar Requerimientos Aprobados

Orden de Compra

Orden de Compra: OC0001

Proveedor Fecha

Listado de Detalle Requerimientos

#	Producto	Precio	Cantidad	IGV	Subtotal
1	Papel Bond A4 75Gr Chamex	16.0	15	18%	
2	Papelote	20.0	10	18%	

Subtotal:
IGV:
Total:

Fig. 30 Prototipo Orden de Compra

Reporte de Órdenes de Compra por Proyecto y Fecha

Proyecto

Fecha

#	Orden Compra	Total	Proyecto	Fecha	Jefe Proyecto	Responsable
1	OC001	48.2	Sistemas	24/08/13	G. Falcetta	Dennis Wilson

Fig. 31 Prototipo Reporte de Órdenes de Compra por Proyecto y Fecha

2.2.2. Diagrama de casos de uso.

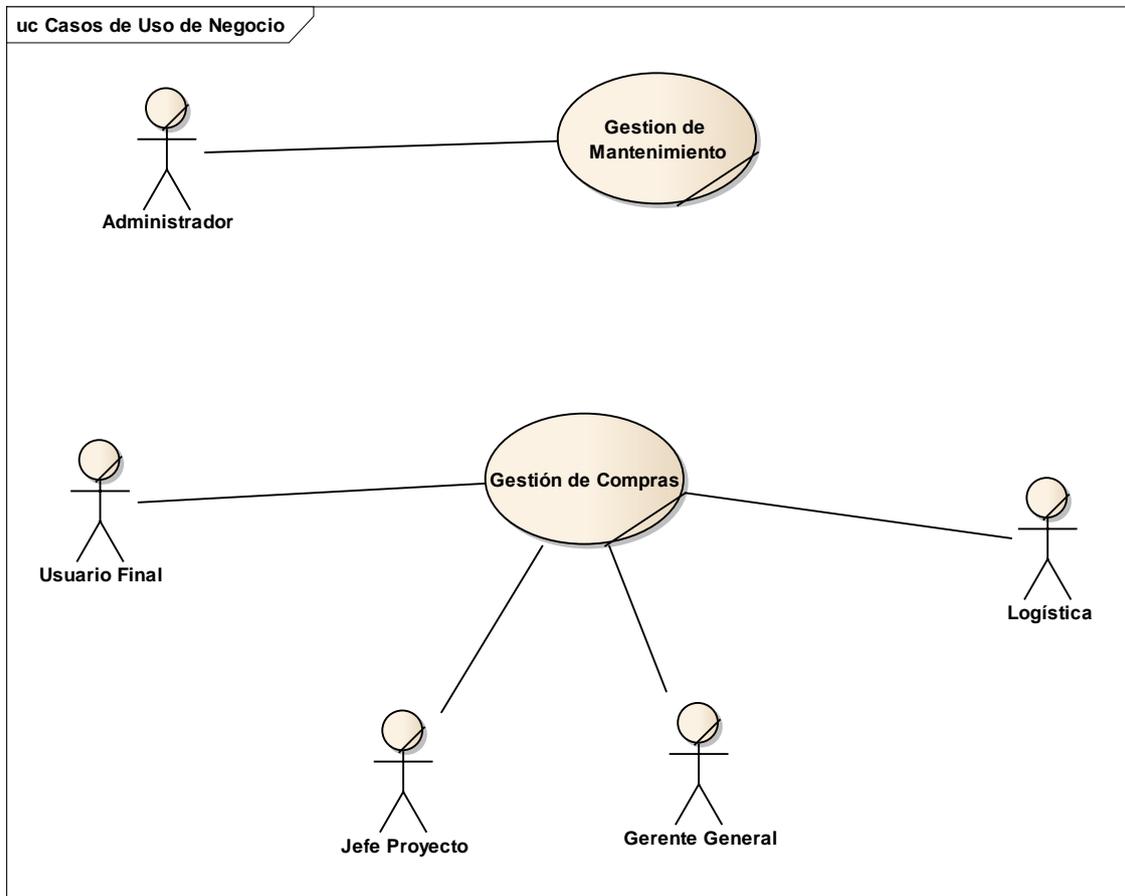


Fig. 32 Caso de Uso de Proceso del Negocio

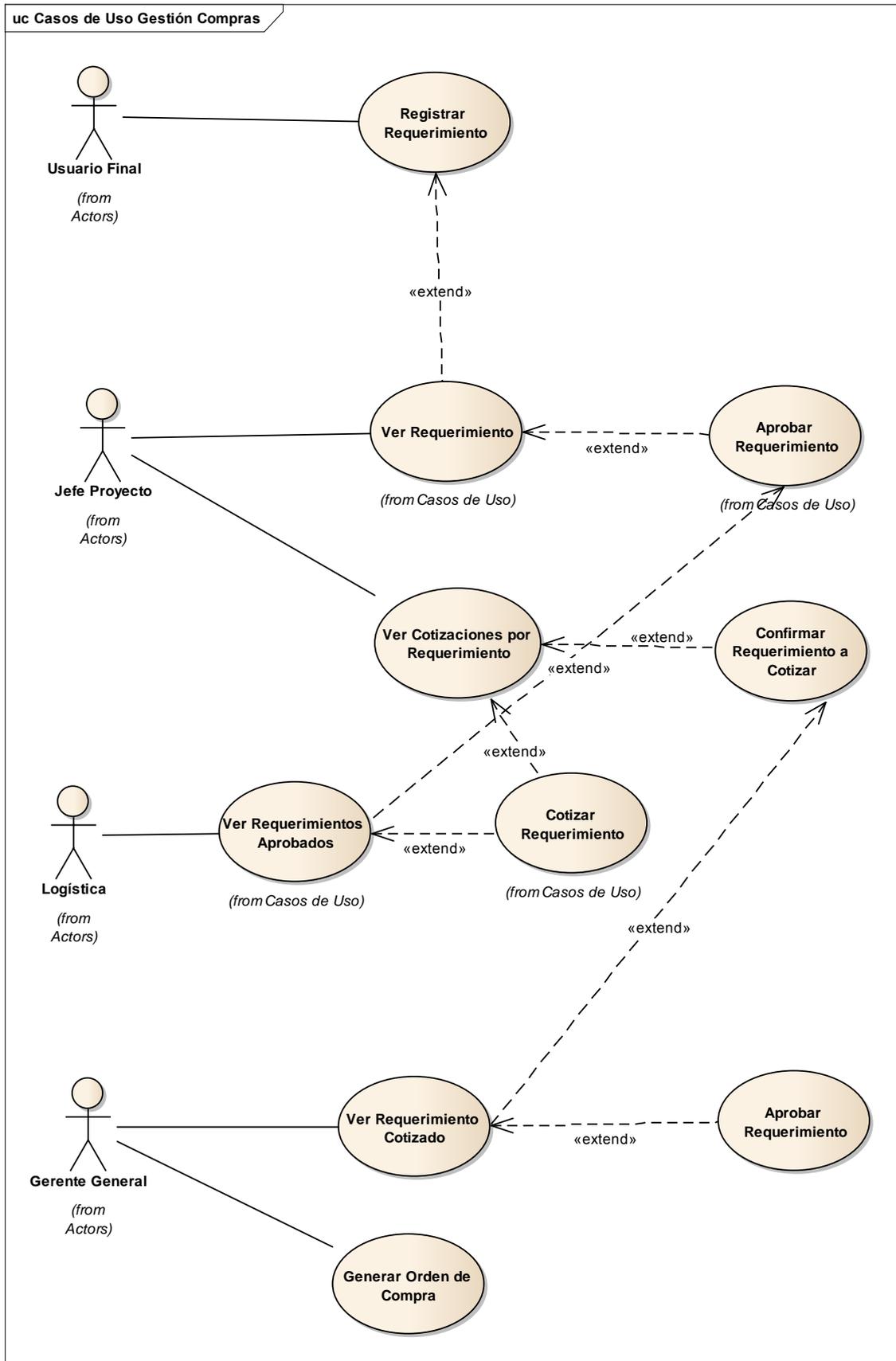


Fig. 33 Caso de Uso de Proceso de Gestión de Compras

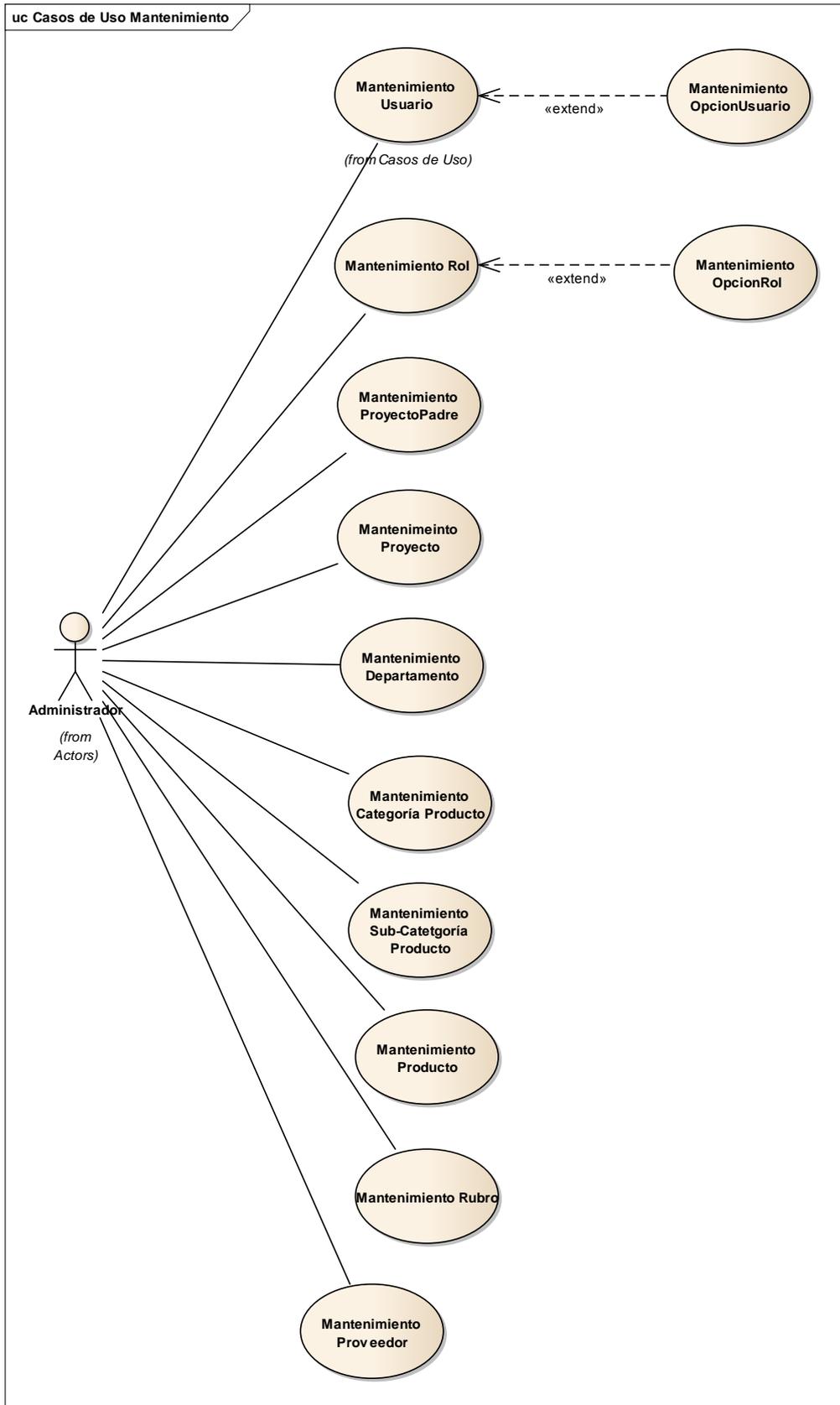


Fig. 34 Caso de Uso de Gestión de Mantenimiento de Información

2.2.3. Modelo de Dominio.

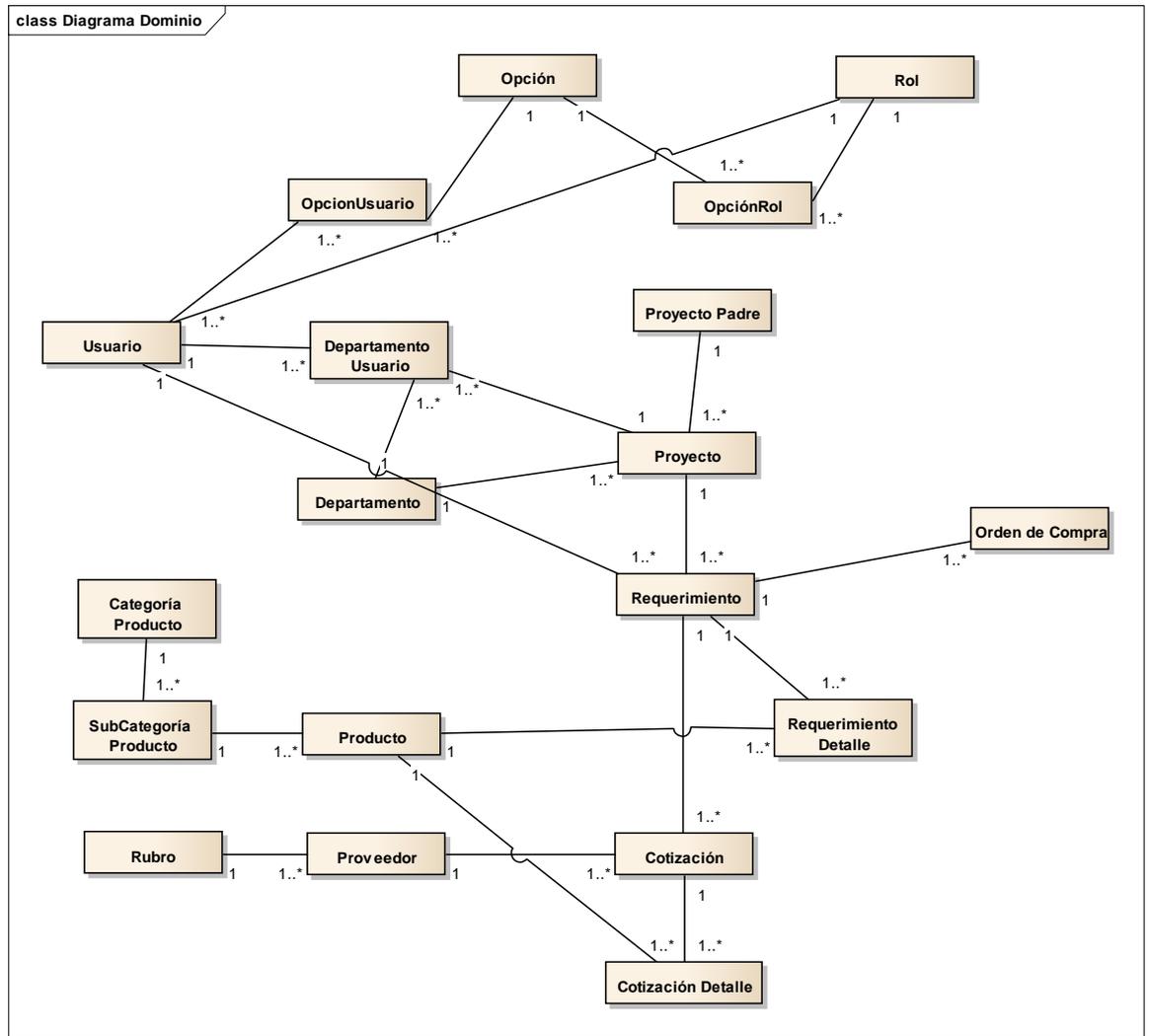


Fig. 35 Modelo de Dominio

2.3. Análisis y Diseño Preliminar

2.3.1. Diagrama de Clases.

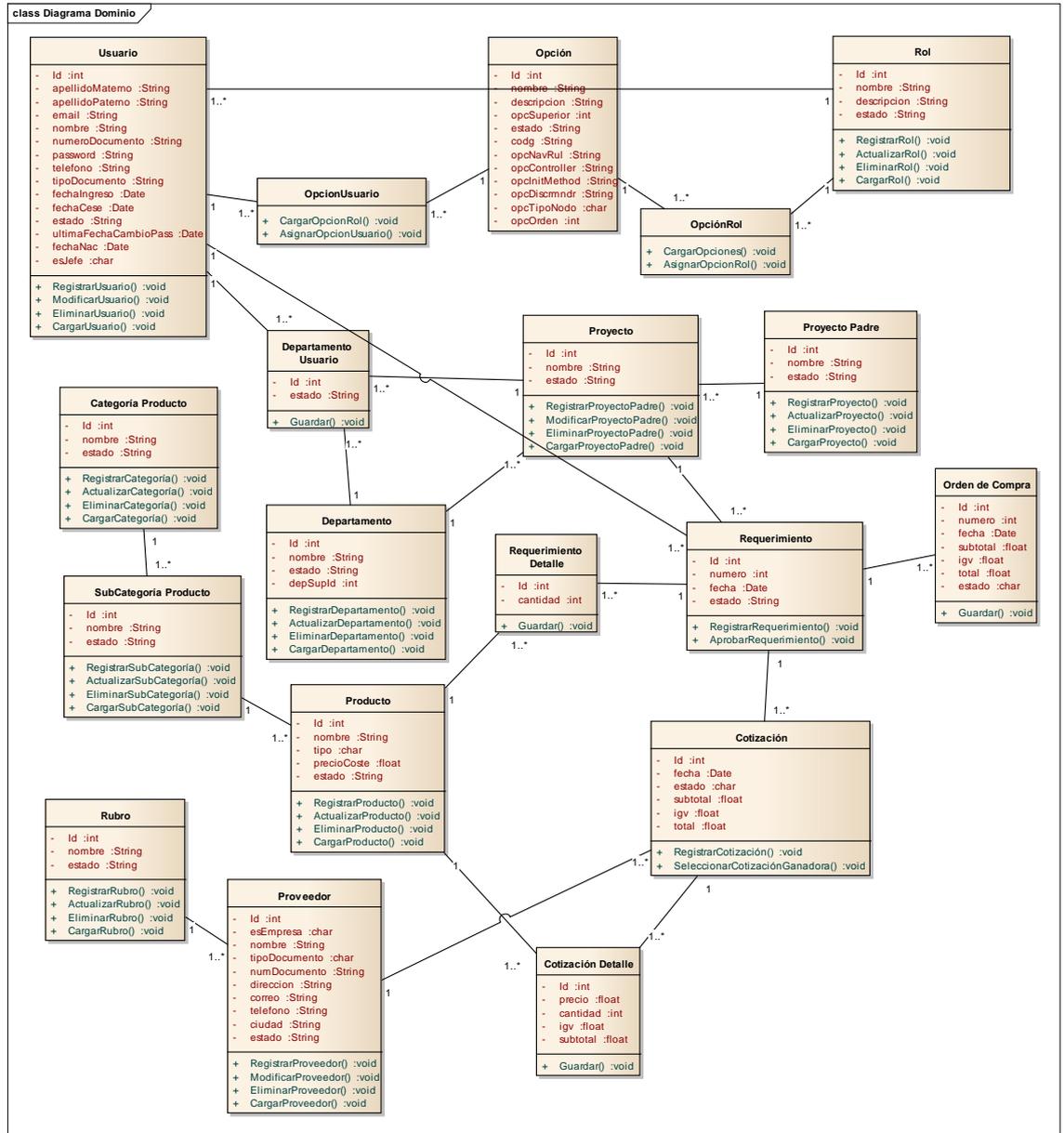


Fig. 36 Diagrama de Clases

2.3.2. Diagrama de Robustez.

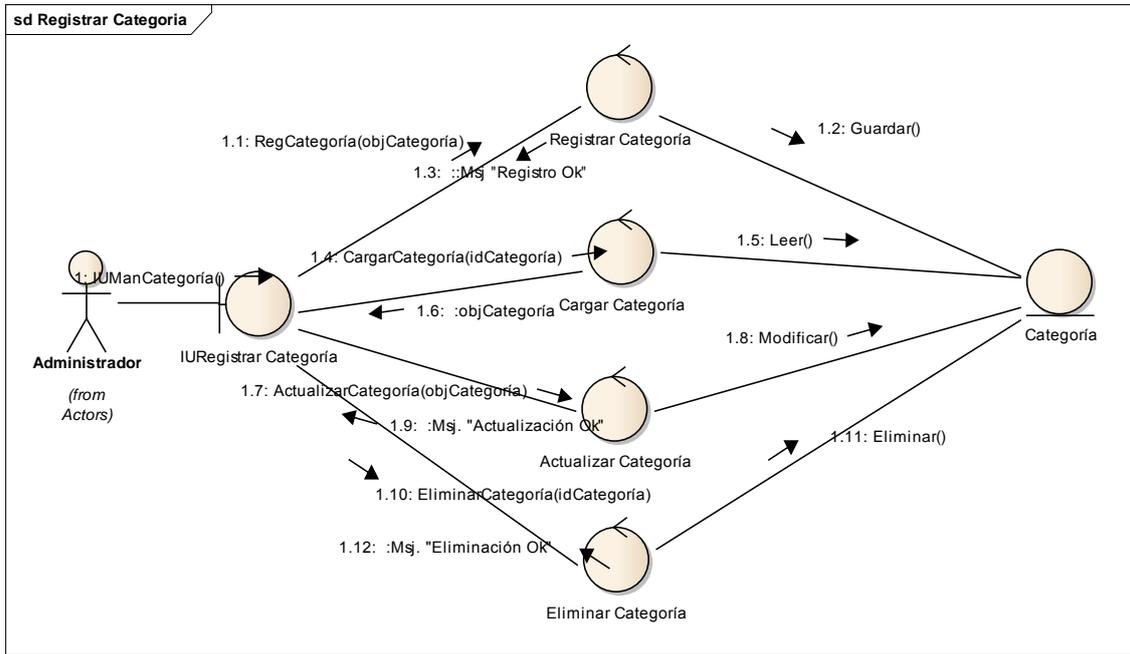


Fig. 37 Diagrama de Robustez Mantenimiento Categoría de Producto

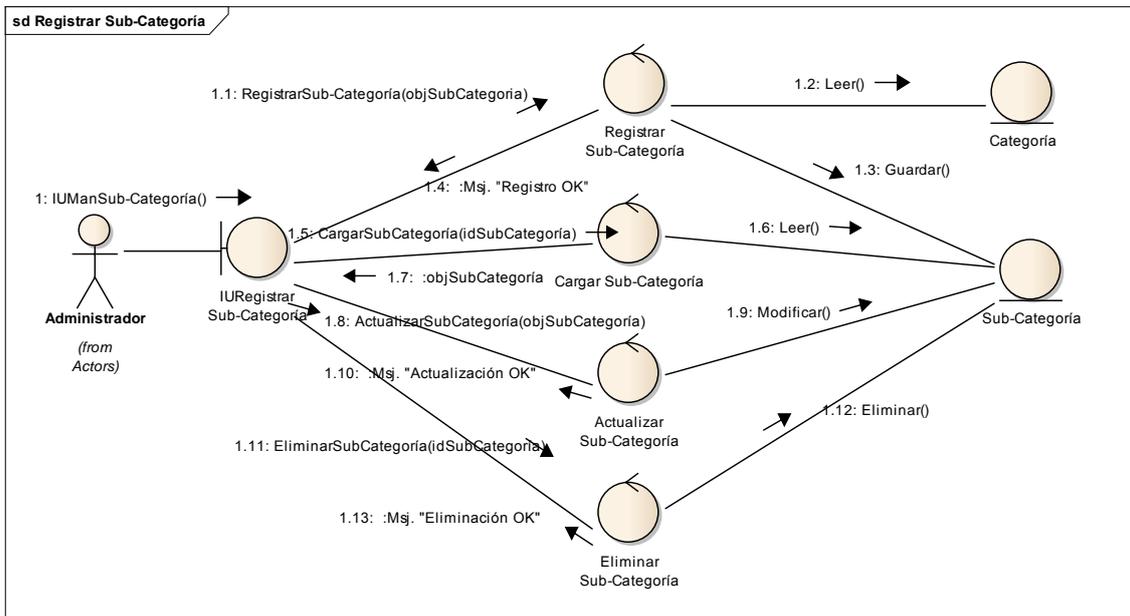


Fig. 38 Diagrama de Robustez Mantenimiento Sub-Categoría de Producto

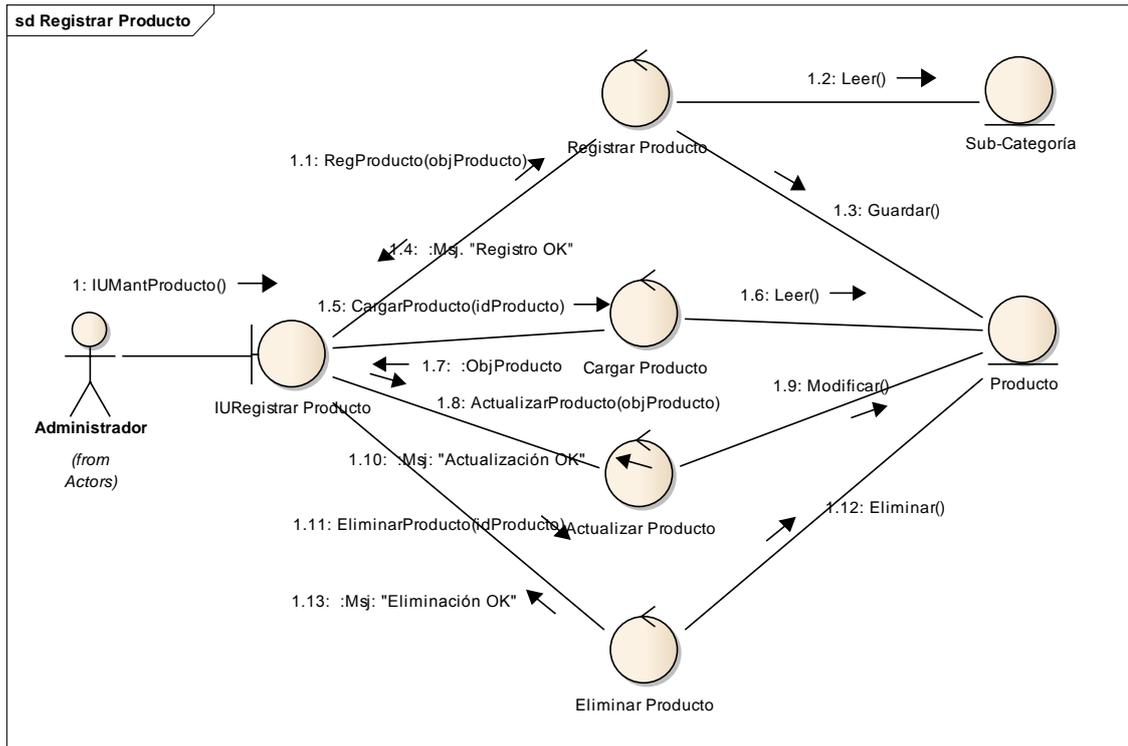


Fig. 39 Diagrama de Robustez Mantenimiento de Producto

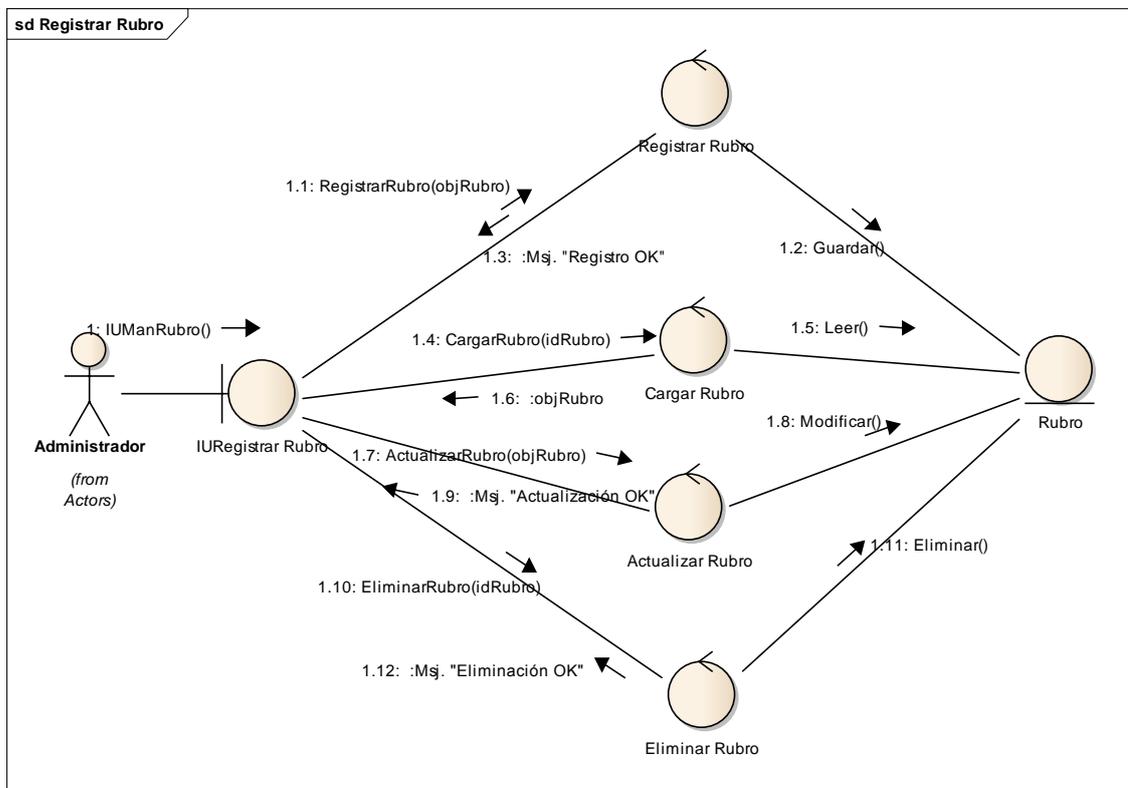


Fig. 40 Diagrama de Robustez Mantenimiento de Rubro

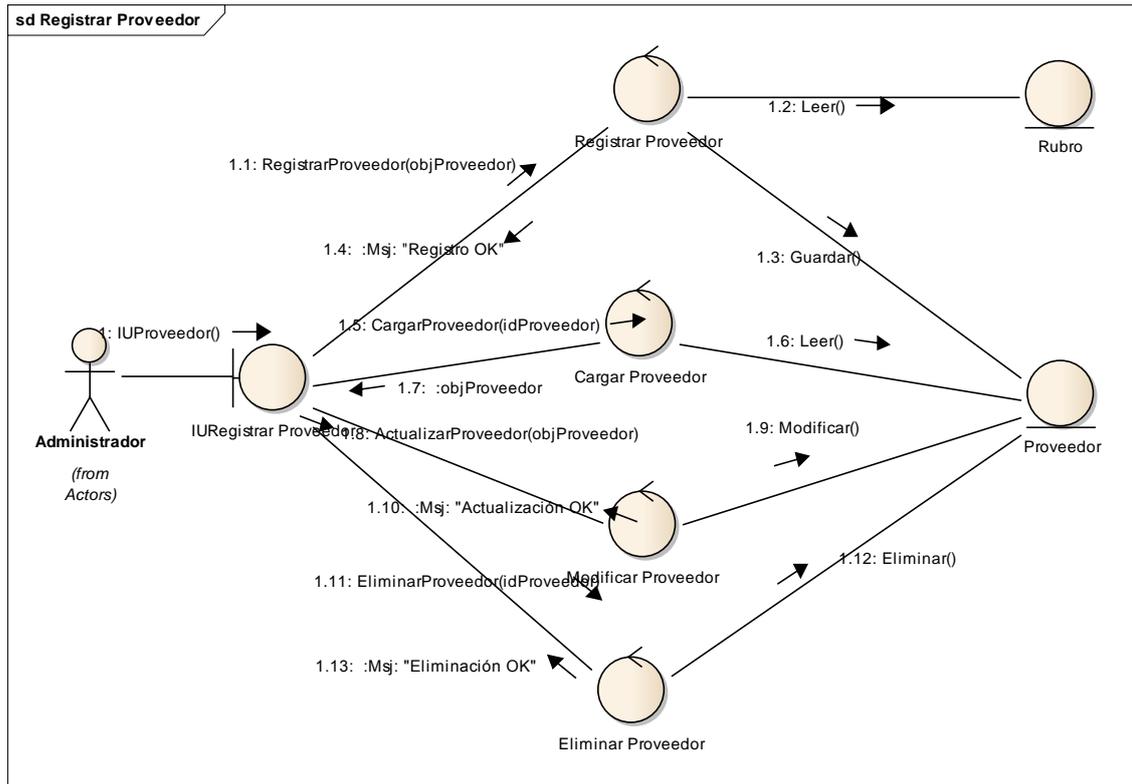


Fig. 41 Diagrama de Robustez de Mantenimiento de Proveedor

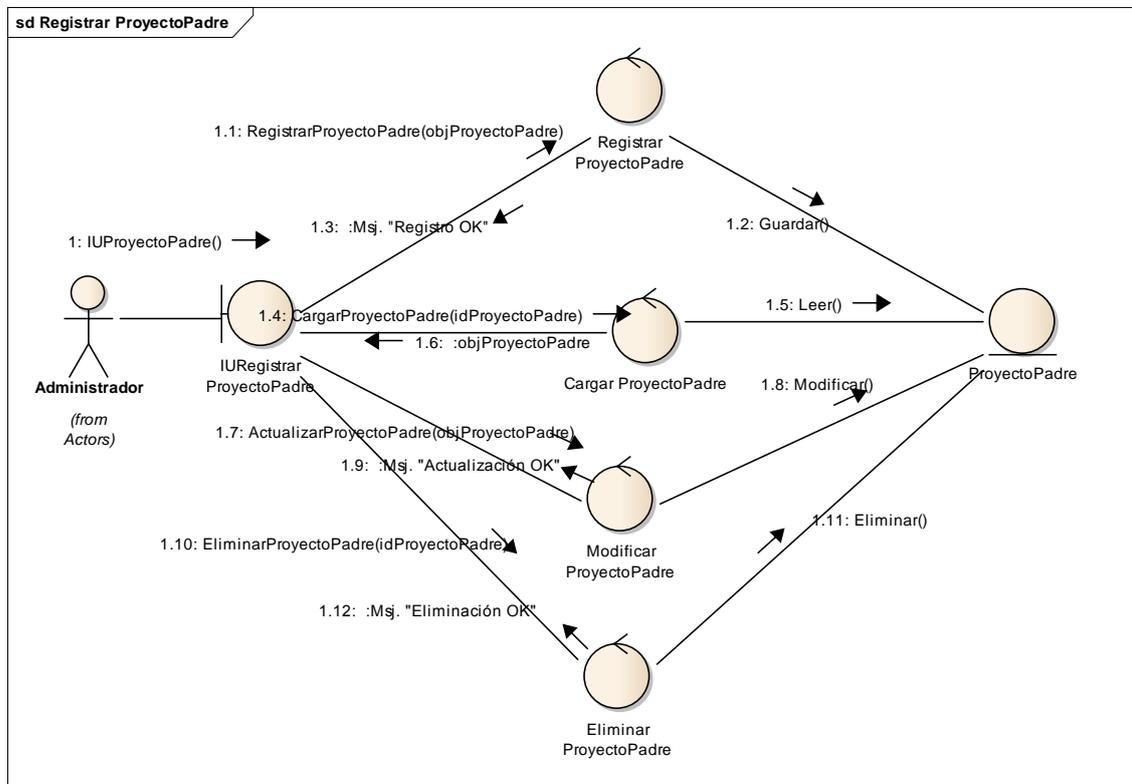


Fig. 42 Diagrama de Robustez de Mantenimiento de Proyecto Padre

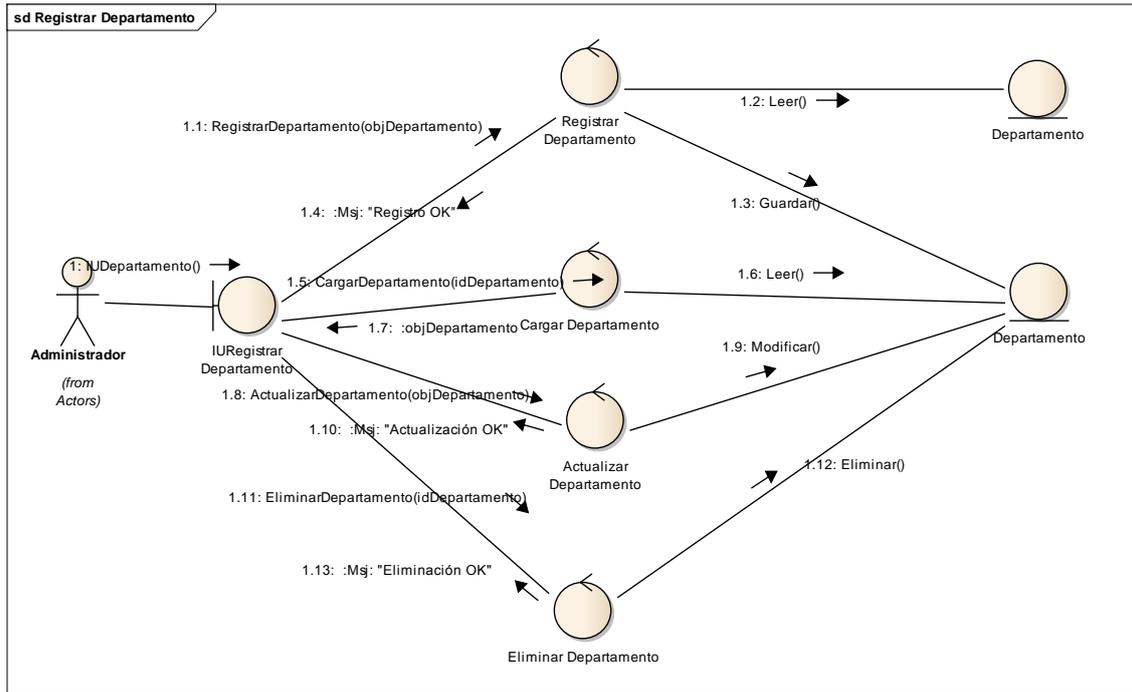


Fig. 43 Diagrama de Robustez de Mantenimiento de Departamento

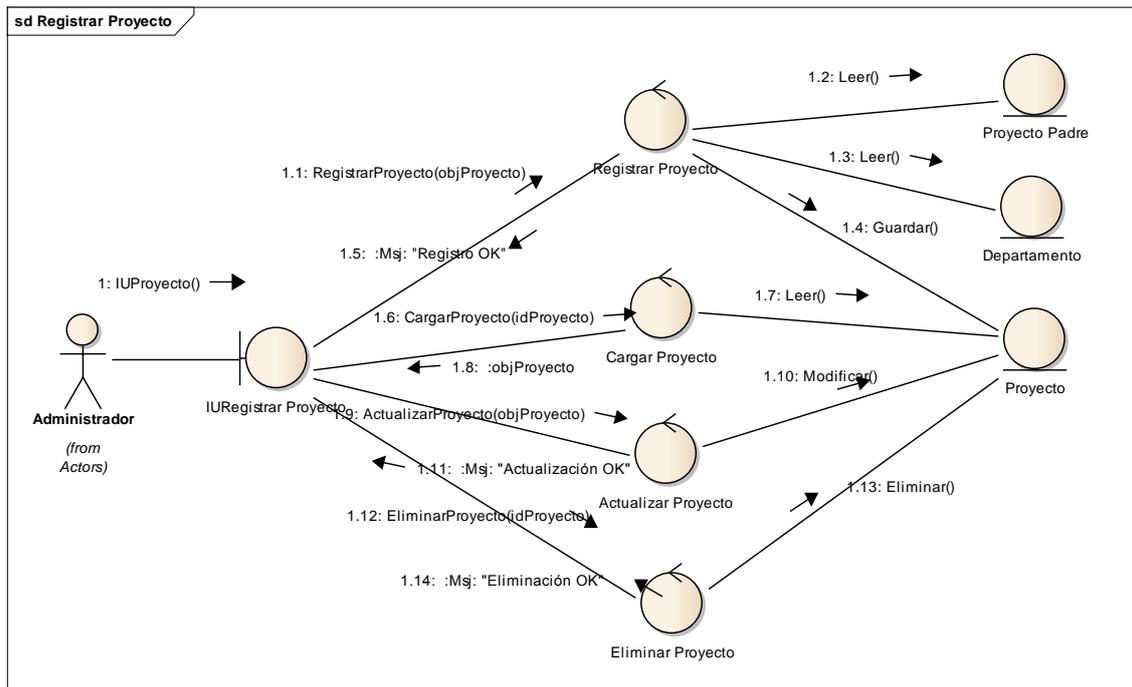


Fig. 44 Diagrama de Robustez de Mantenimiento de Proyecto

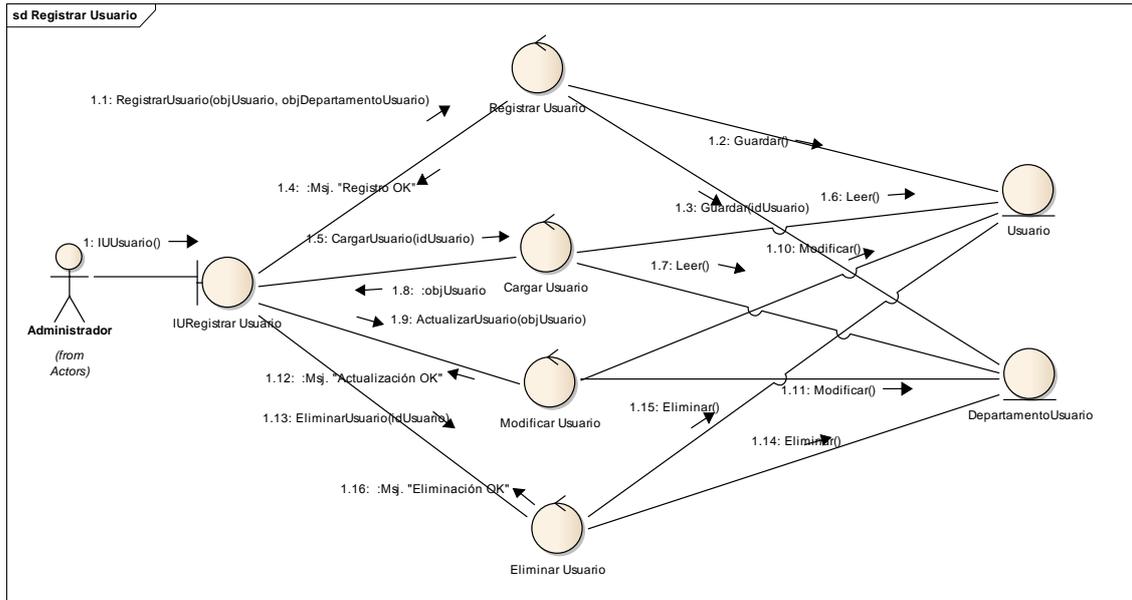


Fig. 45 Diagrama de Robustez de Mantenimiento de Usuario

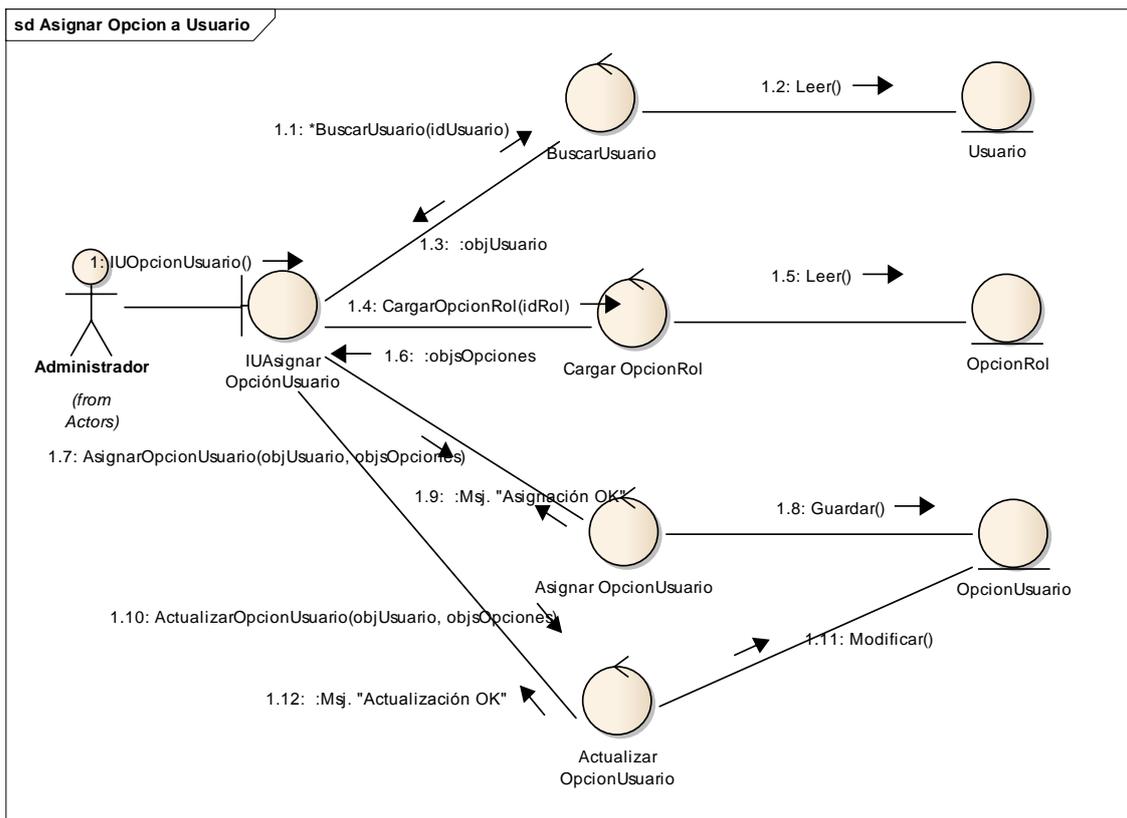


Fig. 46 Diagrama de Robustez de Asignar Opción a Usuario

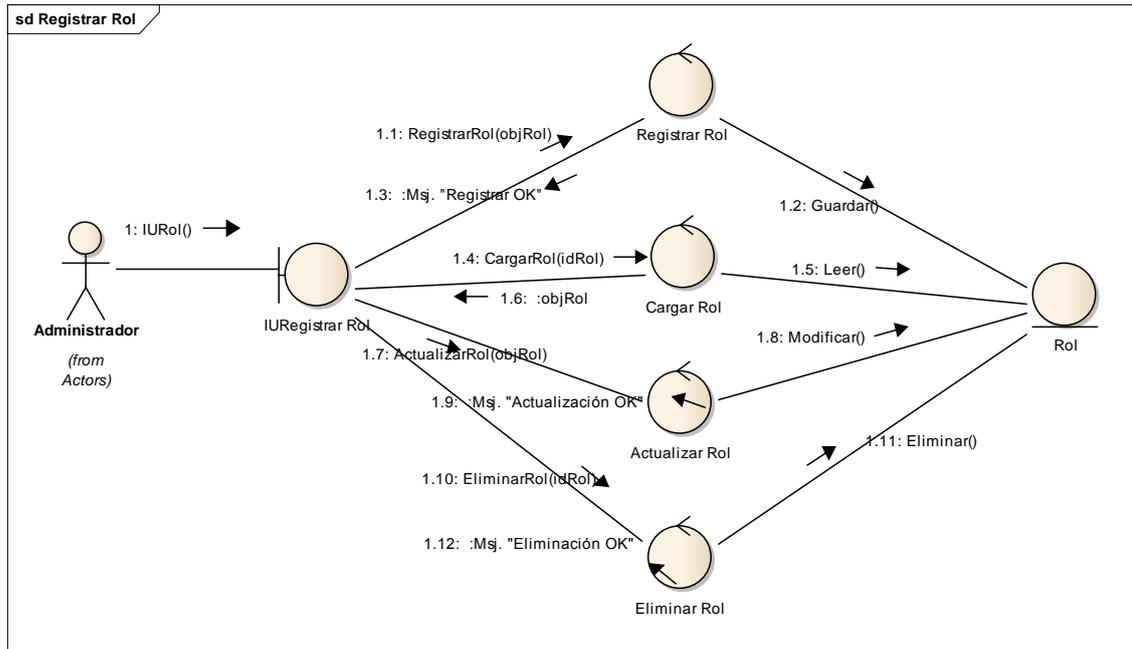


Fig. 47 Diagrama de Robustez de Mantenimiento Rol

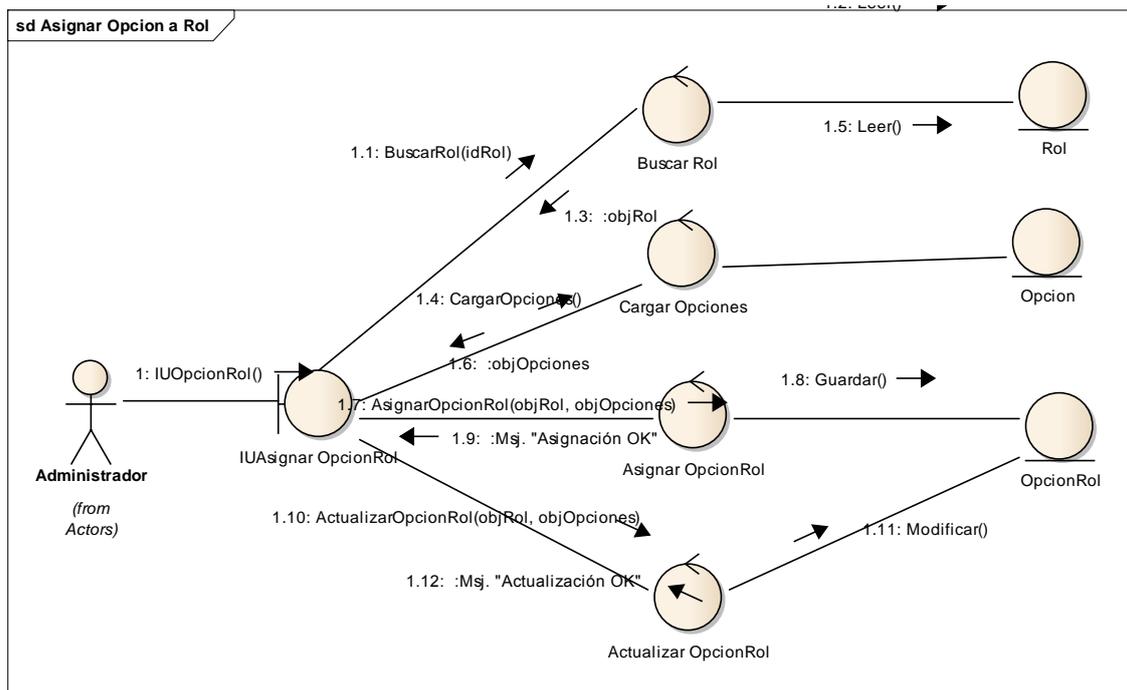


Fig. 48 Diagrama de Robustez de Asignar Opción a Rol

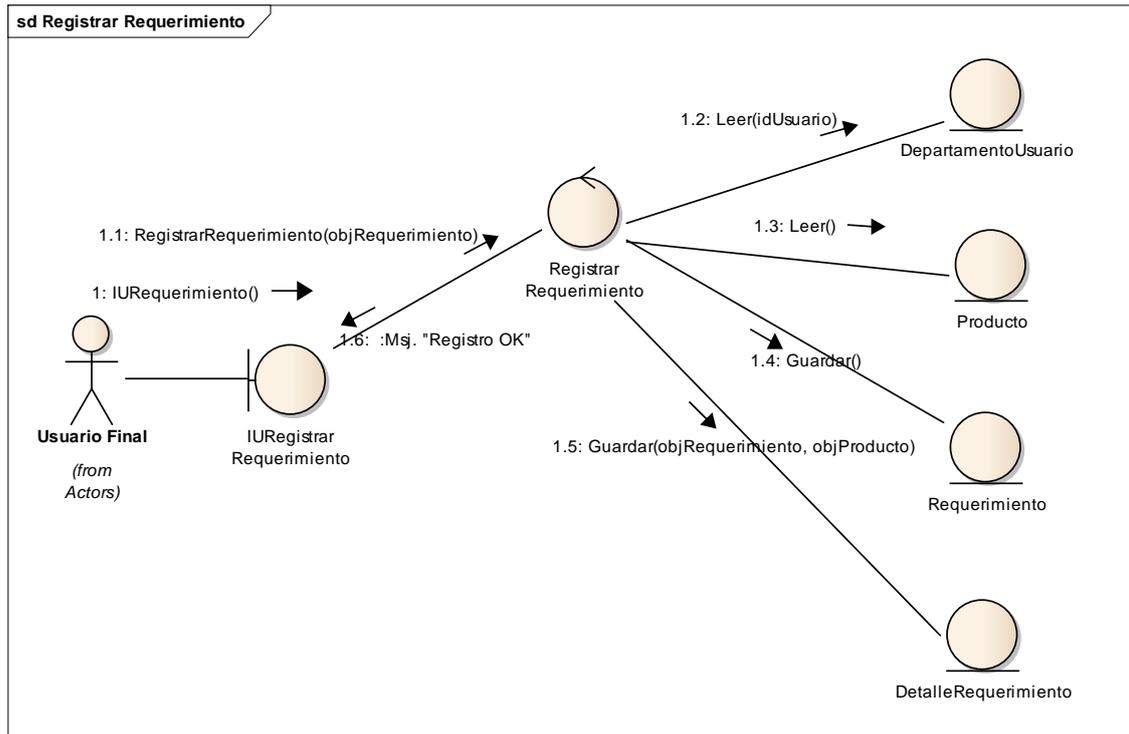


Fig. 49 Diagrama de Robustez de Registrar Requerimiento

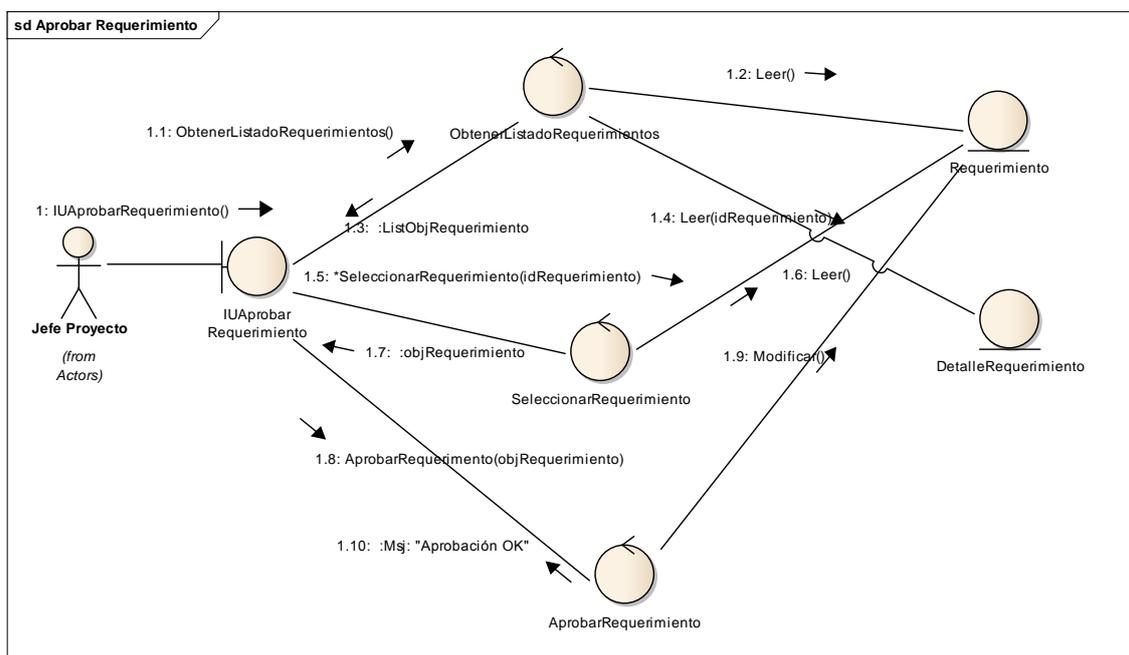


Fig. 50 Diagrama de Robustez de Aprobar Requerimiento

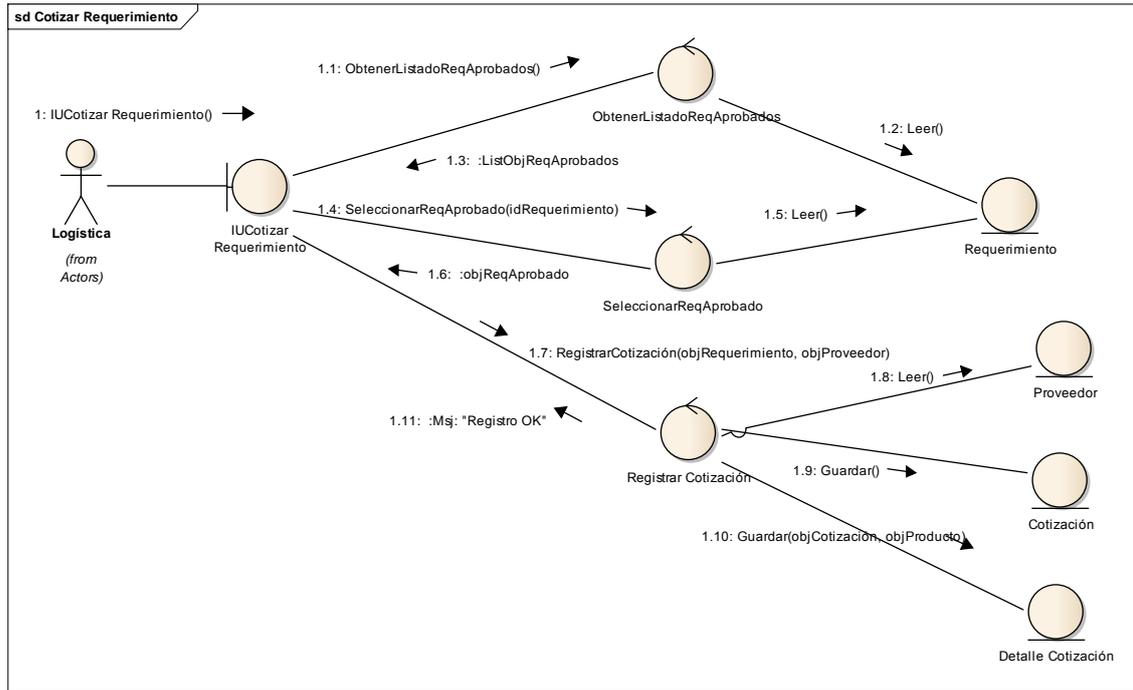


Fig. 51 Diagrama de Robustez de Cotizar Requerimiento

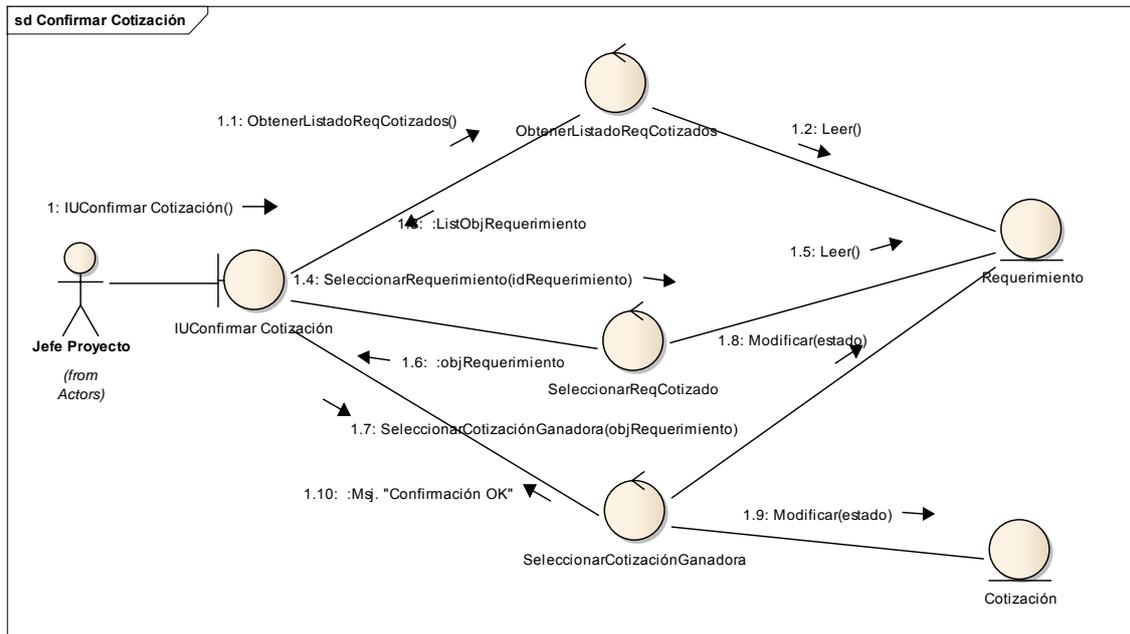


Fig. 52 Diagrama de Robustez de Confirmar Requerimiento Cotizado

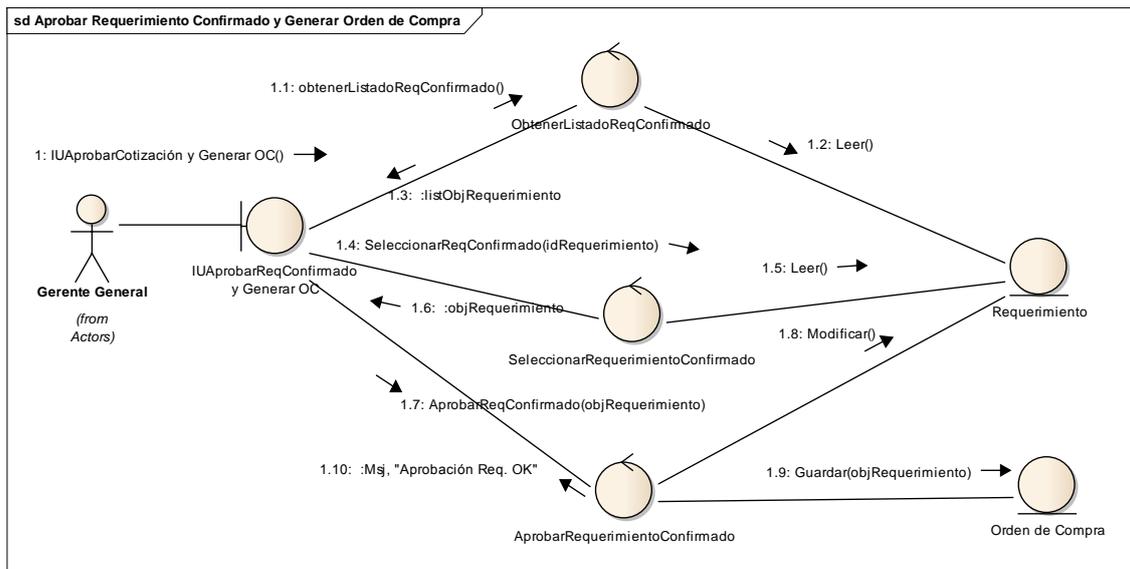


Fig. 53 Diagrama de Robustez de Aprobar Requerimiento Confirmado y Generar Orden de Compra

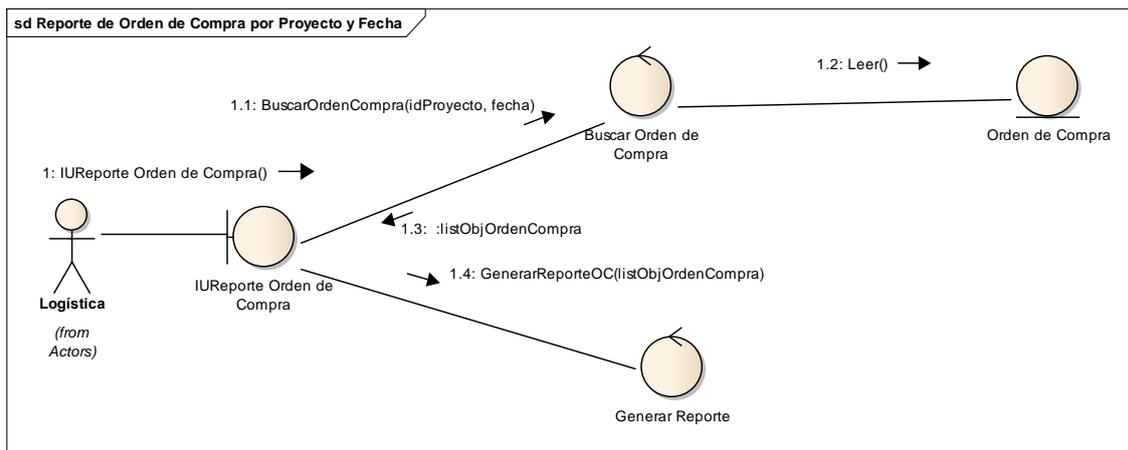


Fig. 54 Diagrama de Robustez de Reportes de Orden de Compra por Proyecto y Fecha

2.3.3. Especificación de los requerimientos

Caso de Uso:	Mantenimiento de Categoría de Productos
Definición u Objetivos	Tener una mayor clasificación de los productos y tener un mayor control de los productos
Pre- Condición:	Ninguna
Post-Condición:	Tener registrado y actualizado las categorías de productos
Flujo Básico:	<ol style="list-style-type: none"> 1. Autenticación del Administrador 2. Ingresar datos obligatorios (nombre) de la categoría de producto y seleccionar estado 3. Registrar categoría de productos 4. Para actualizar, seleccionar registro de la categoría de producto en el listado

5. Modificar datos de la categoría de producto
6. Guardar categoría de producto

Tabla 3Especificación de Mantenimiento de Categoría de Producto

Caso de Uso:	Mantenimiento de Sub-Categoría de Productos
Definición u Objetivos	Registrar y actualizar la sub-categoría según categoría
Pre- Condición:	Categoría registrada
Post-Condición:	Tener a las sub-categoría de productos registrados por categorías
	1. Autenticación delAdministrador
Flujo Básico:	2. Ingresar datos obligatorios de la sub-categoría de producto (nombre) y seleccionar categoría de producto
	3. Registrar sub-categoría de productos
	4. Para actualizar, seleccionar el registro de sub-categoría de producto en el listado
	5. Modificar datos de la sub-categoría de producto
	6. Guardar sub-categoría de producto

Tabla 4Especificación de Mantenimiento de Sub-Categoría de Producto

Caso de Uso:	Mantenimiento de Productos
Definición u Objetivos	Registrar y actualizar el producto según la sub-categoría
Pre- Condición:	Sub-Categoría registrada
Post-Condición:	Tener a los productos registrados por sub-categorías
	1. Autenticación del Administrador
Flujo Básico:	2. Ingresar datos obligatorios del producto (nombre, precio de coste) y seleccionar (sub-categoría, tipo de producto, estado)
	3. Registrar producto
	4. Para actualizar, seleccionar el registro de producto en el listado
	5. Modificar datos del producto
	6. Guardar producto

Tabla 5Especificación de Mantenimiento de Producto

Caso de Uso:	Mantenimiento de Rubros
Definición u Objetivos	Registrar y actualizar el rubro
Pre- Condición:	Ninguna
Post-Condición:	Tener a los rubros registrados y actualizados
	1. Autenticación del Administrador
Flujo Básico:	2. Ingresar datos obligatorios delrubro (nombre) y seleccionar estado
	3. Registrar rubro
	4. Para actualizar, seleccionar el registro de rubro en el

listado
5. Modificar datos del rubro
6. Guardar rubro

Tabla 6 Especificación de Mantenimiento del Rubro

Caso de Uso:	Mantenimiento de Proveedores
Definición u Objetivos	Registrar y actualizar el proveedor
Pre- Condición:	Rubro registrado
Post-Condición:	Tener a los proveedores registrados y actualizados por rubro
Flujo Básico:	<ol style="list-style-type: none"> 1. Autenticación del Administrador 2. Ingresar datos obligatorios del proveedor (nombre, número documento, dirección, correo, teléfono, ciudad) y seleccionar (tipo documento, es empresa, estado) 3. Registrar proveedor 4. Para actualizar, seleccionar el registro de proveedor en el listado 5. Modificar datos del proveedor 6. Guardar proveedor

Tabla 7 Especificación de Mantenimiento del Proveedor

Caso de Uso:	Mantenimiento de Proyecto Padre
Definición u Objetivos	Registrar y actualizar el proyecto padre
Pre- Condición:	Ninguna
Post-Condición:	Tener a los proyecto padres registrados y actualizados
Flujo Básico:	<ol style="list-style-type: none"> 1. Autenticación del Administrador 2. Ingresar datos obligatorios del proyecto padre (nombre) y seleccionar (estado) 3. Registrar proyecto padre 4. Para actualizar, seleccionar el registro del proyecto padre en el listado 5. Modificar datos del proyecto padre 6. Guardar proyecto padre

Tabla 8 Especificación de Mantenimiento del Proyecto Padre

Caso de Uso:	Mantenimiento de Departamentos
Definición u Objetivos	Registrar y actualizar los departamentos
Pre- Condición:	Ninguna
Post-Condición:	Tener a los departamentos registrados y actualizados
Flujo Básico:	<ol style="list-style-type: none"> 1. Autenticación del Administrador 2. Ingresar datos obligatorios del departamento (nombre) y seleccionar (departamento superior, estado)

3. Registrar departamento
4. Para actualizar, seleccionar el registro del departamento en el listado
5. Modificar datos del departamento
6. Guardar departamento

Tabla 9 Especificación de Mantenimiento del Departamento

Caso de Uso:	Mantenimiento de Proyectos
Definición u Objetivos	Registrar y actualizar los proyectos
Pre- Condición:	Proyectos Padres y Departamentos registrados
Post-Condición:	Tener a los proyectos registrados y actualizados
Flujo Básico:	1. Autenticación del Administrador
	2. Ingresar datos obligatorios del proyecto (nombre) y seleccionar (proyecto padre, departamento, estado)
	3. Registrar proyecto
	4. Para actualizar, seleccionar el registro del departamento en el listado
	5. Modificar datos del departamento
	6. Guardar departamento

Tabla 10 Especificación de Mantenimiento del Proyecto

Caso de Uso:	Mantenimiento de Usuarios
Definición u Objetivos	Registrar y actualizar los usuarios
Pre- Condición:	Ninguna
Post-Condición:	Tener a los usuarios registrados y actualizados
Flujo Básico:	1. Autenticación del Administrador
	2. Ingresar datos obligatorios del proyecto (nombre, apellido paterno, apellido materno, teléfono, email, fecha ingreso, fecha cese, número documento) y seleccionar (rol, tipo documento, estado, esJefe, departamento, proyecto)
	3. Registrar usuario
	4. Para actualizar, seleccionar el registro del usuario en el listado
	5. Modificar datos del usuario
	6. Guardar usuario

Tabla 11 Especificación de Mantenimiento del Usuario

Caso de Uso:	Asignación de Opción a Usuario
Definición u Objetivos	Asignar opciones a usuario
Pre- Condición:	Usuarios y Opciones registrados
Post-Condición:	Tener a los opciones por usuario registrados y actualizados
	1. Autenticación del Administrador

Flujo Básico:	2. Para actualizar, seleccionar el registro del usuario en el listado y clic en el botón asignar opciones.
	3. Modificar opciones del usuario
	4. Guardar opciones del usuario.

Tabla 12 Especificación de Asignación de Opción a Usuario

Caso de Uso:	Mantenimiento de roles
Definición u Objetivos	Registrar y actualizar los roles
Pre- Condición:	Ninguna
Post-Condición:	Tener a los roles registrados y actualizados
Flujo Básico:	1. Autenticación del Administrador
	2. Ingresar datos obligatorios del proyecto (nombre, descripción) y seleccionar (estado)
	3. Registrar rol
	4. Para actualizar, seleccionar el registro del rol en el listado
	5. Modificar datos del rol
	6. Guardar rol

Tabla 13 Especificación de Mantenimiento del Rol

Caso de Uso:	Asignación de Opción a Rol
Definición u Objetivos	Asignar opciones a rol
Pre- Condición:	Roles y Opciones registrados
Post-Condición:	Tener a las opciones por roles registrados y actualizados
Flujo Básico:	1. Autenticación del Administrador
	2. Para actualizar, seleccionar el registro del rol en el listado y clic en el botón asignar opciones.
	3. Modificar opciones del rol
	4. Guardar opciones del rol

Tabla 14 Especificación de Asignación de Opción a Rol

Caso de Uso:	Registro de Requerimiento y Detalle de Requerimiento
Definición u Objetivos	Registrar requerimientos
Pre- Condición:	Productos, usuarios, proyectos registrados.
Post-Condición:	Tener a los requerimientos con su detalle registrados
Flujo Básico:	1. Autenticación del Usuario Final
	2. Seleccionar en el listado de productos, el producto e ingresar su cantidad
	3. Guardar requerimiento

Tabla 15 Especificación de Registro de Requerimiento y Detalle de Requerimiento

Caso de Uso:	Aprobar Requerimiento
Definición u Objetivos	Aprobar requerimiento por parte del Jefe de Proyecto
Pre- Condición:	Requerimientos registrados
Post-Condición:	Aprobar requerimiento por parte del jefe de proyecto
Flujo Básico:	1. Autenticación del Jefe de Proyecto
	2. Visualizar los requerimientos en estado “nuevo”
	3. Seleccionar el requerimiento a aprobar
	4. Revisar y Aprobar el requerimiento seleccionado.

Tabla 16 Especificación de Aprobación de Requerimientos

Caso de Uso:	Cotizar Requerimiento
Definición u Objetivos	Cotizar requerimiento por parte del usuario de logística
Pre- Condición:	Requerimientos aprobados, proveedores y productos registrados
Post-Condición:	Cotizar requerimiento por parte del usuario de logística
Flujo Básico:	1. Autenticación del usuario de logística
	2. Visualizar los requerimientos en estado “aprobado”
	3. Seleccionar el requerimiento a cotizar
	4. Clic en nueva cotización y registrar cotización con su detalle
	5. Al tener todas las cotizaciones listas, clic en cotizado.

Tabla 17 Especificación de Cotización de Requerimientos

Caso de Uso:	Confirmar Requerimiento Cotizado
Definición u Objetivos	Confirmar el requerimiento cotizado ganador
Pre- Condición:	Requerimientos cotizados
Post-Condición:	Confirmar el requerimiento cotizado ganador
Flujo Básico:	1. Autenticación del Jefe de Proyecto
	2. Visualizar los requerimientos en estado “cotizado”
	3. Seleccionar el requerimiento cotizado a confirmar
	4. Seleccionar la cotización de requerimiento ganadora
	5. Confirmar requerimiento con cotización ganadora.

Tabla 18 Especificación de Confirmación Requerimiento Cotizado

Caso de Uso:	Aprobar Requerimiento Cotizado
Definición u Objetivos	Aprobar el requerimiento confirmado
Pre- Condición:	Requerimientos confirmados
Post-Condición:	Aprobar el requerimiento confirmado
Flujo Básico:	1. Autenticación del Gerente General
	2. Seleccionar el requerimiento cotizado a aprobar
	3. Revisar el requerimiento confirmado y aprobar

4. Se genera la orden de compra automáticamente

Tabla 19 Especificación de Aprobación Requerimiento Confirmado

Caso de Uso:	Reporte de Órdenes de Compra por Proyecto y Fecha
Definición u Objetivos	Generar una lista de compras por proyecto y fecha para hacer un seguimiento de ellas y asegurarse que van de acuerdo a los objetivos del negocio
Pre- Condición:	Estar registrado como usuario de la empresa
Post-Condición:	Modificar los datos antiguos y actualizarlos para un mejor manejo de los mismos.
Flujo Básico:	<ol style="list-style-type: none"> 1. Autenticación del usuario de Logística 2. Buscar compras (proyecto, fecha) 3. Generar reportes de órdenes de compra por proyecto y fecha

Tabla 20 Especificación de Requerimiento Reporte de Órdenes de Compra por proyecto y fecha

2.4. Diseño Detallado

2.4.1. Diseño de Interfaces

- Seguridad
 - Registro de usuarios
 - Carga masiva de usuarios
 - Administración de perfiles
- Mantenimiento
- Compras

Registrar - Usuario

Nombre: (*)

Apellido Paterno: (*)

Apellido Materno: (*)

Perfil: (*) --SELECCIONAR--

Tipo Documento: (*) --SELECCIONAR--

Correo:

Departamento: (*) --SELECCIONAR--

Es Jefe?: (*) --SELECCIONAR--

Usuario: (*)

Teléfono:

Proyecto: (*) --SELECCIONAR--

Fecha ingreso: (*)

Fecha nacimiento: (*)

(*) Campo obligatorio

<< < > >>

Listado de usuarios							
#	Código	Apellido Paterno	Apellido Materno	Nombres	Estado	Acciones del usuario	Opciones de menú
1	12435687	Barzola	Barzola	Leonor	H		
2	87654321	Zubiate	Alban	Maryn	H		
3	12345678	Falchetta	Falchetta	Giuseope	H		

Fig. 55 Página Registro de Usuarios



Fig. 56 Página Asignar Opción a Usuario



Fig. 57 Página Registro de Roles



Fig. 58 Página Asignar Opción a Rol



Fig. 59 Página de Mantenimiento de Categoría de Productos

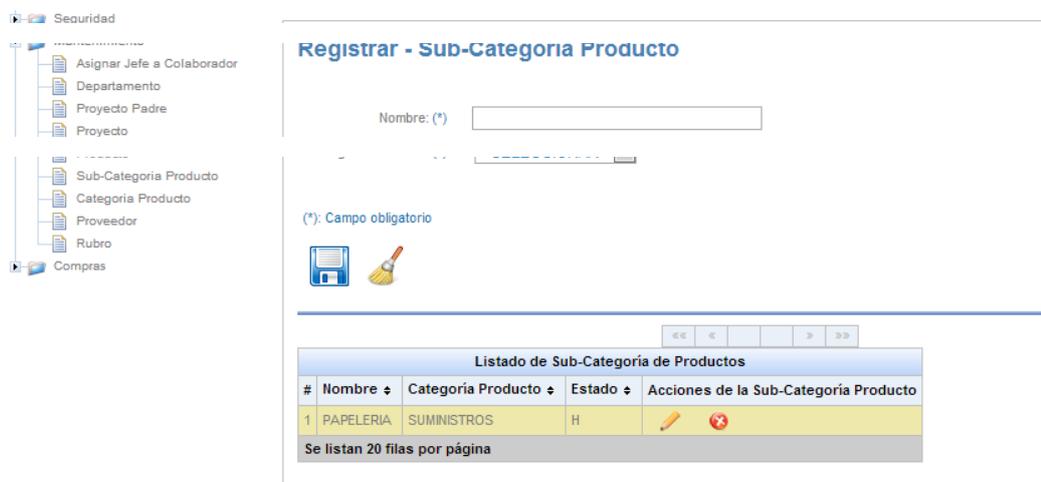
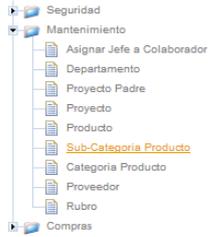


Fig. 60 Página Mantenimiento de Sub-Categoría de Productos



Registrar - Producto

Nombre: (*)

Sub-Categoría Producto: (*) --SELECCIONAR--

Tipo Producto: (*) --SELECCIONAR--

Precio Coste: (*)

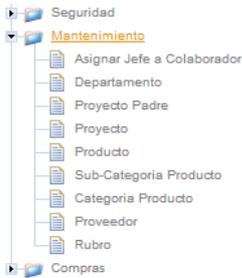
(*): Campo obligatorio

Listado de Productos						
#	Nombre	Sub-Categoría Producto	Tipo de Producto	Precio Coste	Estado	Acciones del Producto
1	PAPELOTE	PAPELERIA	Consumible	20.0	H	 
2	PAPEL BOND A4 75GR CHAMEX	PAPELERIA	Consumible	16.0	H	 

Se listan 20 filas por página

Fig. 61Página Mantenimiento de Productos



Registrar - Proyecto Padre

Nombre: (*)

(*): Campo obligatorio

Listado de Proyecto Padres			
#	Nombre	Estado	Acciones de Proyecto Padre
1	Gastos de Administracion	H	 

Se listan 20 filas por página

Fig. 62Página Mantenimiento de Proyecto Padre

Registrar - Departamento

Nombre: (*)

Departamento Superior:

(*): Campo obligatorio

Listado de Departamentos				
#	Nombre ↕	Estado ↕	Departamento Superior ↕	Acciones del Departamento
1	Logística	H	Gerencia General	 
2	Tesorería	H	Contabilidad	 
3	Contabilidad	H	Gerencia General	 
4	Recursos Humanos	H		 
5	Gerencia Sistemas	H	Gerencia General	 
6	Gerencia General	H		 

Se listan 20 filas por página

Fig. 63Página Mantenimiento de Departamentos

Registrar - Proyecto

Nombre: (*)

Proyecto Padre: (*)

Departamento: (*)

(*): Campo obligatorio

Listado de Proyectos					
#	Nombre ↕	Proyecto Padre ↕	Departamento ↕	Estado ↕	Acciones del Proyecto
1	BBVA	Gastos de Administracion	Gerencia Sistemas	H	 
2	Sistemas	Gastos de Administracion	Gerencia Sistemas	H	 

Se listan 20 filas por página

Fig. 64Página Mantenimiento de Proyectos

Se modificó el rubro 'UTILES ESCOLARES' satisfactoriamente.

Registrar - Rubro

Nombre: (*)

(*) Campo obligatorio

<< < > >>

Listado de Rubro			
#	Nombre ↕	Estado ↕	Acciones del Rubro
1	UTILES ESCOLARES	H	 

Se listan 20 filas por página

Fig. 65Página Mantenimiento de Rubros

Registrar - Proveedor

Es una empresa: (*)

Nombre: (*)

Tipo Documento: (*) --SELECCIONAR--

Numero Documento: (*)

Rubro: (*) --SELECCIONAR--

Direccion: (*)

Correo: (*)

Telefono: (*)

Ciudad: (*)

(*) Campo obligatorio

<< < > >>

Listado de Proveedor			
#	Nombre ↕	Estado ↕	Acciones del Proveedor
1	Reficorp	H	 
2	Acovalco	H	 
3	Tayloy	H	 

Se listan 20 filas por página

Fig. 66Página Mantenimiento de Proveedores



Requerimiento

Numero:

Responsable: Fecha:

Proyecto: Jefe:

«« « »»»

Listado de Productos			
#	Nombre ↕	Precio ↕	Seleccionar
1	Laptop Dell i3 3GB RAM	1800.0	<input type="checkbox"/>
2	Laptop Toshiba i7 6 GB RAM	4000.0	<input type="checkbox"/>
3	PAPELOTE	20.0	<input type="checkbox"/>
4	PAPEL BOND A4 75GR CHAMEX	16.0	<input type="checkbox"/>

Se listan 10 filas por página

Cantidad:

Fig. 67 Página Registrar Requerimiento



Consultar Requerimientos

Estado:

«« « »»»

Listado de Requerimientos							
#	Número Requerimiento ↕	Responsable ↕	Fecha ↕	Jefe Proyecto ↕	Proyecto ↕	Estado ↕	Cargar Requerimiento
1	3	Dennis	22/09/2013	Giuseppe	BBVA	Nuevo	<input type="button" value="⊕"/>

Se listan 10 filas por página

Fig. 68 Página Consultar y Aprobar Requerimiento



Requerimiento por Aprobar

Numero:

Responsable: Fecha:

Proyecto: Jefe:

«« « »»»

Listado de Detalle Requerimientos			
#	Nombre ↕	Precio ↕	Cantidad ↕
1	Laptop Toshiba i7 6 GB RAM	4000.0	2

Se listan 10 filas por página

Fig. 69 Página Aprobar Requerimiento



Consultar Req. Aprobados

«« « »»»

Listado de Requerimientos							
#	Número Requerimiento ↕	Responsable ↕	Fecha ↕	Jefe Proyecto ↕	Proyecto ↕	Estado ↕	Cargar Requerimiento
1	1	Dennis	2013-07-20 00:00:00.0	Giuseppe	BBVA	Aprobado	<input type="button" value="⊕"/>

Se listan 10 filas por página

Fig. 70 Página Consultar Requerimientos Aprobados

Compras
Consultar Req. Aprobados

Requerimiento por Aprobar

Numero:
 Responsable: Fecha:
 Proyecto: Jefe:

Listado de Detalle Requerimientos

#	Nombre	Precio	Cantidad
1	Laptop Toshiba i7 6 GB RAM	4000.0	2

Se listan 10 filas por página

Consultar

Compras
Consultar Req. Aprobados

Nueva Cotización

Proveedor: Fecha:

Listado de Detalle Requerimientos

#	Nombre	Precio	Cantidad	IGV	Subtotal
1	Laptop Toshiba i7 6 GB RAM	4000.0	2	19.0 %	8000.0

Se listan 10 filas por página

Subtotal :8000.0
 IGV :1520.0
 Total :9520.0

Consultar Consultar Aprobados

Fig. 71Página Cotizar Requerimiento

Compras
Consultar Requerimientos

Consultar Req. Cotizados

Estado:

Listado de Requerimientos

#	Número Requerimiento	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Cargar Requerimiento
1	3	Dennis	22/09/2013	Giuseppe	BBVA	Cotizado	

Se listan 10 filas por página

Fig. 72Página Consultar Requerimientos Cotizados

Compras
Consultar Requerimientos

Requerimiento por Confirmar

Confirmar Requerimiento

Datos de Requerimiento

Numero:

Responsable: Fecha:

Proyecto: Jefe:

Listado de Cotizaciones				
#	Proveedor	Total	Seleccionar	Ver Cotizacion
1	Refricorp	9044.0	<input type="checkbox"/>	
2	Acovalco	9520.0	<input type="checkbox"/>	

Se listan 10 filas por página

[Cancelar](#)

Compras
Consultar Requerimientos

Requerimiento por Confirmar

Confirmar Requerimiento

Datos de Requerimiento

Numero:

Responsable: Fecha:

Proyecto: Jefe:

Listado de Cotizaciones				
#	Proveedor	Total	Seleccionar	Ver
1	Refricorp	9044.0	<input type="checkbox"/>	
2	Acovalco	9520.0	<input type="checkbox"/>	

Se listan 10 filas por página

[Cancelar](#)

Detalle Cotizacion

Proveedor: Refricorp Fecha: 22/09/2013

Listado de Detalle Cotizacion					
#	Nombre	Precio	Cantidad	IGV	Subtotal
1	Laptop Toshiba i7 6 GB RAM	3800.0	2	19.0 %	7600.0

Se listan 10 filas por página

Subtotal :7600.0
 IGV :1444.0
 Total :9044.0

[Cancelar](#)

Fig. 73Página Confirmar Requerimiento

Compras
Consultar Req. Confirmados

Consultar Req. Confirmados

Listado de Cotizaciones								
#	Num. Requerimiento	Proveedor	Responsable	Fecha	Jefe Proyecto	Proyecto	Estado	Ver
1	3	Refricorp	Dennis	22/09/2013	Giuseppe	BBVA	Confirmado	
2	3	Acovalco	Dennis	22/09/2013	Giuseppe	BBVA	Cancelado	
3	2	Refricorp	Dennis	21/09/2013	Giuseppe	BBVA	Cancelado	
4	2	Acovalco	Dennis	21/09/2013	Giuseppe	BBVA	Cancelado	

Se listan 10 filas por página

Fig. 74Página Consultar Requerimientos Confirmados



Ver Cotizacion Ganadora

Proveedor: REFRICORP Fecha: 22/09/2013

Listado de Detalle Cotizacion					
#	Nombre	Precio	Cantidad	IGV	Subtotal
1	Laptop Toshiba i7 6 GB RAM	3800.0	2	19.0	7600.0

Se listan 10 filas por página

Subtotal :7600.0
IGV :1444.0
Total :9044.0

[Aprobar](#) [Cancelar](#)

Fig. 75Página Aprobar Cotización y Generar Orden de Compra



Orden de Compra

Listado de Ordenes de Compra								
#	Num. Orden	Num. Requerimiento	Proveedor	Fecha	Sub total	IGV	Total	Ver
1	1	2	Taylor	21/09/2013	10500.0	1995.0	12495.0	
2	2	3	Refricorp	22/09/2013	7600.0	1444.0	9044.0	

Se listan 10 filas por página

Fig. 76 Consultar Órdenes de Compra



Se creo la orden de compra N° '2' satisfactoriamente.

Orden de Compra

Orden de Compra: 2
Proveedor: REFRICORP Fecha: 22/09/2013

Listado de Detalle Cotizacion					
#	Nombre	Precio	Cantidad	IGV	Subtotal
1	Laptop Toshiba i7 6 GB RAM	3800.0	2	19.0	7600.0

Se listan 10 filas por página

Subtotal :7600.0
IGV :1444.0
Total :9044.0

[Cancelar](#)

Fig. 77Página Orden de Compra

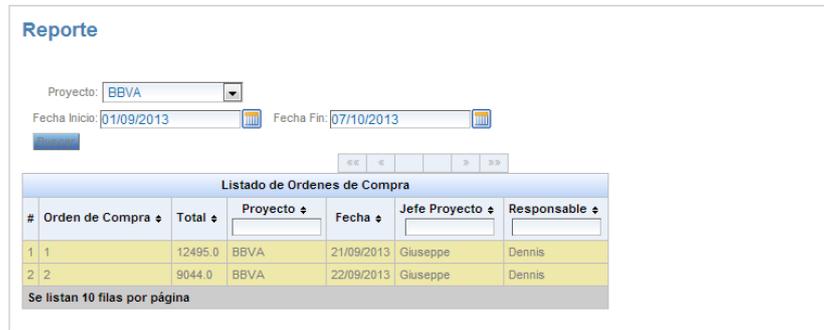


Fig. 78Página Reporte de Órdenes de Compra por Proyecto y Fecha

2.4.2. Diagramas de Secuencia

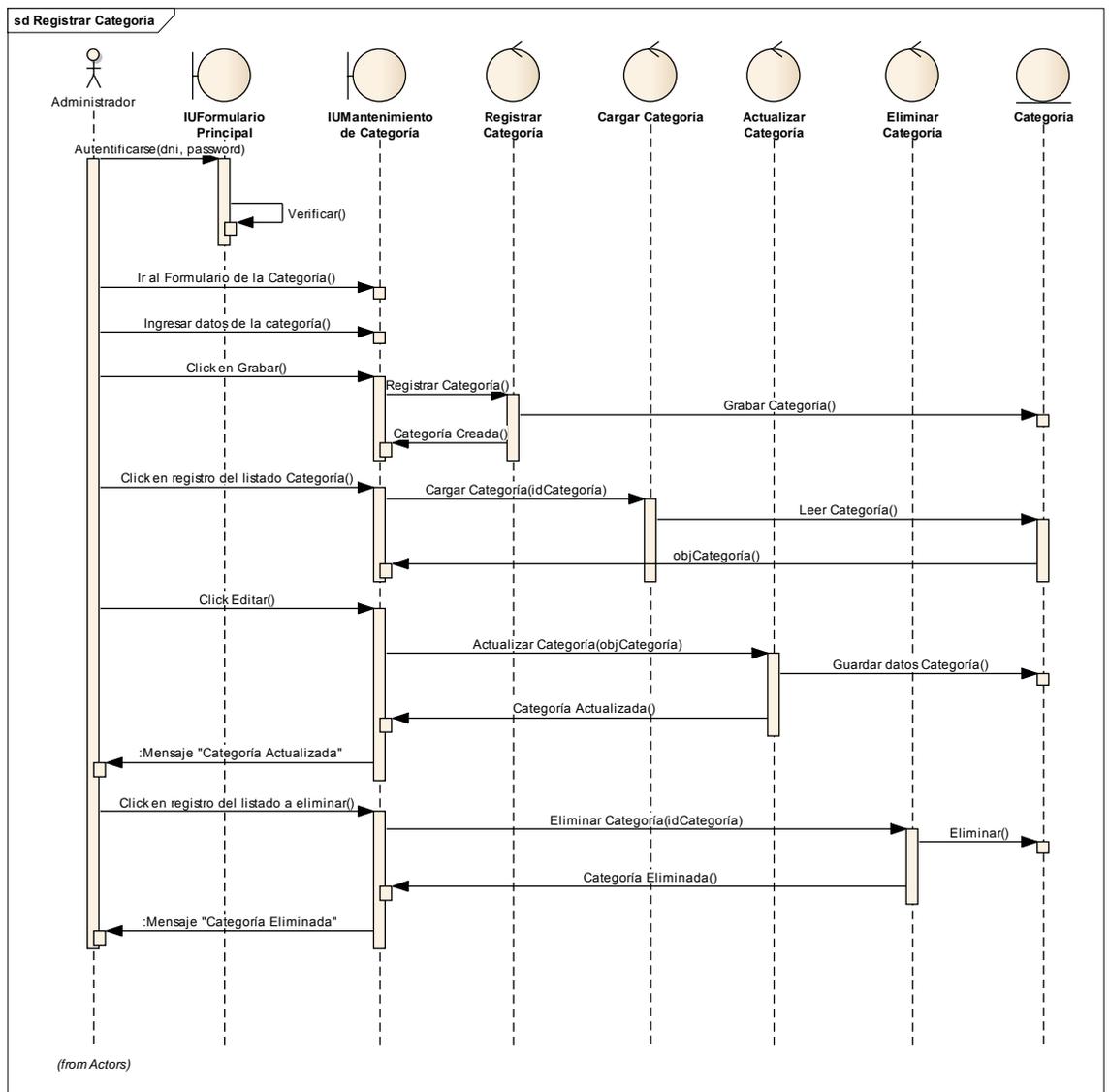


Fig. 79Diagrama de Secuencia de Mantenimiento Categoría

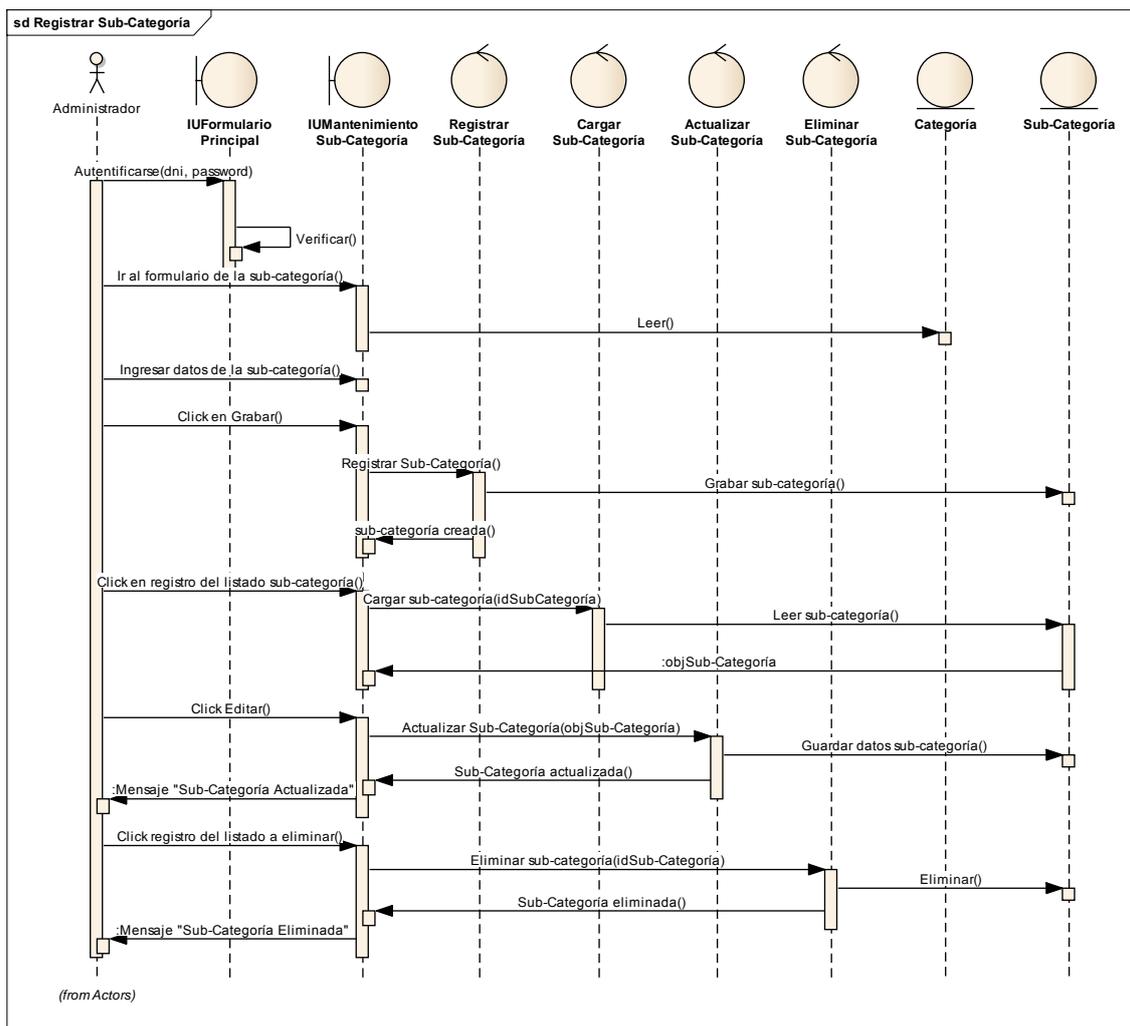


Fig. 80 Diagrama de Secuencia de Mantenimiento Sub-Categoría

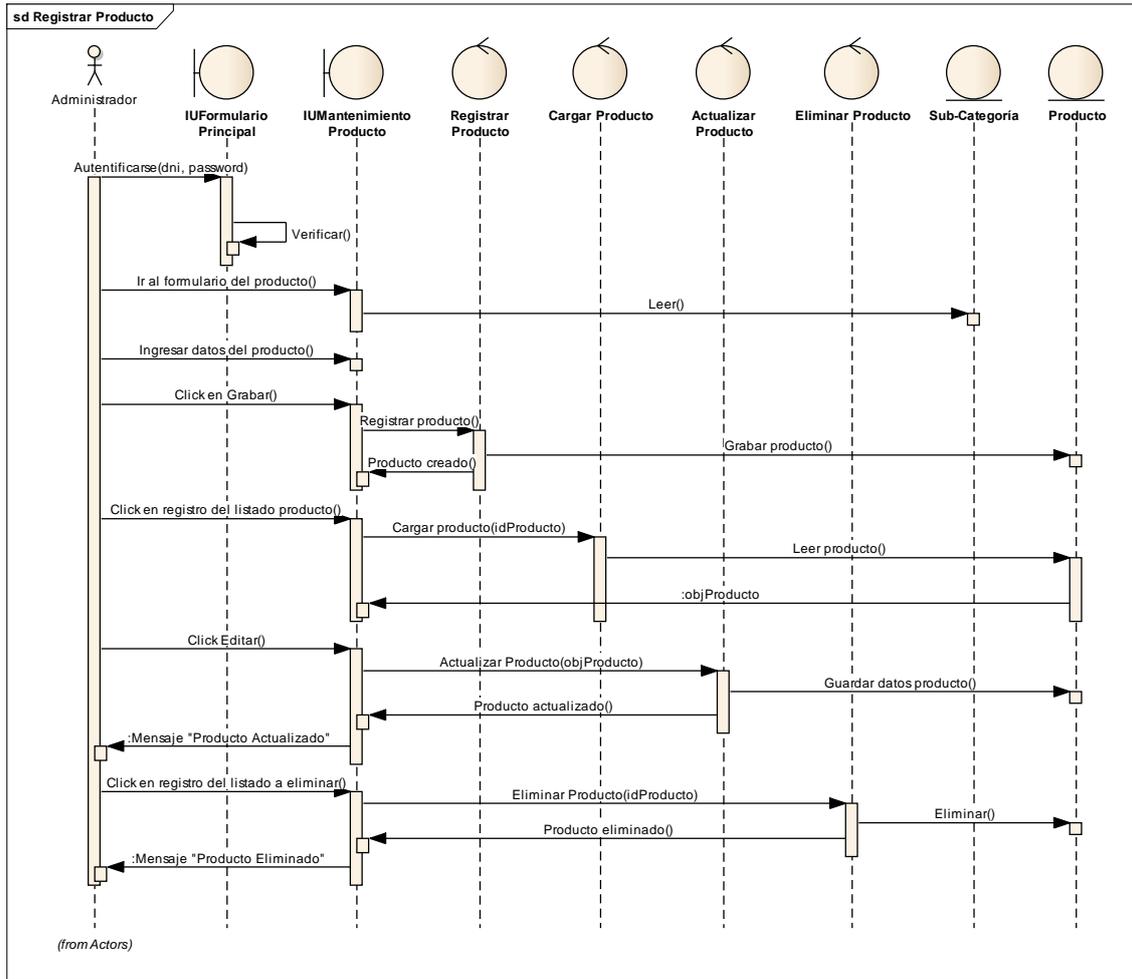


Fig. 81 Diagrama de Secuencia de Mantenimiento Producto

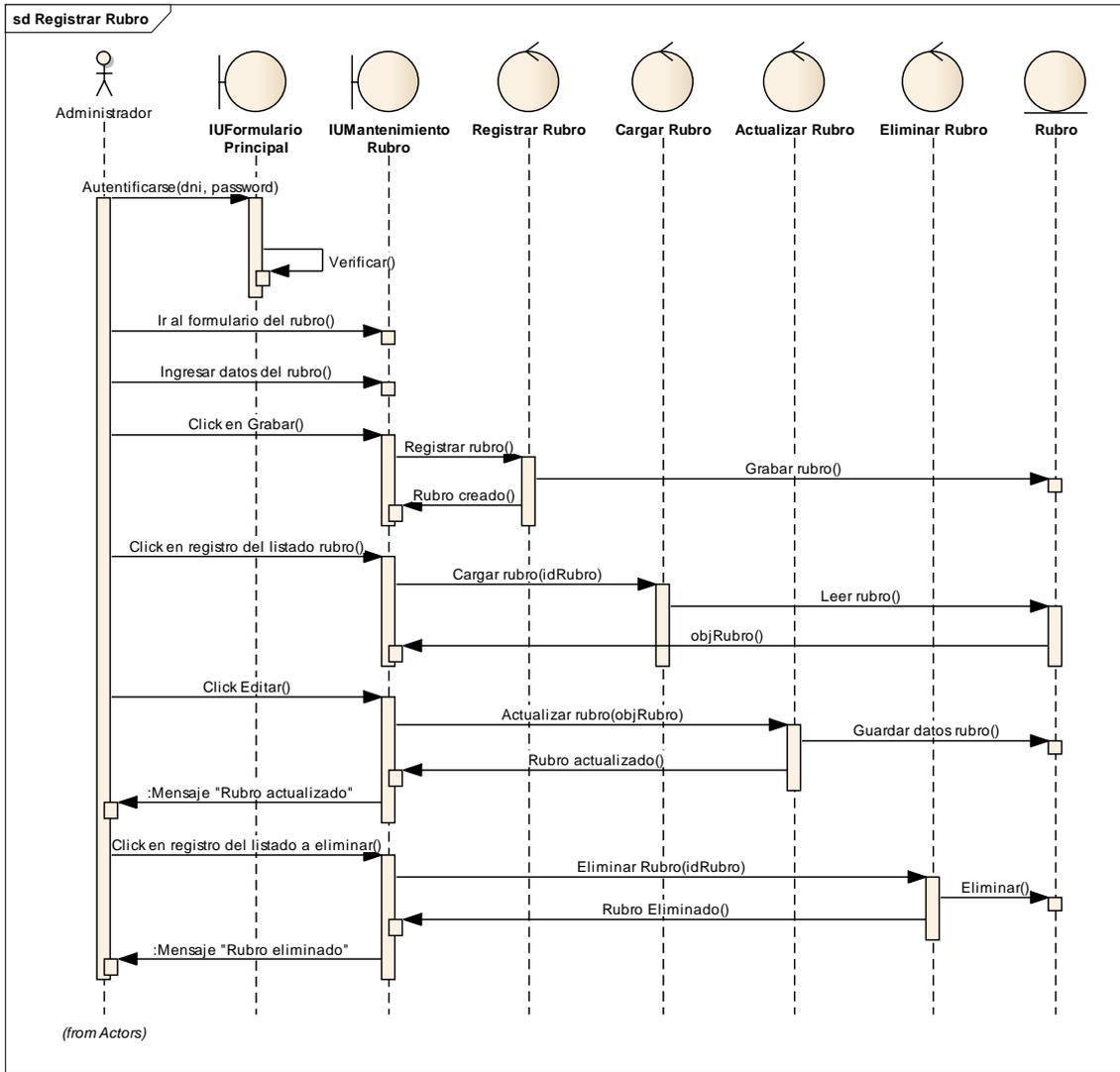


Fig. 82 Diagrama de Secuencia de Mantenimiento Rubro

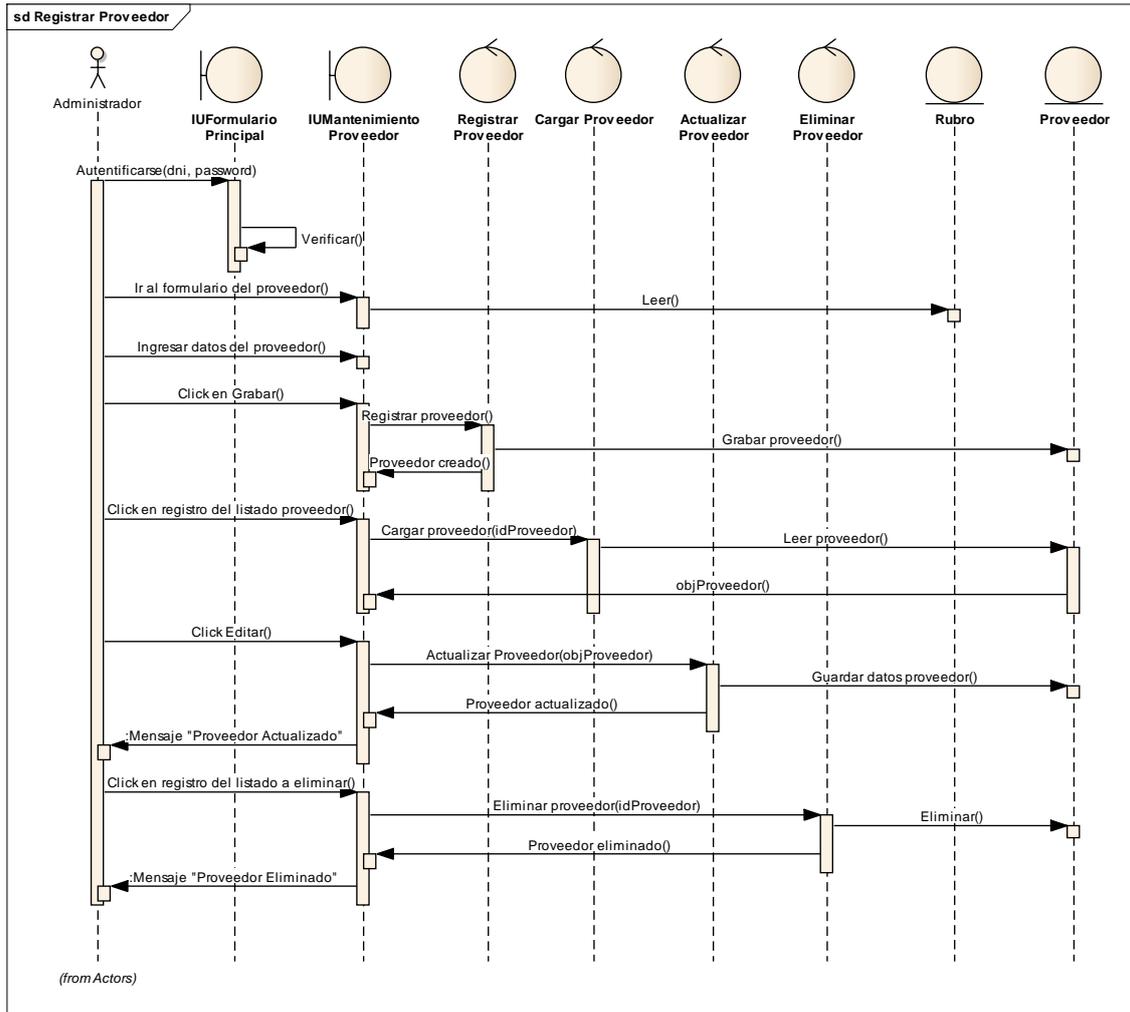


Fig. 83 Diagrama de Secuencia de Mantenimiento Proveedor

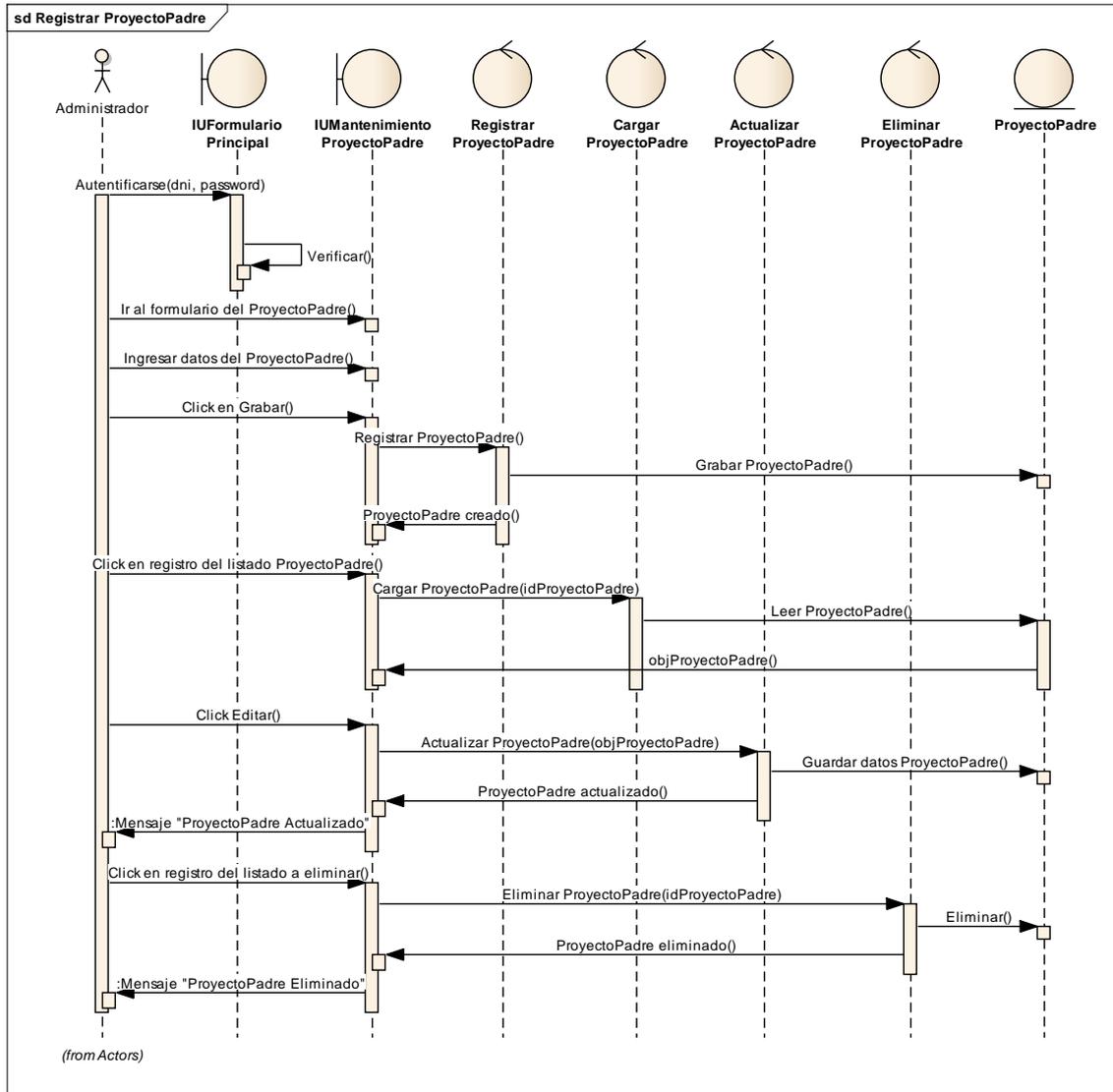


Fig. 84 Diagrama de Secuencia de Mantenimiento Proyecto Padre

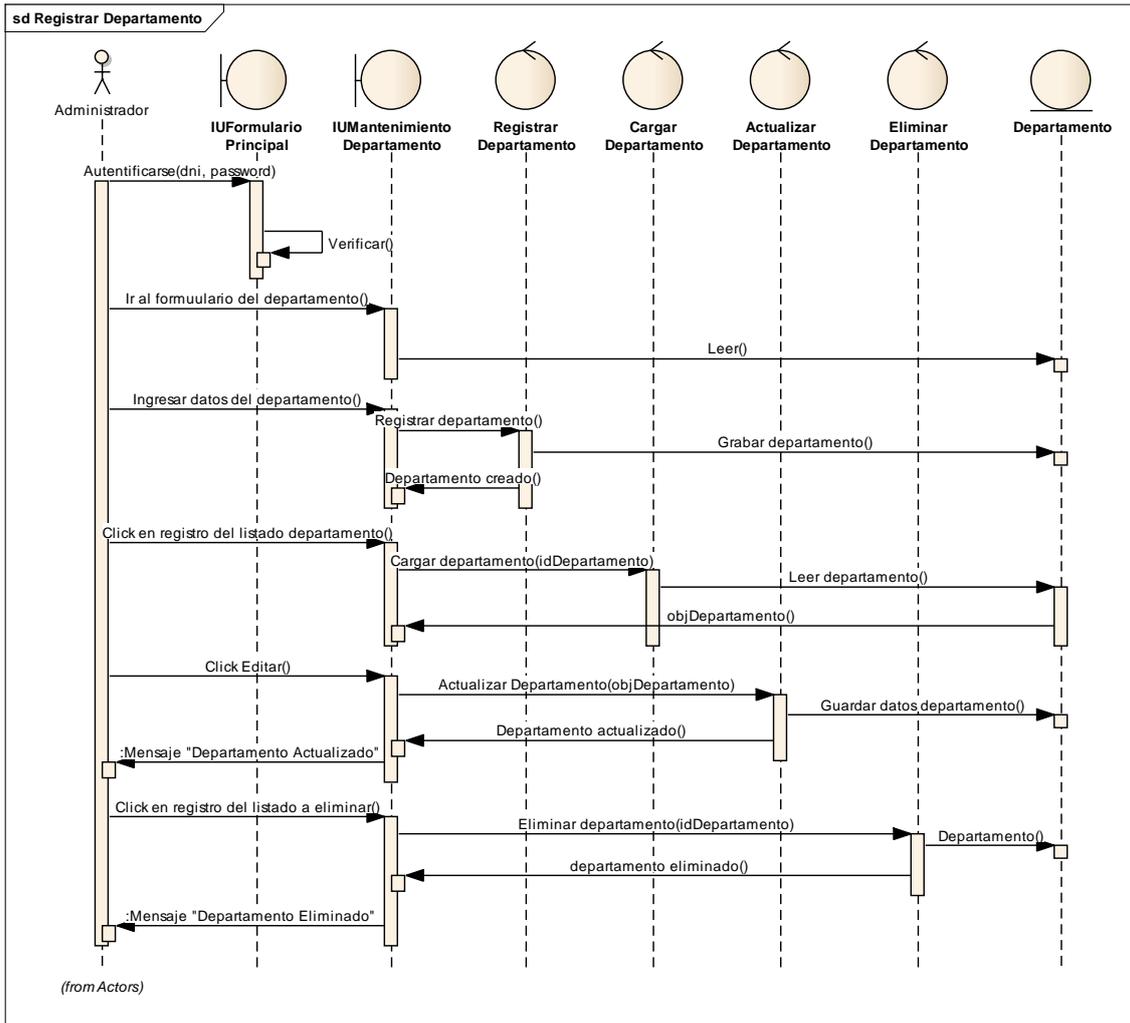


Fig. 85 Diagrama de Secuencia de Mantenimiento Departamento

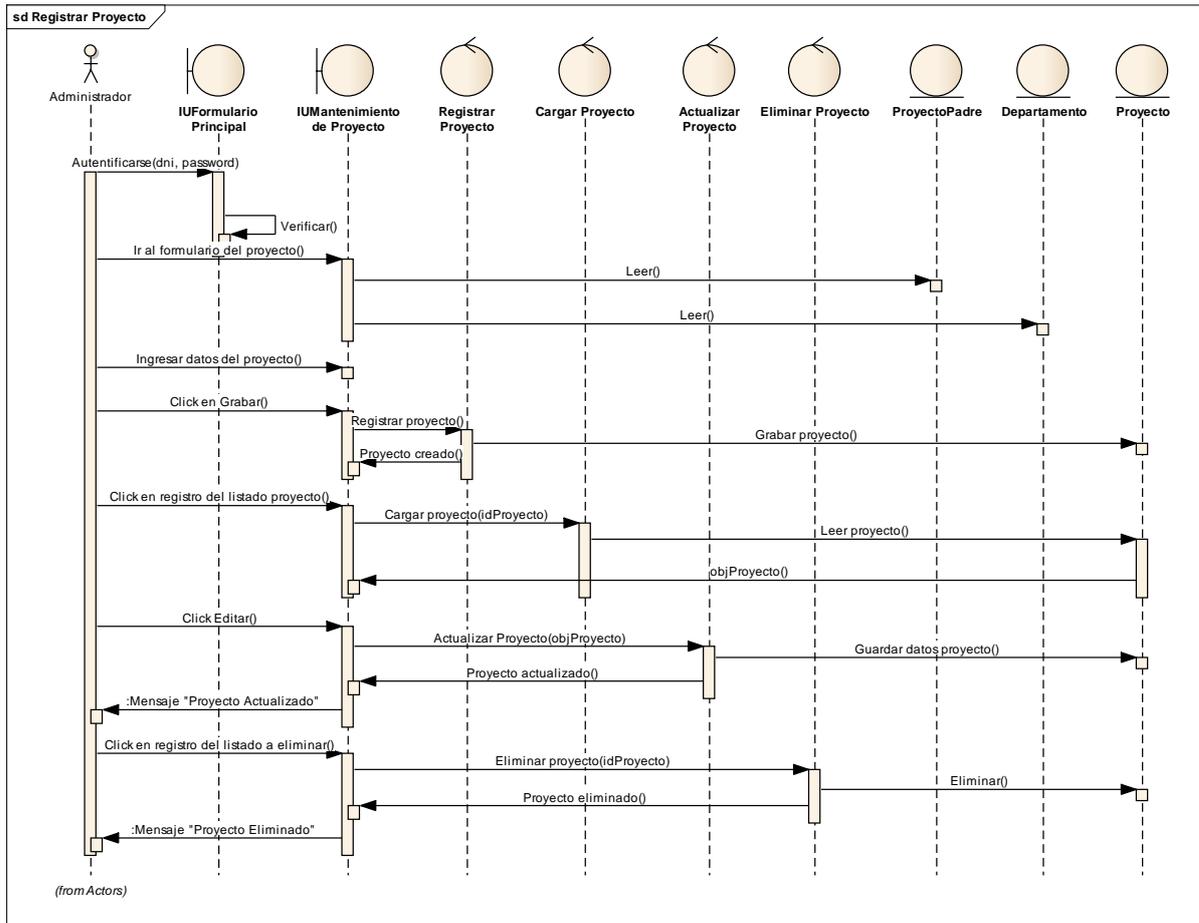


Fig. 86 Diagrama de Secuencia de Mantenimiento Proyecto

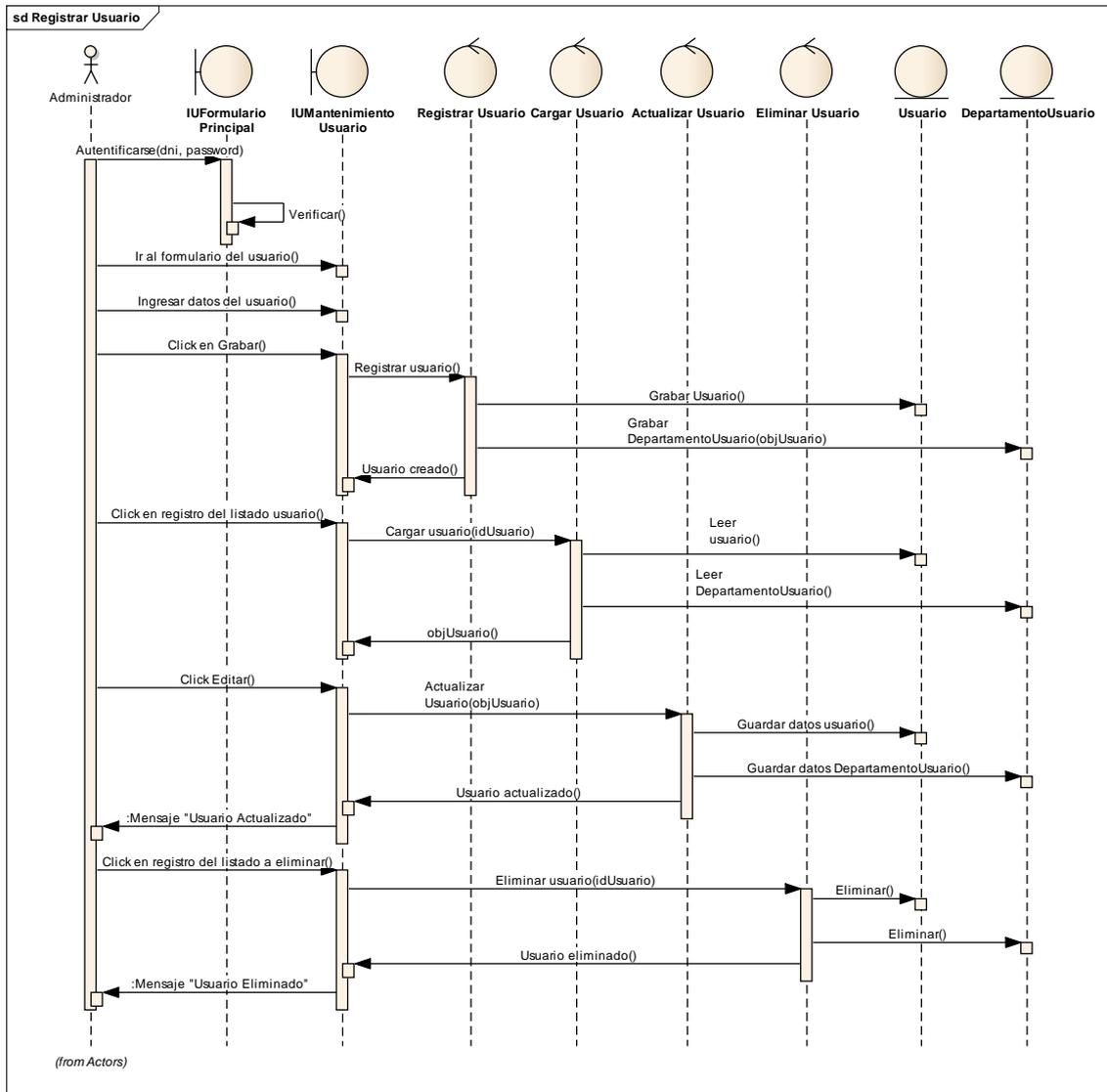


Fig. 87 Diagrama de Secuencia de Mantenimiento Usuario

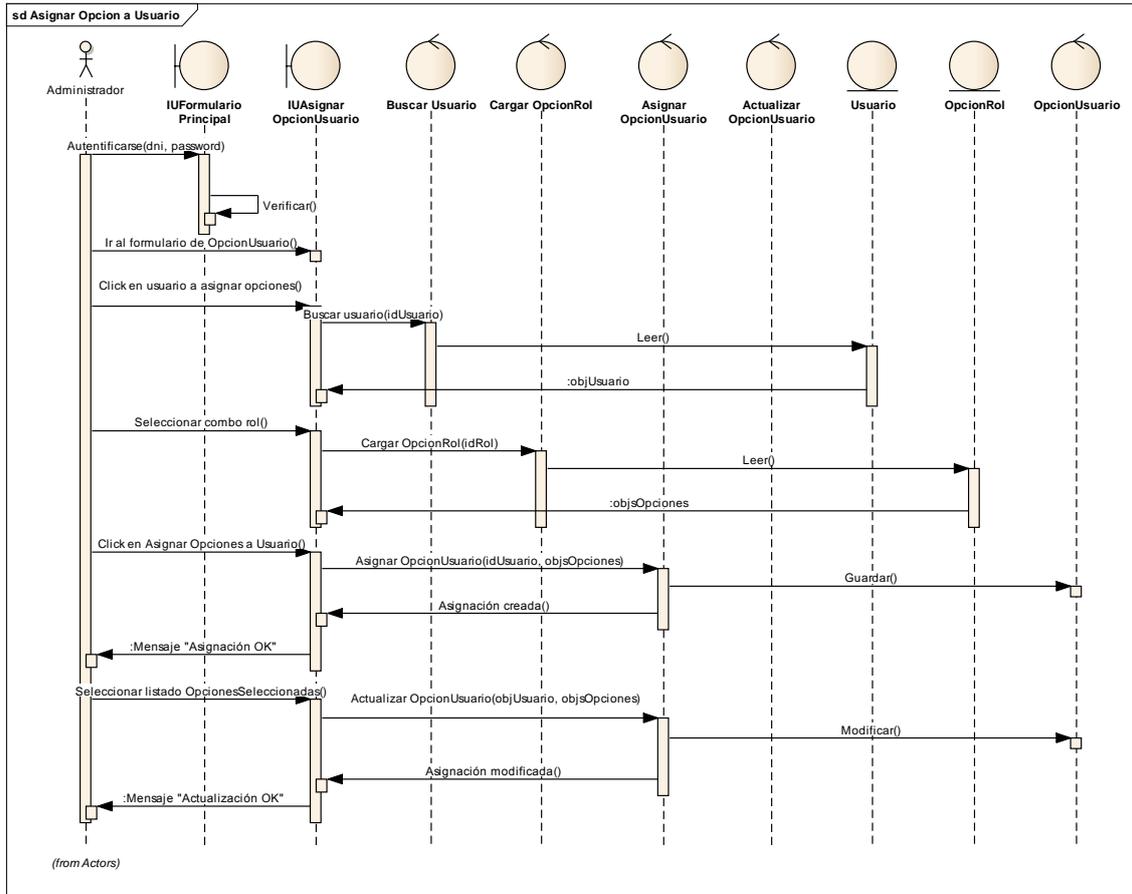


Fig. 88 Diagrama de Secuencia de Asignar Opción a Usuario

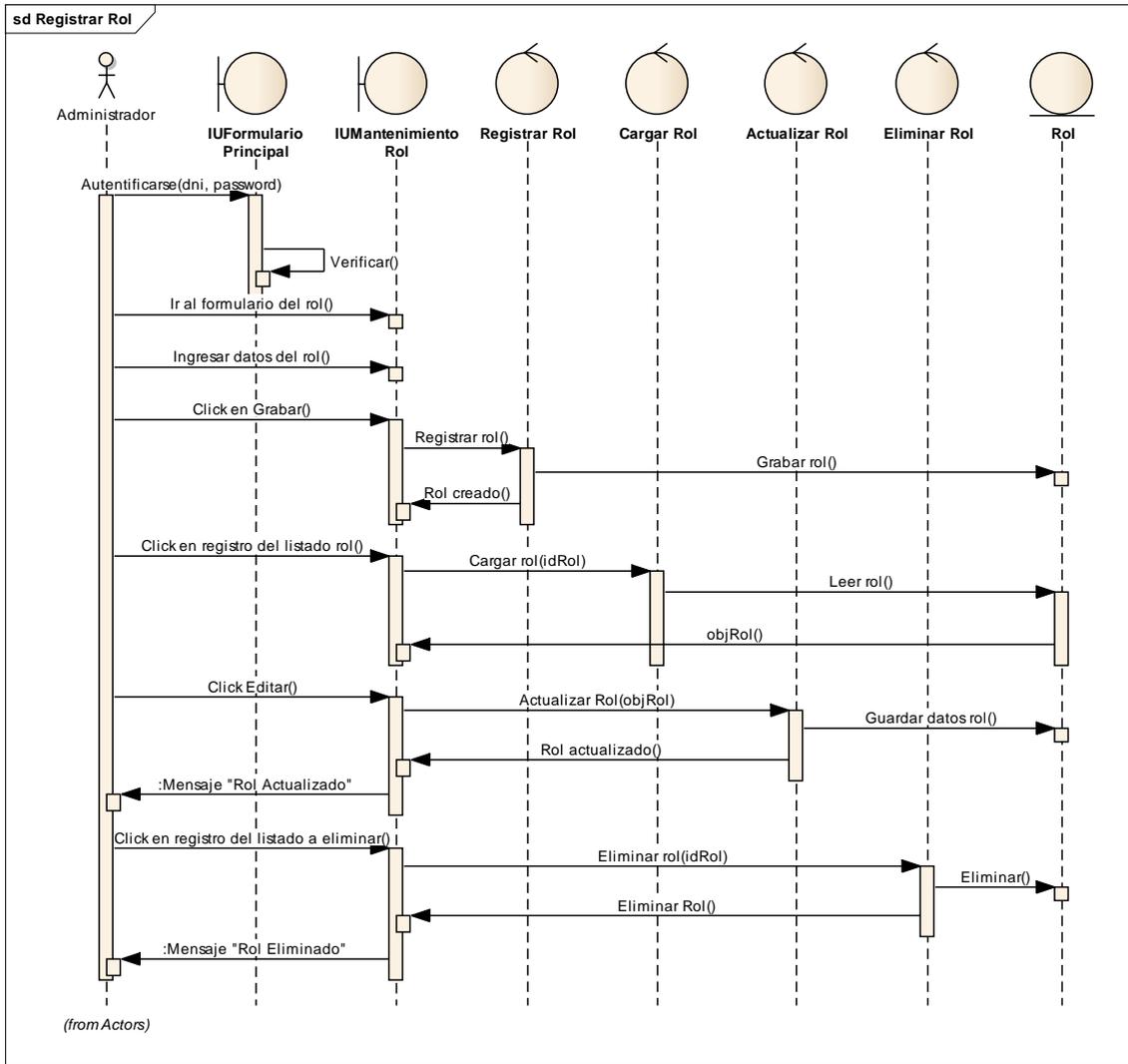


Fig. 89 Diagrama de Secuencia de Mantenimiento Rol

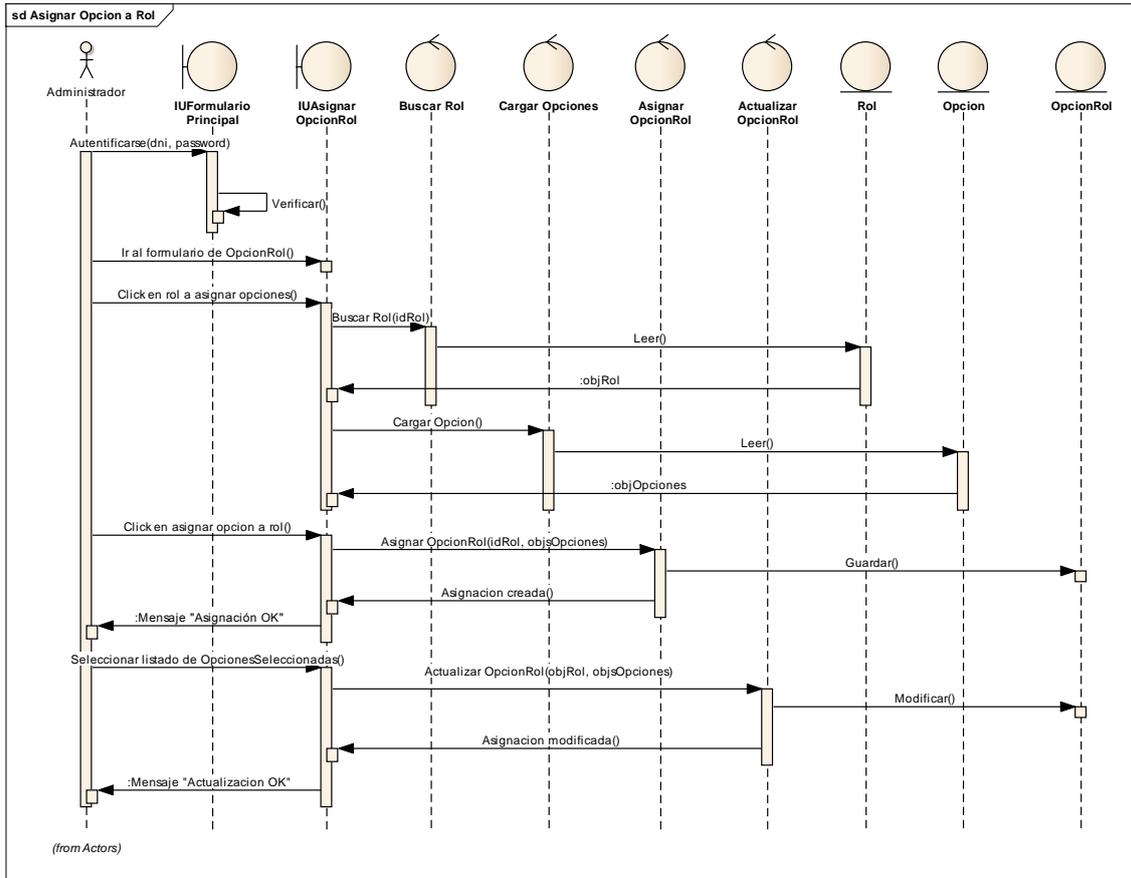


Fig. 90 Diagrama de Secuencia de Asignar Opción a Rol

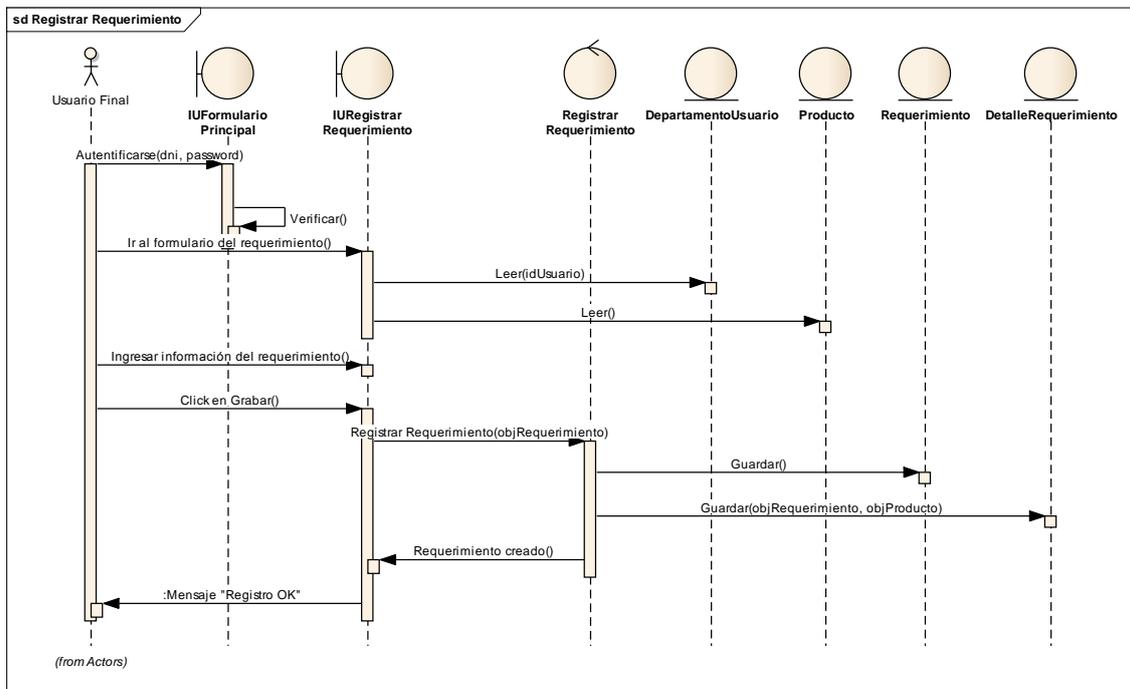


Fig. 91 Diagrama de Secuencia de Registrar Requerimiento

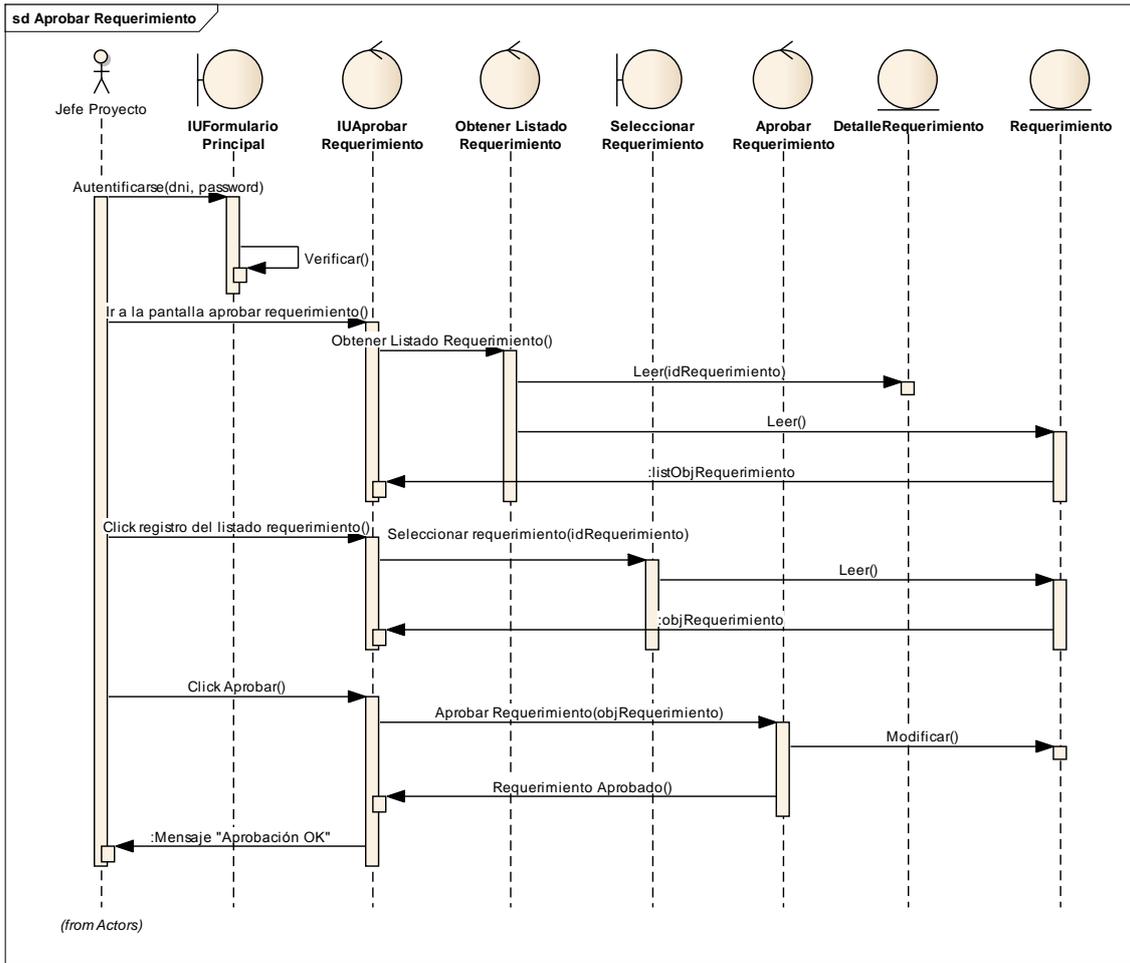


Fig. 92 Diagrama de Secuencia de Aprobar Requerimiento

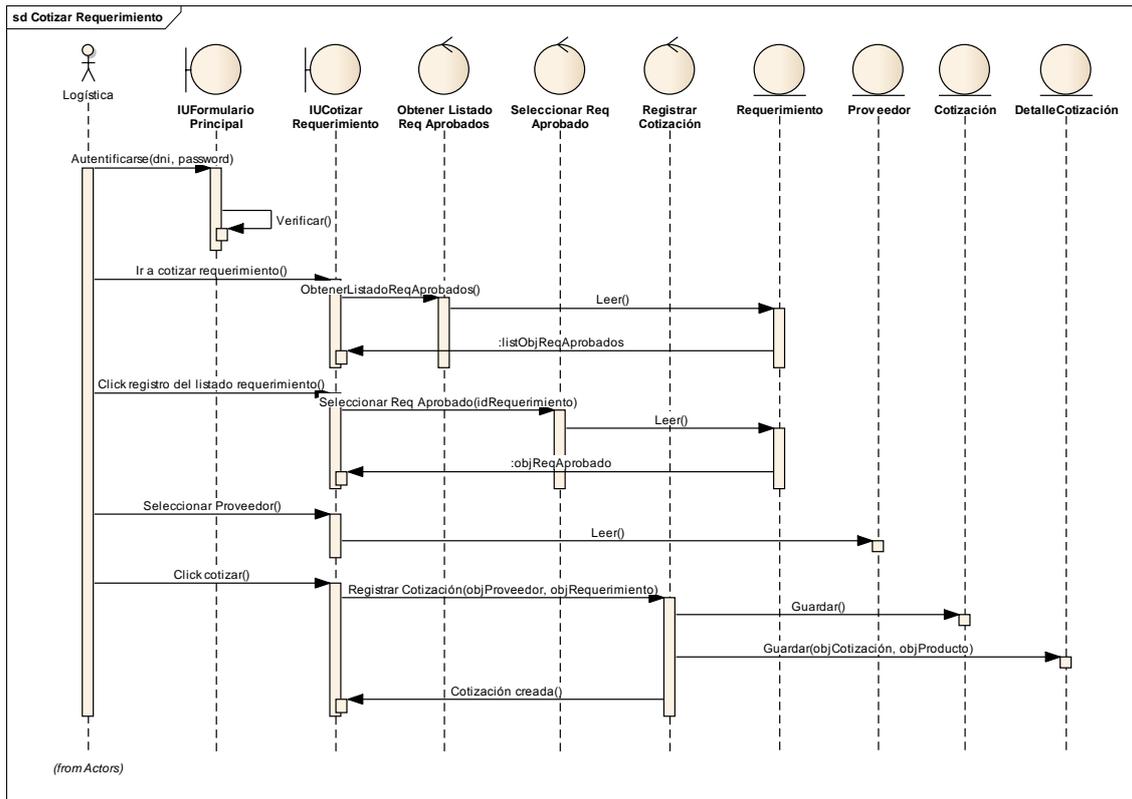


Fig. 93 Diagrama de Secuencia de Cotizar Requerimiento

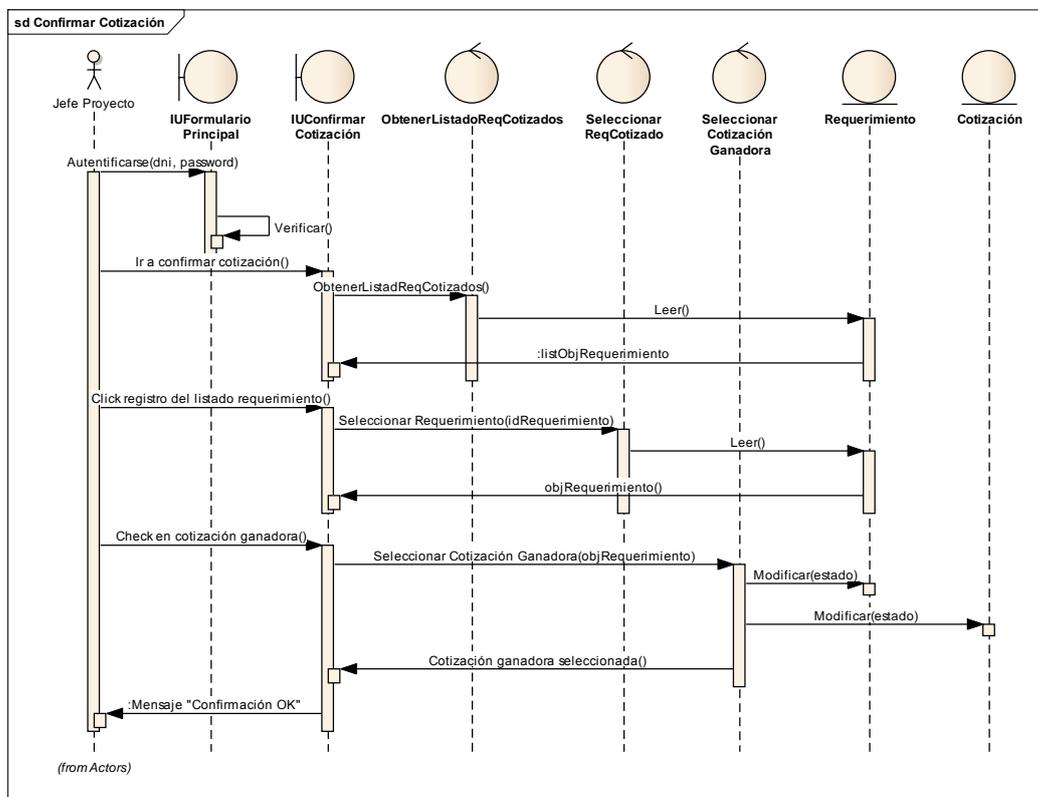


Fig. 94 Diagrama de Secuencia de Confirmar Requerimiento Cotizado

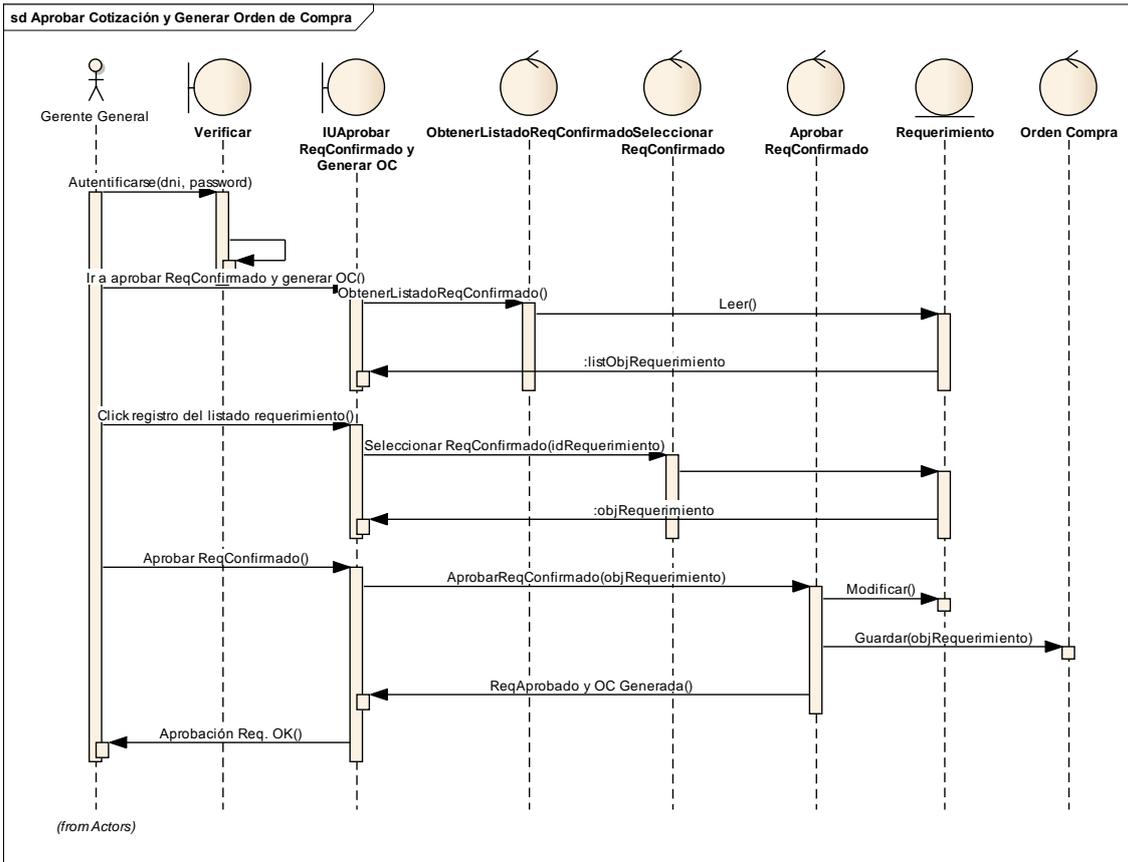


Fig. 95 Diagrama de Secuencia de Aprobar Requerimiento Confirmado y Generar Orden de Compra

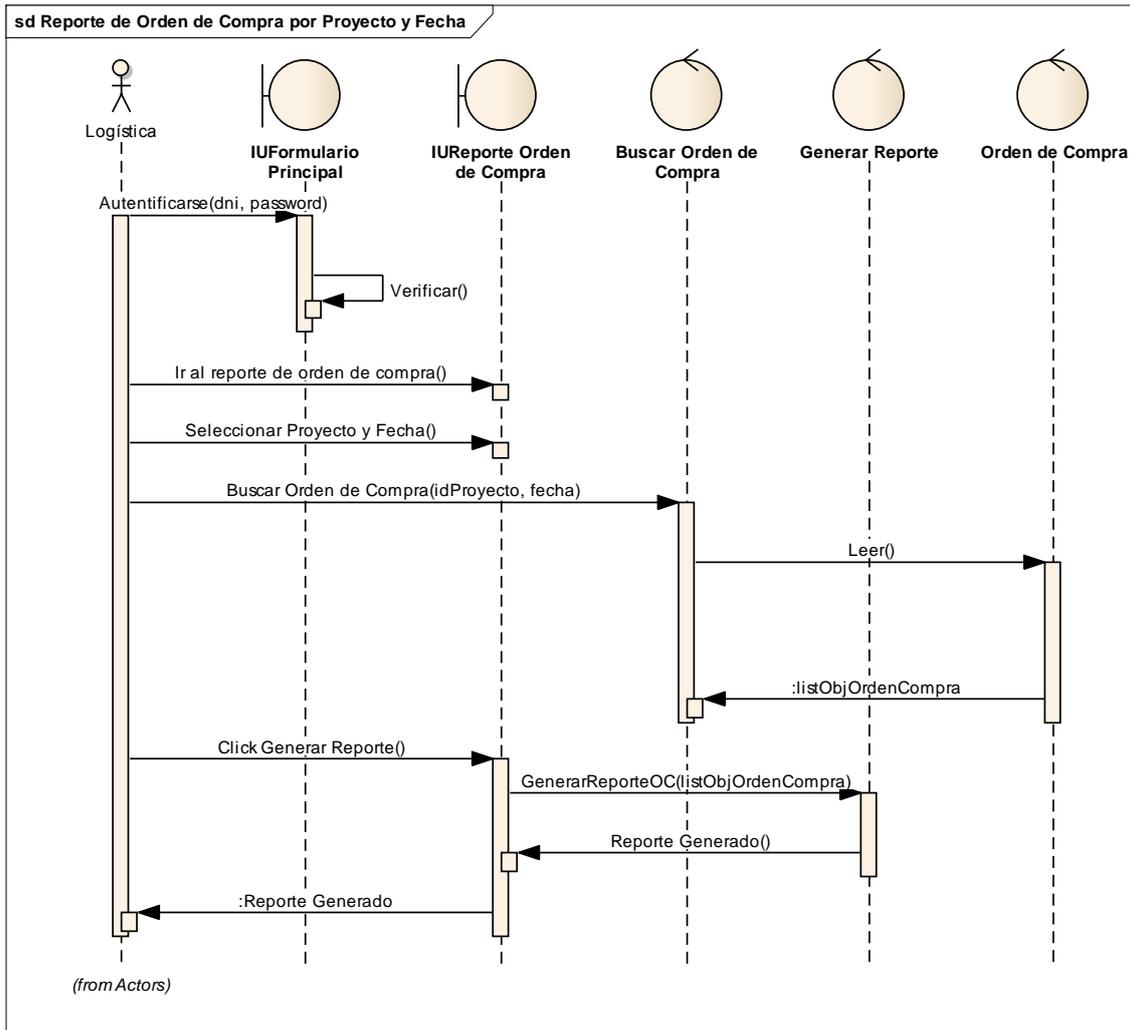


Fig. 96 Diagrama de Secuencia de Reportes de Orden de Compra por Proyecto y Fecha

2.4.3. Modelo Lógico

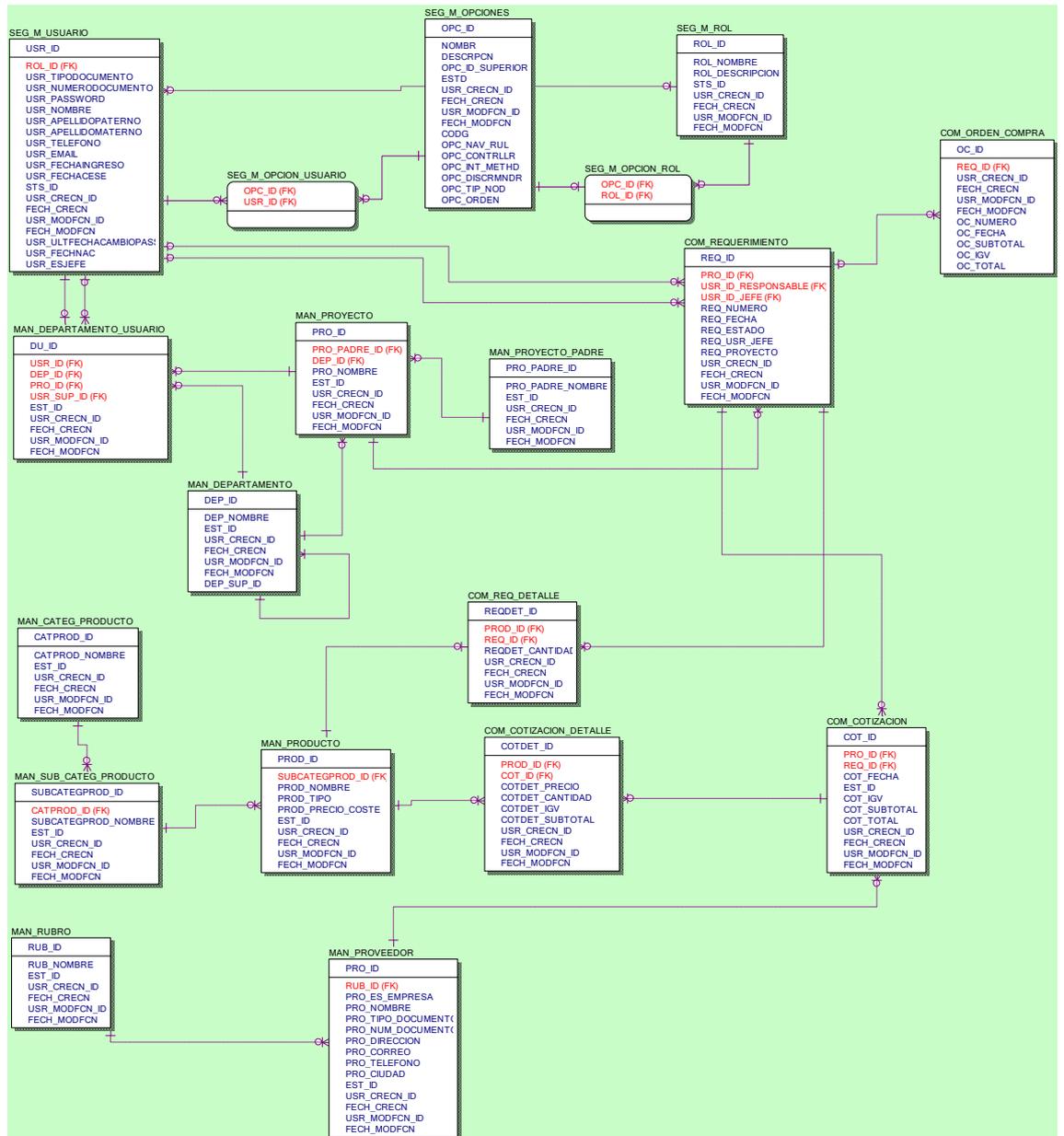


Fig. 97 Diagrama Lógico de Base de Datos

2.4.4. Modelo Físico

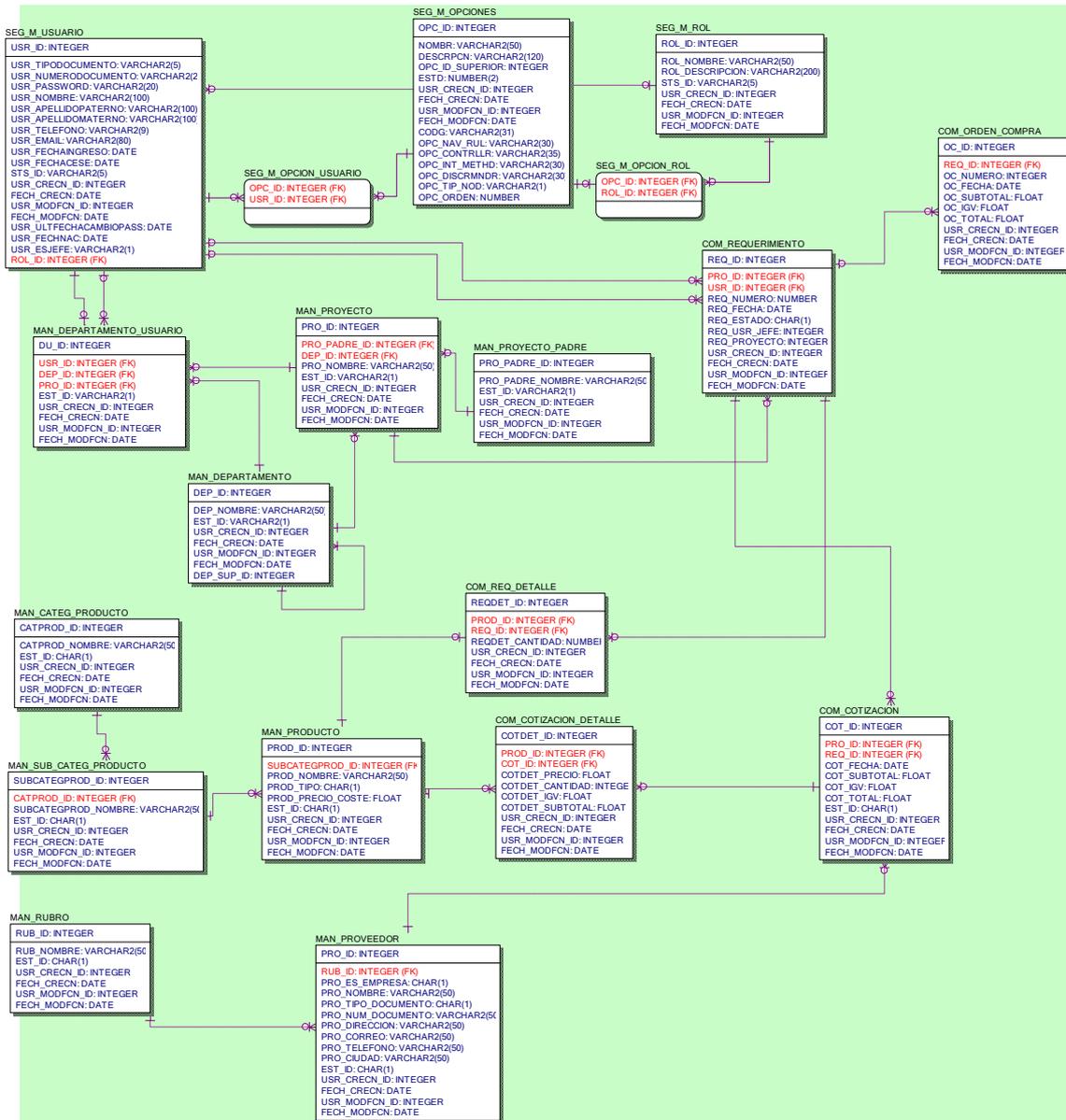


Fig. 98 Diagrama Físico de Base de Datos

2.5. Implementación

2.5.1. Diagrama de despliegue

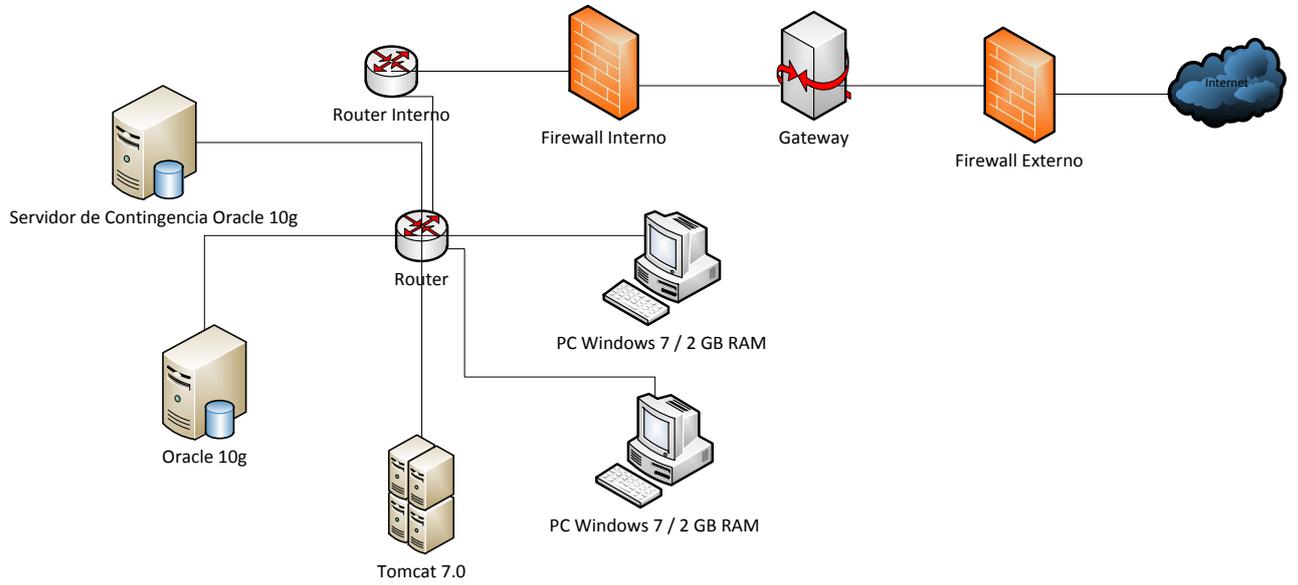


Fig. 99 Diagrama de Despliegue

2.5.2. Diagrama de Componentes

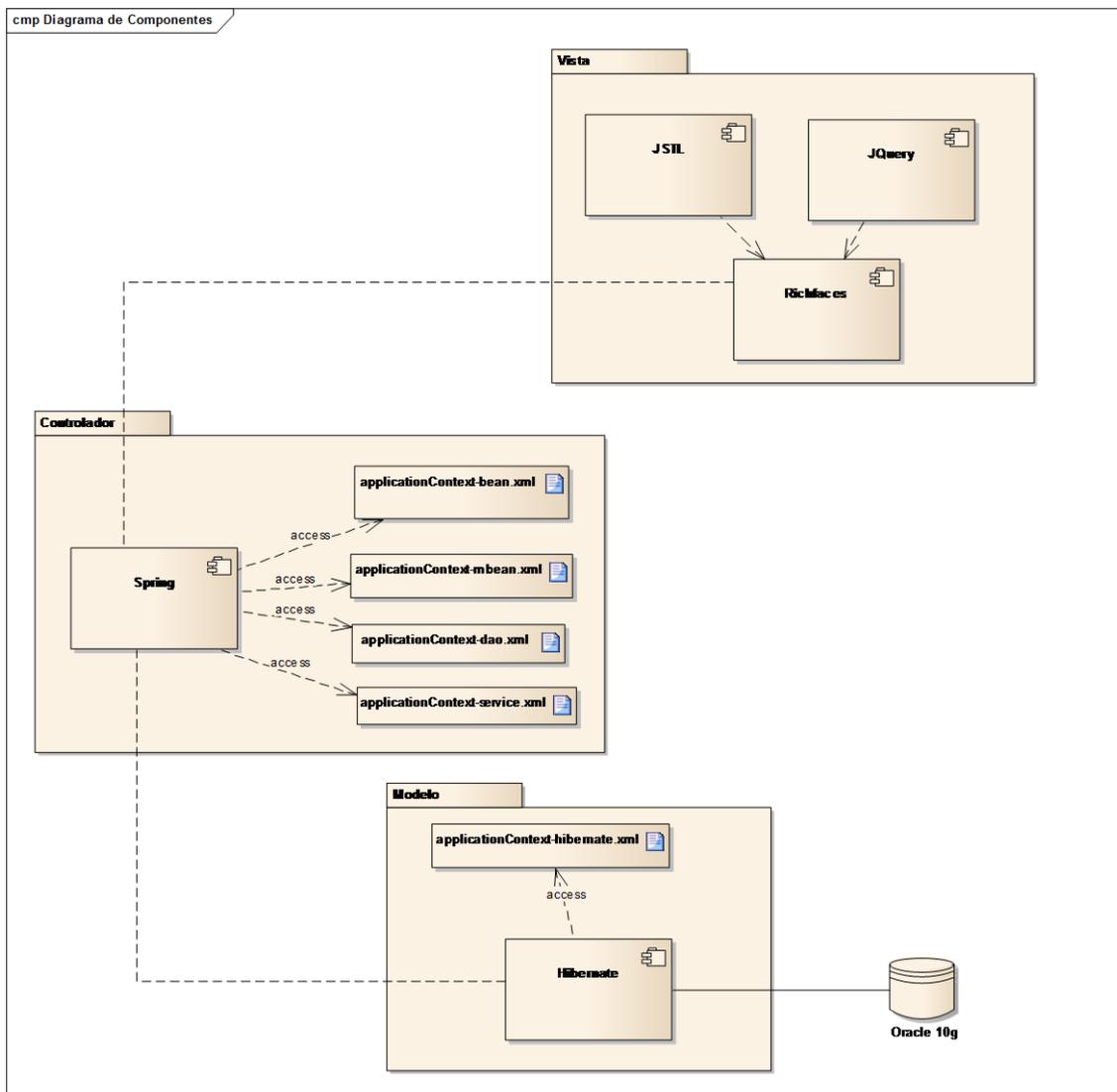


Fig. 100 Diagrama de Componentes

CAPÍTULO III: DISCUSIÓN

2.3. Contratación de la Hipótesis

2.3.1. Identificación de Variables e Indicadores

Es el Método de Contratación será: Pre-Test → Post-Test, siendo las variables las siguientes:

Variable Independiente (VI) : **Implementación del Sistema Informático**

Variable Dependiente (VD) : **Gestión de Compras**

Los indicadores de la VD se muestran en la siguiente Tabla:

Tabla 21: Indicadores

N°	INDICADOR	TIPO
1	Tiempo de Generar Órdenes de Compra	Cuantitativo
2	Tiempo de Obtención de Reportes	Cuantitativo
3	Tiempo de Aprobar Requerimientos	Cuantitativo

Fuente: Elaboración Propia

2.3.2. Método de Análisis para la Prueba de los Indicadores Cuantitativos

Todos los indicadores Cuantitativos se les aplicará la Prueba basadas en la Distribución t-student porque se va evaluar el Pre y Post en medir los indicadores y se seguirá el siguiente Método:

- a. Se definirán las variables antes y después de la implementación.
- b. Se formulará la hipótesis estadística nula y alternativa.
- c. Se determinará el nivel de significancia.
- d. Se determinará el valor de la muestra a la que se aplicará la prueba, utilizando la siguiente expresión:

$$n = \frac{N z_{\alpha/2}^2 P(1-P)}{(N-1)e^2 + z_{\alpha/2}^2 P(1-P)}$$

- e. Se obtendrán datos antes y después de la implementación y serán tabulados en la siguiente Tabla:

N	Pre Test	Post	D
i	X_{1i}	X_{2i}	$X_{1i} - X_{2i} = D_{1i}$

- f. Se realizará la prueba estadística con los datos anteriores, calculándose:

– **La Media Muestral:** $\bar{D} = \frac{D_{1i}+D_{2i}+\dots+D_{ni}}{n}$

La Varianza Muestral: $S^2_{D=} = \frac{\sum_1^n (D_{1i}-\bar{D})^2}{n-1}$

– **El Cálculo Estadístico de la Prueba:** $t_c = \frac{\bar{D}}{s_D/\sqrt{n}}$

– **El Valor Crítico de t** _(n-1).

g. Se escribirá la Conclusión de la Prueba.

A continuación se procederá con la Prueba de Hipótesis de cada Indicador.

2.3.2.1. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de Generacion de orden de compra.

a. Definición de Variables

T_A: Tiempo de Generar las órdenes de compra antes de la Implementación Sistema Informático Web.

T_D: Tiempo de Generar las órdenes de compra después de la Implementación Sistema Informático Web.

b. Hipótesis Estadística

Hipótesis H₀: No se redujo el tiempo de generación de las órdenes de compra después de la implementación del sistema informático.
(D=T_D- T_A)

$$H_0: D=0$$

Hipótesis H_a: Reducir el tiempo de generación de las órdenes de compra después de la implementación del sistema informático.

$$H_a: D<0$$

c. Nivel de Significancia

Nivel de significancia (α) escogido para la prueba de la hipótesis es del 5%. Por lo tanto el nivel de confianza (1- α = 0.95) será del 95%.

d. Tipo de Prueba a Aplicar

La muestra es 30 órdenes.

e. Resultados de la Hipótesis

Media Aritmética Prueba Pareada.

Es el promedio de todas las diferencias de los valores obtenidos antes y después de la implementación del sistema informático.

$$\bar{D} = \frac{\sum_1^{30} D_i}{n} = \frac{-515}{30} = -17.17$$

Varianza Muestral de la Prueba Pareada:

Es la diferencia al cuadrado de las diferencias obtenidas.

$$S_D^2 = \frac{140.17}{(30-1)} = 4.833$$

Calculo Estadístico de la Prueba

Reemplazando los valores obtenidos anteriormente calculados.

$$T_c = T = \frac{\bar{D}}{s/\sqrt{n}} = -42.768$$

Valor Crítico de t (n-1)

$$t_{(\alpha/29)} = t_{(0.05/29)} = -1.699$$

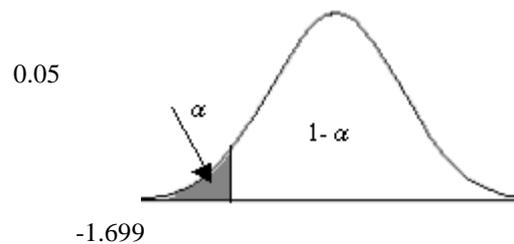


Fig. 101 Región de Aceptación y Rechazo para la prueba de Hipótesis

f. Conclusión

Puesto que $t_c = -42.768$ calculando es mayor que $t_{(0.05/29)} = -1.699$ y estando este valor dentro de la región de rechazo, entonces se rechaza H_0 y por consiguiente se acepta H_a , se concluye entonces que el tiempo de obtención del tiempo en la generación de las órdenes de compra después de la implementación del sistema informático se logró minimizar, entonces la implementación es una alternativa de solución.

2.3.2.2. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de obtención de reportes.

a. Definición de Variables

T_A : Tiempo de generar los reportes antes de la implementación sistema Informático Web.

T_D : Tiempo de Generar los reportes despues de la Implementación Sistema Informático Web.

b. Hipótesis Estadística

Hipótesis H_0 : No se redujo el tiempo de generación de los reportes después de la implementación del sistema informático. ($D = T_D - T_A$)

$$H_0: D = 0$$

Hipótesis H_a : Reducir el tiempo de generación de los reportes después de la implementación del sistema informático.

$$H_a: D < 0$$

c. Nivel de Significancia

Nivel de significancia (α) escogido para la prueba de la hipótesis es del 5%. Por lo tanto el nivel de confianza ($1 - \alpha = 0.95$) será del 95%.

d. Tipo de Prueba a Aplicar

La muestra es 30 órdenes.

e. Resultados de la Hipótesis

Media Aritmética Prueba Pareada.

Es el promedio de todas las diferencias de los valores obtenidos antes y después de la implementación del sistema informático.

$$\bar{D} = \frac{\sum_1^{30} D_i}{n} = \frac{-5858}{30} = -195.27$$

Varianza Muestral de la Prueba Pareada:

Es la diferencia al cuadrado de las diferencias obtenidas.

$$S_D^2 = \frac{42651.86}{(30-1)} = 1470.75$$

Calculo Estadístico de la Prueba

Reemplazando los valores obtenidos anteriormente calculados.

$$T_c = T = \frac{\bar{D}}{s/\sqrt{n}} = -27.888$$

Valor Crítico de t (n-1)

$$t_{(\alpha/29)} = t_{(0.05/29)} = -1.699$$

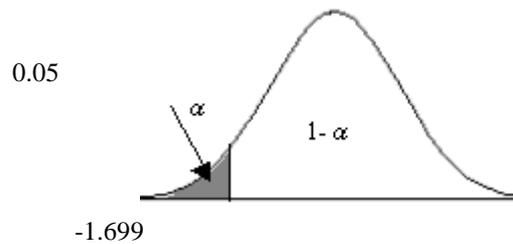


Fig. 102Región de Aceptación y Rechazo para la prueba de Hipótesis

f. Conclusión

Puesto que $t_c = -27.8888$ calculando es mayor que $t_{(0.05/29)} = -1.699$ y estando este valor dentro de la región de rechazo, entonces se rechaza H_0 y por consiguiente se acepta H_a , se concluye entonces que el tiempo de obtención de los reportes después de la implementación del sistema informático se logró minimizar, entonces la implementación es una alternativa de solución para este problema presentado.

2.3.2.3. Prueba de Hipótesis para el indicador Cuantitativo Tiempo de aprobación de requerimiento.

a. Definición de Variables

T_A :Tiempo de aprobación de requerimientos antes de la Implementación Sistema Informático Web.

T_D : Tiempo de aprobación de requerimientos después de la Implementación Sistema Informático Web.

b. Hipótesis Estadística

Hipótesis H_0 :No se redujo el tiempo aprobación de los requerimientos después de la implementación del sistema informático.
($D=T_D- T_A$)

$$H_0: D=0$$

Hipótesis H_a :Reducir el tiempo de aprobación de los requerimientos después de la implementación del sistema informático.

$$H_a: D<0$$

c. Nivel de Significancia

Nivel de significancia (α) escogido para la prueba de la hipótesis es del 5%. Por lo tanto el nivel de confianza ($1- \alpha = 0.95$) será del 95%.

d. Tipo de Prueba a Aplicar

La muestra es 30 órdenes.

e. Resultados de la Hipótesis

Media Aritmética Prueba Pareada.

Es el promedio de todas las diferencias de los valores obtenidos antes y después de la implementación del sistema informático.

$$\bar{D} = \frac{\sum_{i=1}^{30} D_i}{n} = \frac{-1836}{30} = -61.20$$

Varianza Muestral de la Prueba Pareada:

Es la diferencia al cuadrado de las diferencias obtenidas.

$$S_D^2 = \frac{3614.8}{(30-1)} = 124.65$$

Calculo Estadístico de la Prueba

Reemplazando los valores obtenidos anteriormente calculados.

$$T_c = T = \frac{\bar{D}}{s/\sqrt{n}} = -30.02$$

Valor Crítico de t (n-1)

$$t_{(\alpha/29)} = t_{(0.05/29)} = -1.699$$

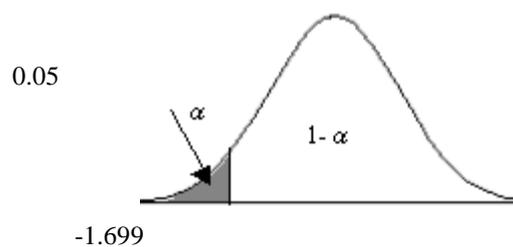


Fig. 103Región de Aceptación y Rechazo para la prueba de Hipótesis

f. Conclusión

Puesto que $t_c = -30.03$ calculado es mayor que $t_{(0.05/29)} = -1.699$ y estando este valor dentro de la región de rechazo, entonces se rechaza H_0 y por consiguiente se acepta H_a , se concluye entonces que el tiempo de generación de los requerimientos después de la implementación del sistema informático se logró minimizar, entonces la implementación es una alternativa de solución.

2.4. Discusión de Resultados

2.4.1. Minimizar el Tiempo de Generar Órdenes de Compra:

Tabla 22: Resultado para el indicador Minimizar el Tiempo de Generar Órdenes de Compra

Antes		Después		Nivel de Impacto	
Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)
15.6	100	2.8	17.95	12.8	82.1

Fuente: Elaboración Propia

Se puede observar que el indicador Tiempo de generación de las órdenes de compra antes de la implementación del Sistema Informático en promedio es 15.6 horas y el tiempo después de la implementación del Sistema propuesto es de 2.8 horas, lo que representa un decremento de 12.8 horas y en porcentaje de 82.1%

2.4.2. Minimizar el Tiempo de Obtención de Reportes

Tabla 23: Resultado para el indicador Minimizar el Tiempo de Obtención de Reportes

Antes		Después		Nivel de Impacto	
Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)
1.8	100	0.78	43.33	1.0	56.7

Fuente: Elaboración Propia

Se puede observar que el indicador Tiempo de obtención de reportes antes de la implementación del Sistema Informático en promedio es 1.8 horas y el tiempo después de la implementación del Sistema propuesto es de 0.78 horas, lo que representa una reducción de 1.0 hora y en porcentaje de 56.7%

2.4.3. Minimizar el Tiempo de Aprobar Requerimientos

Tabla 24: Resultado para el indicador Minimizar el Tiempo de Aprobar Requerimientos

Antes		Después		Nivel de Impacto	
Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)	Tiempo (hr)	Porcentaje (%)
3.8	100	0.7	18.42	3.1	81.6

Fuente: Elaboración Propia

Se puede observar que el indicador Tiempo en aprobar los requerimientos antes de la implementación del Sistema Informático en promedio es 3.8 horas y el tiempo

después de la implementación del Sistema propuesto es de 0.7 horas, lo que representa una reducción de 3.1 hora y en porcentaje de 81.6%.

CASOS DE PRUEBAS

ID Caso de Prueba	Requerimiento a Probar	Descripción	Pre-Requisitos	Resultado Esperado	Resultado Obtenido	Estado
CP001	Registrar Requerimiento	Verificar que el usuario final registre el requerimiento y se autogenera el número de requerimiento.	<ul style="list-style-type: none"> El usuario del sistema que realizará el registro del requerimiento deberá estar logueado. 	OK	OK	Concluido
CP002	Aprobar Requerimiento	Verificar que el jefe de proyecto apruebe el requerimiento en estado nuevo.	<ul style="list-style-type: none"> El usuario del sistema que realizará el registro del requerimiento deberá estar logueado. Deberá existir requerimientos en estado nuevo. 	OK	OK	Concluido
CP003	Cotizar Requerimiento	Verificar que el usuario de logística realice la cotización sobre los requerimientos en estado aprobado	<ul style="list-style-type: none"> El usuario del sistema que realizará el registro del requerimiento deberá estar logueado. Deberá existir requerimientos en estado aprobado. 	OK	OK	Concluido

CONCLUSIONES

1. Se realizó una investigación bibliográfica sobre sistema de información web y framework haciendo uso de buscadores web y libros.
2. Se recopiló información propia de la empresa, mediante la aplicación de entrevistas y observaciones, lo cual permitió identificar los procesos principales y los problemas que presentaba la Empresa.
3. Se realizó la definición de los requerimientos funcionales del sistema utilizando la metodologíaalconix, identificándose 19 requerimientos funcionales, 18 diagramas de robustez, 18 diagramas de secuencia.
4. La implementación del sistema ayudó a centralizar y explotar la información para la buena toma decisiones y optimización del proceso de compras.
5. Se realizaron con éxito las pruebas con el usuario final por cada requerimiento implementado.

RECOMENDACIONES

1. Mejorar proyectos pequeños con metodologías de modelamiento ágiles y libres como es el caso de Iconix.
2. Continuar con el estudio, uso y desarrollo de los Frameworks Spring, Hibernate y Richfaces, a fin de que se desarrolle, madure e implante como herramienta metodológica de diseño en creaciones de aplicaciones web.
3. Es necesario el uso de entrevistas y observaciones para realizar el estudio del contexto y de las necesidades del Sistema.
4. Para realizar el análisis de requerimientos y el diseño, se recomienda utilizar el Lenguaje de Modelamiento Unificado (UML), por ser el estándar para modelar distintos tipos de Sistemas.
5. Diseñar interfaces gráficas interactivas, que permitan un fácil manejo del sistema con el usuario.

REFERENCIAS

1. Richard Dios, (2009). Sistemas [En-línea]. Disponible en <http://www.monografias.com/trabajos87/sistemas-general/sistemas-general.shtml>. [Accesado el 31 de Mayo 2013].
2. Monografías, (2006). Teoría de Sistemas [En-línea]. Disponible en <http://www.monografias.com/trabajos11/teosis/teosis.shtml>. [Accesado el 31 de Mayo 2013].
3. Alfonso, (2010). Sistema Computacional [En-línea]. Disponible en <http://www.slideshare.net/SuperFonso/intro-sistema-informatico>. [Accesado el 31 de Mayo 2013].
4. Wikipedia, (2013). Sistema Informático [En-línea]. Disponible en http://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico#Estructura. [Accesado el 31 de Mayo 2013].
5. Instituto Tecnológico Tijuana, (2009). Web [En-línea]. Disponible en <https://sites.google.com/site/ingenieriaensistemassite/web>. [Accesado el 31 de Mayo 2013].
6. Azhlee, (2009). Web 2.0 y Web 3.0 [En-línea]. Disponible en <http://www.slideshare.net/Azhlee/web-20-y-web-30>. [Accesado el 31 de Mayo 2013].
7. Ricardo Marmolejo, (2008). Aplicaciones Web con UML [En-línea]. Disponible en <http://blogricardo.files.wordpress.com/2008/11/trabajo.pdf>. [Accesado el 31 de Mayo 2013].
8. Wikipedia, (2013). Lenguaje Unificado de Modelado [En-línea]. Disponible en http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado#Estructura. [Accesado el 31 de Mayo 2013].
9. Carla de San Martín, (2009). Metodología Iconix [En-línea]. Disponible en <http://www.portalhuarpe.com.ar/Seminario09/archivos/MetodologiaCONIX.pdf>. [Accesado el 31 de Mayo 2013].

10. Wikipedia, (2013). Java [En-línea]. Disponible en [http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)). [Accesado el 31 de Mayo 2013]
11. Wikipedia, (2013). Java EE [En-línea]. Disponible en http://es.wikipedia.org/wiki/Java_EE#APIs_generales. [Accesado el 31 de Mayo 2013]
12. DocsOracle, (2013). J2EE Containers [En-línea]. Disponible en <http://docs.oracle.com/javaee/1.3/tutorial/doc/Overview4.html>. [Accesado el 31 de Mayo 2013]
13. Wikipedia, (2013). Framework [En-línea]. Disponible en <http://es.wikipedia.org/wiki/Framework>. [Accesado el 31 de Mayo 2013]
14. Junta de Andalucía, (2013). Patrón Modelo Vista Controlador [En-línea]. Disponible en <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>. [Accesado el 31 de Mayo 2013]
15. Lbcnet, (2013). Esquema de funcionamiento del patrón de diseño MVC [En-línea]. Disponible en <http://lbcnet.wordpress.com/>. [Accesado el 31 de Mayo 2013]
16. Fernando Briano, (2010). Introducción a Spring Framework Java [En-línea]. Disponible <http://picandocodigo.net/2010/introduccion-a-spring-framework-java/>. [Accesado el 31 de Mayo 2013]
17. José Sánchez, (2010). Introducción a Richfaces [En-línea]. Disponible <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsflntro>. [Accesado el 31 de Mayo 2013]
18. Wikipedia, (2013). Hibernate [En-línea]. Disponible en <http://es.wikipedia.org/wiki/Hibernate>. [Accesado el 31 de Mayo 2013]
19. JoeJil, (2010). Hibernate [En-línea]. Disponible en <http://joeljil.wordpress.com/>. [Accesado el 31 de Mayo 2013]
20. Roberto Velasco, (2013). Oracle [En-línea]. Disponible en <http://www2.rhernando.net/modules/tutorials/doc/bd/oracle.html>. [Accesado el 31 de Mayo 2013]

21. Wikipedia, (2013). Servidor HTTP Apache [En-línea]. Disponible en http://es.wikipedia.org/wiki/Servidor_HTTP_Apache#M.C3.B3dulos. [Accesado el 31 de Mayo 2013]