



UNIVERSIDAD NACIONAL DE LOJA

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES**

CARRERA DE INGENIERÍA EN SISTEMAS

TÍTULO:

**“Software para la gestión de información del estado
mecánico de los vehículos del Ministerio de Transporte
y Obras Públicas”**

*Tesis previa a la obtención del
Título de Ingeniero en Sistemas.*

AUTORAS

*Carla Lisbeth Romero Córdova
Yesica Norelly Saquicela Celi*

DIRECTOR:

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

**LOJA - ECUADOR
2014**

Certificación del Director

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

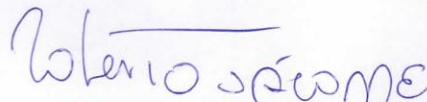
DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS

CERTIFICA

Haber dirigido, revisado y corregido en todas sus partes el desarrollo de la tesis de Ingeniería en Sistemas titulada: **“SOFTWARE PARA LA GESTIÓN DE INFORMACIÓN DEL ESTADO MECÁNICO DE LOS VEHÍCULOS DEL MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS”**, con autoría de las egresadas **Yesica Norelly Saquicela Celi y Carla Lisbeth Romero Córdova**.

En razón de que la misma reúne a satisfacción los requisitos de fondo y forma, exigidos para la investigación de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal designado para el efecto.

Loja, 12 de noviembre del 2014.



.....
Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Nosotras, **Yesica Norelly Saquicela Celi** y **Carla Lisbeth Romero Córdova**, declaramos ser autoras del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación de nuestra tesis en el Repositorio Institucional-Biblioteca Virtual.

Autor: Yesica Norelly Saquicela Celi

Firma:.....

Cédula: 1104871882.

Autor: Carla Lisbeth Romero Córdova

Firma:.....

Cédula: 1104009871

Fecha: 3 de diciembre del 2014.

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LAS AUTORAS, PARA LA CONSULTA, REPRODUCCION PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Nosotras, CARLA LISBETH ROMERO CÓRDOVA y YESICA NORELLY SAQUICELA CELI declaramos ser autoras de la tesis titulada: **SOFTWARE PARA LA GESTIÓN DE INFORMACIÓN DEL ESTADO MECÁNICO DE LOS VEHÍCULOS DEL MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS**, como requisito para optar el grado de: **Ingeniero en Sistemas**; autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los cuatro días del mes de diciembre del dos mil catorce.

Firma:.....

Autor: Yesica Norelly Saquicela Celi

Cédula: 1104871882.

Dirección: Loja (Celi Román: Ramón Burneo y José María Riofrío)

Correo electrónico:

ynsaquicelac@unl.edu.ec

Teléfono:072578238 **Celular:**0939141680

Firma:.....

Autor: Carla Lisbeth Romero Córdova

Cédula: 1104009871

Dirección: Loja (Ciudad Victoria: Transito Amaguhaña y Jaime Hurtado)

Correo electrónico:

clromeroc@unl.edu.ec

Teléfono:072326003 **Celular:**0985537193

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Tribunal de Grado: Ing. Marco Augusto Ocampo Carpio, Mg. Sc.

Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Ing. Jorge Ivan Tocto, Mg. Sc.

Agradecimiento

A la UNIVERSIDAD NACIONAL DE LOJA a través de la Carrera de Ingeniería en Sistemas a sus autoridades y personal Administrativo.

A todos los profesores de quienes recibimos la formación y preparación académica así como sus experiencias desde un enfoque logístico y científico los cuales han permitido acceder al título de Ingeniero en Sistemas y de manera especial a los profesores de la carrera que merecen nuestro reconocimiento y gratitud por su destacada labor en beneficio de nuestra formación.

Así mismo agradecemos al Ministerio de Transporte y Obras Públicas de la ciudad de Loja y a su grupo de colaboradores que nos brindaron el apoyo técnico e informativo, para el desarrollo de nuestro trabajo de titulación.

Y finalmente a nuestro tutor de tesis quien ha contribuido con sus conocimientos y sabiduría para guiarnos de mejor manera para hacer posible el desarrollo de este trabajo.

A ellos gracias por su ayuda incondicional.

Las Autoras

Dedicatoria

Dedico este trabajo de titulación principalmente a Dios por haberme guiado durante mi carrera universitaria y permitirme llegar hasta este momento tan importante de mi formación profesional.

En especial a mi madre por su amor, dedicación y apoyo incondicional durante toda mi educación, a mi padre que desde el cielo ha permanecido siempre en mi corazón, a Rolando, Cristian y mis hermanas por su confianza y sacrificio que ha puesto en mí. De la misma manera agradezco a mis familiares, amigos y a una persona tan importante en mi vida Pablo. Les quedaré eternamente agradecidos y que Dios los bendiga.

Yesica

El presente trabajo de titulación les agradezco a Dios y al Divino niño quienes me han guiado en los momentos más difíciles de mi carrera universitaria y por darme la fuerza, valor y resistencia para cumplir con esta etapa de mi vida. A mis padres por su amor, apoyo, confianza y por su lucha para ver reflejados sus sueños en mí, gracias por su esfuerzo y sacrificio que han hecho durante este largo tiempo para hacer de mí una persona de bien.

A mis familiares, amigos, hermanas (o) y en especial a mi hija Sarahi y prometido Cristian quienes me han brindado día a día su apoyo incondicional.

Carla

A.Título

“SOFTWARE PARA LA GESTIÓN DE INFORMACIÓN DEL ESTADO MECÁNICO DE LOS VEHÍCULOS DEL MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS.”

B. Resumen

El presente trabajo de titulación se centra en el análisis e implementación de una Aplicación Web para la gestión de información del estado mecánico de los vehículos del departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas de la ciudad de Loja, la cual permite llevar el registro de vehículos, solicitudes de reparación y mantenimiento, requisiciones, kardex, entre otros, de una forma rápida y eficiente brindando confianza y facilidad para que los usuarios realicen sus procesos. También se realiza la evaluación del estado mecánico de los vehículos en base a la valoración del estado de sus partes mecánicas, para obtener reportes del estado mecánico y de ubicación de los vehículos.

El desarrollo de este trabajo empezó con el análisis de diferentes metodologías así como de las herramientas y tecnologías acordes al proyecto a realizar, de esta forma se logró determinar la metodología Iconix que consta de cuatro fases las cuales permitieron primeramente obtener los requerimientos del software y a partir de ellos realizar el diagrama de clases y modelo del dominio, también se logró obtener la descripción de los casos de uso, diagramas de robustez y de secuencia, con la finalidad de obtener mayor conocimiento sobre el flujo de las actividades.

Además se realizó el análisis de un modelo dinámico, en el cual se consideró utilizar dos niveles de adaptación dinámica, permitiendo al usuario agregar nuevas propiedades y entidades conforme vayan surgiendo nuevos requerimientos, en la sección de resultados fase 3, se indica cómo se crea inicialmente las propiedades para una clase que extiende de este modelo y también se muestra la creación de los mismos por medio de la interfaz gráfica.

Mediante las pruebas de funcionalidad y usabilidad se verificó el correcto funcionamiento del sistema según los requerimientos planteados inicialmente, con lo cual se pudo concluir que este software es de gran ayuda ya que permite aumentar la eficacia a la hora de realizar las actividades.

Summary

This research focused on the analysis and implementation of a web application for managing information on mechanical conditions of vehicles from the road maintenance department of the “Ministerio de Transporte y Obras Públicas” of Loja, Ecuador. The end result was an application that allowed for a quick and efficient way to provide confidence and ease for users. The research focus and application design was primarily based on the utilization of vehicle registry, repair requests and requisitions, and kardex, so that vehicles can be used in a quick and efficient manner. Additionally, the evaluation of the mechanical condition of the vehicles was based on the assessment of the condition of their mechanical spares and to obtain reports from mechanical condition and location of vehicle as well.

The development of this work began with the analysis of different methodologies, tools and technologies in accordance with the project parameters. As a result, it was possible to identify and determine the four phased Iconix methodology. This research obtained the software requirements which were the starting point in making the domain paradigm. A description of the user cases, diagrams related to robustness, and flow sequence was used in creating a dynamic model.

The analysis of a dynamic model was developed which was completed by using two levels of dynamic adaptation, allowing the user to add new properties and entities as emerging new requirements as deemed necessary. Similarly, the results of the section titled Phase 3, indicates how the user can initially create properties for a class extending from this model and creating them through the graphic interface

By testing the functionality and usefulness, the system operation was verified according to the requirements and parameters initially proposed, whereby it was concluded that this software is very helpful because it allows increasing the efficiency while allowing for the information record in performing the necessary vehicle activities including those related to maintenance and use.

Índice de Contenidos

C. Introducción.....	17
D. Revisión de Literatura.....	19
1. Especificación de requisitos del software según el estándar IEEE 830.....	19
2. Metodología Iconix.....	19
3. Tecnologías y herramientas de desarrollo de software.....	19
3.1. Enterprise Architect (AE).....	19
3.2. Entorno de desarrollo integrado (IDE).....	20
3.3. Base de datos relacional.....	20
3.3.1. Java Database Connectivity (JDBC).....	20
3.4. Servidor de aplicaciones.....	21
3.5. Patrones de diseño.....	21
3.5.1. Modelo-Vista-Controlador (MVC).....	21
3.6. Java Platform, Enterprise Edition (Java EE6).....	21
3.7. Enterprise JavaBeans (EJB).....	22
3.8. API de persistencia Java (JPA).....	22
3.9. Frameworks MVC.....	23
3.9.1. Java Server Faces (JSF).....	23
3.9.1.1. Lenguaje de Expresiones.....	23
3.10. JBoss Seam.....	27
3.11. Picketlink.....	27
3.12. Maven.....	27
3.13. Bootstrap.....	28
3.14. JMeter.....	28
E. Materiales y Métodos.....	29
1. Métodos de investigación.....	29
1.1. Observación activa.....	29
1.1. Estudio de casos.....	29
2. Técnicas de recolección de información.....	29
2.1. Entrevista.....	29
2.2. Encuesta.....	30

2.3. Investigación bibliográfica	30
3. Metodología de desarrollo Iconix.....	30
F. Resultados	31
Fase 1: Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar	31
1.1. Descripción de la metodología seleccionada	33
Fase 2: Implementar la metodología seleccionada que permita guiar la gestión del proyecto	34
1. Análisis de requerimientos	34
1.1 Definir requerimientos funcionales	34
1.2 Modelado del dominio.	40
1.3 Modelado de casos de uso	43
1.4. Prototipo de la interfaz gráfica	48
1.5. Diagrama de Paquetes	54
2. Análisis y Diseño preliminar.....	54
2.1 Descripción de los casos de uso.....	54
2.2 Realiza diagrama de robustez.....	65
3. Diseño detallado	77
3.1. Realizar diagrama de secuencia.....	77
3.2. Diagrama del modelo del dominio final.....	88
3.3. Diagrama de clases final.....	90
4. Implementación.....	91
4.1 Diagrama de componentes	91
4.2 Arquitectura de la aplicación	91
4.3 Código fuente	92
Fase 3: Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la Aplicación Web.....	106
1. Pruebas de funcionalidad.....	107
2. Pruebas de usabilidad.....	115
G. Discusión	130
1. Desarrollo de la propuesta alternativa	130
1.1. Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar	130
1.2. Implementar la metodología seleccionada que permita guiar la gestión del proyecto	130
1.3. Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la Aplicación Web ..	131
2. Valoración técnica económica ambiental	132

2.1. Valoración técnica económica	132
2.2 Valoración ambiental	134
H. Conclusiones	135
I. Recomendaciones	136
I. Bibliografía	138
k. Anexos	141

Índice de figuras

Figura 1. Arquitectura de Java EE multinivel	222
Figura 2. Modelo del dominio inicial (a)	40
Figura 3. Modelo del dominio inicial (b)	41
Figura 4. Diagrama de clases inicial	42
Figura 5. Diagrama de casos de uso general	46
Figura 6. Diagrama de casos de uso del administrador	47
Figura 7. Diagrama de casos de uso del secretario	48
Figura 8. Prototipo del CU023	49
Figura 9. Prototipo del CU021	50
Figura 10. Prototipo del CU002	51
Figura 11. Prototipo del CU003	52
Figura 12. Prototipo del CU017	53
Figura 13. Diagrama de paquetes	54
Figura 14. Diagrama de robustez CU023 curso normal eventos.....	66
Figura 15. Diagrama de robustez CU023 curso alternativo de eventos (a)	67
Figura 16. Diagrama de robustez CU023 curso alternativo de eventos (b)	68
Figura 17. Diagrama de robustez CU021 curso normal de eventos	69
Figura 18. Diagrama de robustez CU021 curso alternativo de eventos	70
Figura 19. Diagrama de robustez CU002 curso normal de eventos	71
Figura 20. Diagrama de robustez CU002 curso alternativo de eventos	72
Figura 21. Diagrama de robustez CU003 curso normal de eventos	73
Figura 22. Diagrama de robustez CU003 curso alternativo de eventos	74
Figura 23. Diagrama de robustez CU017 curso normal de eventos	75
Figura 24. Diagrama de robustez CU017 curso alternativo de eventos	76
Figura 25. Diagrama de secuencia CU023 curso normal de eventos	78
Figura 26. Diagrama de secuencia CU023 curso alternativo de eventos	79
Figura 27. Diagrama de secuencia CU021 curso normal de eventos	80
Figura 28. Diagrama de secuencia CU021 curso alternativo de eventos	81
Figura 29. Diagrama de secuencia CU002 curso normal de eventos	82
Figura 30. Diagrama de secuencia CU002 curso alternativo de eventos	83
Figura 31. Diagrama de secuencia CU003 curso normal de eventos	84

Figura 32. Diagrama de secuencia CU003 curso alterno de eventos	85
Figura 33. Diagrama de secuencia CU017 curso normal de eventos	86
Figura 34. Diagrama de secuencia CU017 curso alterno de eventos	87
Figura 35. Modelo del dominio final (a)	88
Figura 36. Modelo del dominio final (b)	89
Figura 37. Diagrama de clases final	90
Figura 38. Diagrama de componentes	91
Figura 39. Arquitectura de la Aplicación	992
Figura 40. Lista de propiedades de la entidad vehículo	100
Figura 41. Crear propiedad	101
Figura 42. Crear entidad	101
Figura 43. Propiedades dentro de la clase vehículo	10102
Figura 44. Crear propiedades tipo estado parte mecánica	103
Figura 45. Evaluación del vehículo	104
Figura 46. Tabla de resultados prueba 1	108
Figura 47. Tabla de resultados prueba 2	110
Figura 48. Tabla de resultado prueba 3	111
Figura 49. Gráfica de prueba 1	113
Figura 50. Gráfica de prueba 2	113
Figura 51. Gráfica de prueba 3	114

Índice de tablas

TABLA I. CONFIGURACIÓN PARA CONECTARSE A LA BASE DE DATOS	20
TABLA II. LISTA DE OPERADORES [15]	24
TABLA III. ATRIBUTOS DE LAS EXPRESIONES [15].....	244
TABLA IV.EJEMPLO UTILIZANDO SENTENCIAS	25
TABLA V. METODOLOGÍAS ÁGILES VS TRADICIONALES [23]	31
TABLA VI.COMPARATIVA DE METODOLOGÍAS	32
TABLA VII. LISTA DE REQUERIMIENTOS	35
TABLA VIII.GLOSARIO DE TÉRMINOS.....	37
TABLA IX.LISTA ACTORES DEL SISTEMA.....	43
TABLA X. LISTA DE CASOS DE USOS	44
TABLA XI.DESCRIPCIÓN DEL CASO DE USO GESTIONAR FUNCIONALIDAD.....	55
TABLA XII. DESCRIPCIÓN DEL CASO DE USO REGISTRAR MANTENIMIENTO DEL VEHÍCULO	58
TABLA XIII. DESCRIPCIÓN DEL CASO DE USO REGISTRAR ESTADO MECÁNICO DEL VEHÍCULO	60
TABLA XIV. DESCRIPCIÓN DEL CASO DE USO REGISTRAR ESTADO DE UBICACIÓN DEL VEHÍCULO.....	61
TABLA XV. DESCRIPCIÓN DEL CASO DE USO REGISTRAR KARDEX.....	63
TABLA XVI. RESTRICCIONES DE LA TECNOLOGÍA DE LA APLICACIÓN	92
TABLA XVII. BUSQUEDA ENTIDAD EN LA BASE DE DATOS	94
TABLA XVIII. CREACIÓN DE LA ENTIDAD	94
TABLA XIX. CREACIÓN DE LA ESTRUCTURA	95
TABLA XX. CREACIÓN DE LISTA DE PROPIEDADES	95
TABLA XXI. .FIJAR PROPIEDADES Y ESTRUCTURA A LA ENTIDAD	96
TABLA XXII. CONSTRUCCIÓN DE PROPIEDADES TIPO ESTRUCTURA	96
TABLA XXIII. CREAR PROPIEDAD TIPO ESTRUCTURA COMO ENTIDAD	97
TABLA XXIV. CONSTRUCCIÓN DE PROPIEDADES	99
TABLA XXV. CONSTRUCCIÓN DE PROPIEDADES TIPO PARTE MECÁNICA	99
TABLA XXVI. OBTENER EVALUACIÓN	105
TABLA XXVII. RESUMEN DE LOS DATOS OBTENIDOS EN LAS TRES PRUEBAS	114
TABLA XXVIII. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE VEHÍCULOS	115
TABLA XXIX. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PERSONAL	117
TABLA XXX. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE SOLICITUDES DE REPARACIÓN Y MANTENIMIENTO	119
TABLA XXXI. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE REQUISICIÓN	121
TABLA XXXII. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE KARDEX	123
TABLA XXXIII. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PLANES DE	

MANTENIMIENTO.....	124
TABLA XXXIV. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PARTIDAS DE CONTABILIDAD	126
TABLA XXXV. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PRODUCTOS	127
TABLA XXXVI. RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE FUNCIONALIDADES ...	128
TABLA XXXVII. RECURSOS HUMANOS	132
TABLA XXXVIII. RECURSOS MATERIALES	133
TABLA XXXIX. RECURSOS TÉCNICOS Y TECNOLÓGICOS.....	133
TABLA XL. PRESUPUESTO DEL PROYECTO	134

C. Introducción

Actualmente uno de los principales activos que tiene toda empresa es la información, por lo cual toda organización busca la excelencia en las actividades que realiza, para así garantizar la disponibilidad, confidencialidad e integridad de sus procesos.

Por ello se ha considerado pertinente realizar un análisis en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas (MTO) de la ciudad de Loja, en el cual se logró identificar diferentes problemas, entre ellos que los usuarios realizan la digitación de información utilizando paquetes de Microsoft Office (Excel, Word, entre otros), provocando de esta forma inconsistencia y redundancia en los datos, así como pérdida de información ya que estos programas han sido creados para cubrir necesidades generales mas no específicas y a la hora de manejar mucha información se vuelven poco eficientes.

De acuerdo a los problemas encontrados se ha determinado el siguiente problema general de investigación: “¿La implementación de un software estándar parametrizable, permitirá minimizar la desorganización de la información del estado mecánico de los vehículos?”.

En la construcción de un software se considera importante empezar con un análisis de una metodología que permita determinar el ciclo de vida del desarrollo del software, Iconix es una metodología ágil y flexible que maneja suficiente documentación para tener bases sólidas en el desarrollo del software. Además es iterativa e incremental lo cual permite ir encontrando nuevas características a implementarse.

Para dar cumplimiento al presente trabajo de titulación, en este documento se desarrolla la revisión de literatura que sirvió como base para obtener suficiente información de las herramientas y tecnologías a utilizar, seguidamente se describe los materiales y métodos que se aplicaron, así como las herramientas y técnicas que facilitaron la obtención de datos que manejan los usuarios en el departamento de mantenimiento vial del MTO.

Después se detallan los resultados obtenidos de acuerdo a los objetivos específicos que permitieron alcanzar el objetivo general. Para cumplir con el primer objetivo se realizaron

cuadros comparativos en los cuales se indican las características más importantes que ayudaron a seleccionar la metodología acorde al proyecto a desarrollarse, la cual fue Iconix. El segundo objetivo se basa en el cumplimiento de las fases que propone la metodología seleccionada con la finalidad de llevar de forma ordenada el proceso de desarrollo de software. Finalmente el tercer objetivo se centra en realizar pruebas de funcionalidad y usabilidad para verificar el correcto funcionamiento de la Aplicación Web y que cubra con los requerimientos necesarios para dar solución al problema planteado.

Posteriormente se describe la discusión en la cual se plasmó de forma general los objetivos de acuerdo a los resultados obtenidos y finalmente se encuentran las conclusiones y recomendaciones que fueron expuestas de acuerdo a la culminación del presente trabajo de titulación.

D. Revisión de Literatura

1. Especificación de requisitos del software según el estándar IEEE 830.

La norma IEEE (Instituto de Ingeniería Eléctrica y Electrónica), “es una asociación mundial de técnicos e ingenieros dedicada a la estandarización y el desarrollo en áreas técnicas” [1]. Dentro de esta norma existe el estándar IEEE 830 que indica, que para todo trabajo de software es necesario entregar a los usuarios la especificación de requerimientos, los cuales deben estar correctamente documentados. El mismo es muy importante dentro del análisis de requerimientos ya que permite identificar claramente las necesidades del usuario, logrando que el analista y el usuario lleguen a un acuerdo del producto de software a conseguir.

2. Metodología Iconix.

Es de gran importancia utilizar una metodología que permita guiar el desarrollo del software, por ello se ha realizado un análisis (Ver anexo 1) mediante el cual se ha seleccionado la metodología Iconix que permite seguir un ciclo de vida iterativo e incremental, obteniendo claramente las actividades a realizar y de esta forma definir lo que se espera del programa.

3. Tecnologías y herramientas de desarrollo de software.

Es importante conocer para que sirven cada una de las herramientas y tecnologías a utilizar en la implementación del software, a continuación se explica cada una de ellas.

3.1. Enterprise Architect (AE)

Enterprise Architect (EA) es una herramienta multi-usuario, utilizada en el análisis del lenguaje de modelo unificado (UML), que ayuda a diseñar el prototipo de pantallas, diagramas de secuencia, robustez, entre otros; permitiendo obtener documentación flexible y de alta calidad. La descarga de esta herramienta la puede realizar desde la siguiente página oficial [2].

3.2. Entorno de desarrollo integrado (IDE)

Netbeans es un entorno de desarrollo integrado, escrito en el lenguaje de programación Java, es gratuito y de código abierto, sirve como editor de código, en el cual se plasma cada uno de los requerimientos obtenidos. Este IDE se lo puede descargar de la página oficial [3].

3.3. Base de datos relacional

Una base de datos “es una colección integrada de registros o archivos lógicamente relacionados consolidados en una unidad que provee datos para usos múltiples” [4]. PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia de software de libre permiso BSD (Berkeley Software Distribution). Esta base de datos sirve para almacenar la información ingresada desde el software así como realizar diferente consultas a la misma.

3.3.1. Java Database Connectivity (JDBC)

El puente de conexión a la base de datos JDBC permite establecer la conexión con la base de datos postgresQL y utilizando el lenguaje de consulta estructurado SQL (Structured Query Language) se puede realizar consultas a la misma. En la tabla I se indica el punto de conexión que se realizó desde Netbeans con la base de datos postgresQL.

TABLA I.
CONFIGURACIÓN PARA CONECTARSE A LA BASE DE DATOS.

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <datasource jndi-name="java:/datasources/mtop-ds" enabled="true"
use-      java-context="true"
pool-name="mtop-ds-pool">
    <connection-url>jdbc:postgresql:mtop1</connection-url>
    <driver>postgresql-driver</driver>
    <security>
      <user-name>mtop</user-name>
      <password>mtopmtop</password>
    </security>
  </datasource>
</datasources>
```

3.4. Servidor de aplicaciones

JBoss es un servidor de aplicaciones Java EE (Enterprise Edition) de código abierto basado en J2EE, e implementado en Java [5]. Se ejecuta continuamente en un ordenador, el cual sirve para la espera de peticiones y responde a las mismas adecuadamente. Este servidor se lo puede descargar desde la página oficial [6].

3.5. Patrones de diseño

“Un patrón de diseño es un esquema de las soluciones a problemas comunes en el desarrollo de software” [7]. A continuación se explica para qué sirve el patrón de diseño Modelo-Vista-Controlador.

3.5.1. Modelo-Vista-Controlador (MVC)

Uno de los patrones más conocidos en el desarrollo de Aplicaciones Web es el patrón MVC, el cual separa el problema en tres capas como son:

La capa del modelo.- En esta capa se tiene la lógica de control, es decir, se sabe que cosas hay que hacer pero no como hacerlas [8].

La capa controlador.- En esta capa se trabaja la lógica de negocio que es la que sabe cómo se hacen las cosas [8].

La capa vista.- En esta capa se presenta el modelo mediante páginas HTML (HyperText Markup Language) para poder interactuar con el usuario [8].

3.6. Java Platform, Enterprise Edition (Java EE6)

Java EE es un entorno de programación, utilizado para aplicaciones empresariales que define un conjunto de especificaciones de APIs Java, para el desarrollo de aplicaciones distribuidas multicapas (3 capas, n-capas) basadas en la Web, empleando el lenguaje de programación Java [9].

En la figura 1 se indica la arquitectura multi-nivel.

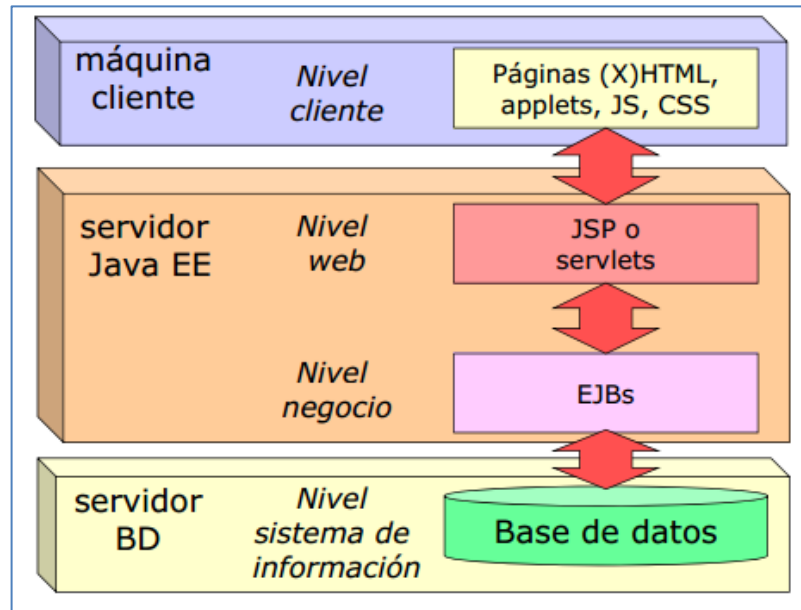


Figura 1. Arquitectura de Java EE multinivel

Java EE6 simplifica el desarrollo de aplicaciones y reduce la necesidad de programación y formación para programadores al crear componentes normalizados y reutilizables, así también permite controlar muchos aspectos de la programación automáticamente por nivel [10].

3.7. Enterprise JavaBeans (EJB)

Al momento de construir una aplicación existen algunas responsabilidades a tener en cuenta como seguridad, transaccionabilidad, concurrencia, etc. EJB es un modelo de programación que permite centrarse en la lógica del problema a resolver, además es un API que forma parte de la plataforma Java EE6 que permite la construcción de aplicaciones Java mediante objetos simples llamados POJOs (Objeto Java Plano y a la Antigua) [11].

3.8. API de persistencia Java (JPA)

Para optimizar velocidad y eficiencia de las bases de datos de las aplicaciones, es esencial mapear los objetos Java, JPA proporciona un buen modelo de persistencia y mapeo permanente basado en POJOs, que se refiere a instancias de una clase que no

extienden ni se implementan métodos abstractos de algún tipo en especial como: `HttpServlet SimpleFormController`.

La unidad de persistencia define un conjunto de todas las entidades o clases que son gestionada por la instancia `EntityManager` de una aplicación, dicho conjunto de clases representa los datos contenidos en una única base de datos [12].

3.9. Frameworks MVC.

Los frameworks ayudan a la organización de carpetas y archivos dentro del trabajo de desarrollo de software. A continuación se explica el framework JSF.

3.9.1. Java Server Faces (JSF).

Este framework de aplicaciones está basado en la Web y simplifica el desarrollo de interfaces de usuarios en aplicaciones Java EE. Cuenta con un `FrontController` (controlador central) que se encarga de manejar todas las peticiones del cliente y gestionar su ciclo de vida, dentro de sus especificaciones útiles se encuentra: un modelo de eventos en el lado del servidor, mejor soporte para Ajax, acceso a beans por nombres asignados previamente. Además permite la utilización del lenguaje de expresiones [8].

3.9.1.1. Lenguaje de Expresiones

Muchas de las veces para la presentación de las aplicaciones se necesita realizar validaciones, es decir según cierta condición XY renderice componentes como: botones, links, entradas de texto, tablas etc. Mediante la utilización del tag `render` [15].

En otros casos una simple variable booleana resulta poco eficiente por lo que es necesario realizar comparaciones lógicas que devuelvan un resultado booleano, para esto se utiliza el lenguaje de expresiones; una expresión va siempre entre los caracteres `#{expresión}`. La sintaxis de la expresión es la siguiente: `#{MiObjeto.miPropiedad}`, con la cual se accede a una propiedad dentro de un objeto, además cuenta con un operador de acceso a array para datos que representan una colección, ejemplo: `#{MiLista[2]}` o `#{MiLista[variable + 1]}`. A parte de los operadores de propiedad y elemento array, se pueden utilizar operadores aritméticos, relacionales y lógicos; en la tabla II se muestran todos los operadores que se pueden utilizar con su descripción [15].

TABLA II.
LISTA DE OPERADORES. [15]

Operador	Descripción
.	Accede a una propiedad
[]	Accede a un elemento de un array/lista
()	Agrupar una sub expresión
+	Suma
-	Resta o negación de un número
/ o div	División
% o mod	Módulo (resto)
== o eq	Comprueba igualdad
!= o ne	Comprueba desigualdad
< o lt	Comprueba menor que
> o gt	Comprueba mayor que
<= o le	Comprueba menor o igual que
>= o ge	Comprueba mayor o igual que
&& o and	Comprueba AND lógico
o or	Comprueba OR lógico
! o not	Complemento binario booleano
Empty	Comprueba un valor vacío (null, string vacío, o una colección vacía)

Entre los atributos que utilizan las sentencias tenemos los que se muestran en la tabla III:

TABLA III.
ATRIBUTOS DE LAS EXPRESIONES [15]

Atributo	Definición
varStatus	Contiene información del estado de iteración
value	Lista o colección de elementos sobre con que se va a iterar durante el bucle
var	Nombre que se asigna a la variable temporal que se utilizará en cada paso de iteración.
offset	Indica la posición de inicio del bucle dentro de la colección. Si no se

	indica se iniciara desde la primera posición.
size	Indica el tamaño de la lista sobre la que se está iterando. Debe tener un número exacto de elementos sino dará una excepción.
step	Solo se procesa los datos indicados en la lista, ejemplo si vale 2 solo se mostrara cado dos posiciones.

En la tabla IV se muestra un ejemplo sobre el manejo de las sentencias JSTL:

TABLA IV.
EJEMPLO UTILIZANDO SENTENCIAS

```

<c:forEach items="#{ui.getProperties(controladorVehiculo.instance)}"
var="p" >
  <div class="panel panel-primary">
    <div class="panel-heading"><center><h:outputLabel
value="#{p.label}"></h:outputLabel>
</center>
</div>
    <p:tab title="#{p.label}" closable="false"
titleStyleClass="clearfix" rendered="#{p.type ==
'org.mtop.modelo.dinamico.Structure' and
controladorVehiculo.obtenerAtributoDinamico(p.name)}" >
      <c:choose>
        <c:when test="#{p.type ==
'org.mtop.modelo.dinamico.Structure'}">
          <h:panelGroup layout="block" class="clearfix">
            <ui:repeat
value="#{controladorVehiculo.instance.findBussinesEntityAttribute(p.name)}"
var="a" varStatus="status">
              <h:panelGroup id="panell" rendered="#{a.property.type ne
'org.mtop.modelo.EstadoParteMecanica'}" layout="block"
class="clearfix">
                <label for="#{a.name}" class="#{(not empty
a.property.required and a.property.required) ? 'required' :
'}">#{a.property.label}#{(not empty a.property.required and
a.property.required) ? '*&#160;' : ''}
</label>
                <div class="input">
                  <p:inputText id="str" value="#{a.value}"
styleClass="form-control"
required="#{a.property.required}" label="#{a.property.label}"
rendered="#{a.property.type eq 'java.lang.String'}">
                    <f:ajax execute="str" render="panell"/>
                  </p:inputText>
                  <p:inputNumber id="str1" value="#{a.value}"
style="width: 100%"
required="#{a.property.required}" label="#{a.property.label}"

```

```

decimalPlaces="2" rendered="#{a.property.type eq 'java.lang.Double' or
a.property.type eq 'java.lang.Long' or type eq 'java.lang.Float'}">
    <f:ajax execute="str1" render="panell1"/>
</pe:inputNumber>

    <p:calendar id="str2" value="#{a.value}"
navigator="true" label="#{a.property.label}" style="width: 100%"
rendered="#{a.property.type eq 'java.util.Date'}"
required="#{a.property.required}" >
    <f:ajax execute="str2" render="panell1"/>
</p:calendar>

    <p:selectOneMenu id="str3" value="#{a.value}"
effect="fade" converter="#{omnifaces.SelectItemsConverter}"
required="#{a.property.required}"
label="#{a.property.label}" style="width: 100%"
rendered="#{a.property.type eq 'java.lang.String[]' or a.property.type
eq 'org.mtop.modelo.EstadoParteMecanica'}">
    <f:selectItems
value="#{ui.getValuesAsSelectItem(a.property.values)}" var="v"
itemLabel="#{v.label}" itemValue="#{v}"/>
    <f:ajax execute="str3" event="change"
render="panell1"/>
</p:selectOneMenu>
<p:inputTextarea id="str4" value="#{a.value}" style="width: 100%;
height: 100px;" required="#{a.property.required}"
label="#{a.property.label}"
rendered="#{a.property.type eq 'java.lang.MultiLineString'}">
    <f:ajax execute="str4" render="panell1"/>
</p:inputTextarea>

    <pe:inputNumber id="str5" value="#{a.value}"
style="width: 100%"
required="#{a.property.required}" label="#{a.property.label}"
decimalPlaces="0" rendered="#{a.property.type eq
'java.lang.Integer'}">
    <f:ajax execute="str5" render="panell1"/>
</pe:inputNumber>

    <p:selectOneButton value="#{a.value}" style="width:
100%" rendered="#{a.property.type eq 'java.lang.Boolean'}" >
    <f:selectItem itemLabel="true" itemValue="true" />
    <f:selectItem itemLabel="false" itemValue="false"
/>
</p:selectOneButton>
<br/>

    <h:graphicImage value="#{resource['visto.png']}"
rendered="#{a.value ne '' and
controladorVehiculo.instance.persistent}"
styleClass="validation-status" />
    <h:graphicImage value="#{resource['visto.png']}"
rendered="#{a.value ne null and
controladorVehiculo.instance.persistent eq false}"

```

```

styleClass="validation-status" />
        <span class="help-inline text-muted">
<ocp:message forId="str" />
            <ocp:message forId="str1" />
            <ocp:message forId="str2" />
            <ocp:message forId="str3" />
            <ocp:message forId="str4" />
            <ocp:message forId="str5" />
        </span>
<br/><br/>
    </div>
</h:panelGroup>
</ui:repeat>
</h:panelGroup>
</c:when>
</c:choose>
</p:tab>
</div>
</c:forEach>

```

3.10. JBoss Seam

Es un framework desarrollado por JBoss, Seam security proporciona un número de características útiles para dar seguridad a las aplicaciones Java EE, como son la autenticación, gestión de identidad, autenticación externa, autorización [16].

3.11. Picketlink

Proyecto para la gestión global de la seguridad y gestión de identidades para aplicaciones Java EE, entre sus principales funciones están la autenticación de los usuarios, garantizando el acceso a los métodos de negocio de la aplicación según usuarios, grupos, roles y permisos de la aplicación [17].

3.12. Maven

Este framework de gestión de proyectos de software, ayuda al proceso de creación de un proyecto Java mediante el uso de arquetipos predefinidos en el archivo xml (Pom.xml), seguidamente se descargan automáticamente las librerías necesarias, generando plugins relacionados con el jar necesario para un proyecto [18]. Este framework lo puede descargar desde la página oficial [19].

3.13. Bootstrap

Bootstrap permite crear interfaces Web con CSS (cascading style sheets) y JavaScript, en la Aplicación Web ha sido de gran utilidad porque permite que la pantalla se adapte automáticamente al tamaño de un ordenador o de cualquier dispositivo móvil, esto se denomina diseño adaptativo o Responsive Design.

Para trabajar con Bootstrap, debe descargarlo desde la página oficial [20], el archivo descargado debe descomprimirlo y luego agregarlo a su proyecto creado en Netbeans.

3.14. JMeter

JMeter es una herramienta abierta muy completa, implementada en Java que permite realizar tests de comportamiento funcional y medir el rendimiento [21]. Esta herramienta sirve para analizar el tiempo de respuesta del servidor, mediante la configuración de un servidor Proxy se puede construir un camino de navegación aleatorio y así simular la visita de varios usuarios. La descarga de esta herramienta la puede realizar desde la página oficial [22].

E. Materiales y Métodos

Para el presente trabajo de titulación se empleó diferentes métodos de investigación así como técnicas de recolección de información, las cuales permitieron realizar un estudio profundo de la información obtenida y de esta forma adquirir los conocimientos necesarios para dar solución al problema de investigación. También se utilizó una metodología de desarrollo de software, la cual es una parte esencial de todo proyecto ya que permite guiar y organizar las actividades para cumplir con los requerimientos del usuario y así obtener un software de calidad.

1. Métodos de investigación

1.1. Observación activa

Este método se utilizó para integrarse de forma activa al entorno en donde se desarrollan las actividades de los usuarios y de esta manera determinar la información necesaria para dar solución al problema de estudio.

1.1. Estudio de casos

Este método se empleó para recolectar toda la información útil, mediante la utilización de entrevistas (ver anexo 7, 8, 9, 16, 20.3, 20.5, 20.6) y así obtener un conocimiento más amplio sobre la situación actual.

2. Técnicas de recolección de información

2.1. Entrevista

Esta técnica fue utilizada para conocer de forma exhaustiva cada una de las actividades que se realizan en el departamento de mantenimiento vial y así llegar a cumplir con cada uno de sus requisitos.

2.2. Encuesta

Esta técnica se la utilizó con la finalidad evaluar la navegabilidad, entradas y salidas de datos, mensajes de validación e información, entre otros y así determinar la facilidad de uso del software.

2.3 Investigación bibliográfica

Mediante el uso de esta técnica se logró recolectar información necesaria que permita sustentar teóricamente el presente trabajo de titulación.

3. Metodología de desarrollo Iconix.

Tomando en cuenta que para todo proceso de desarrollo de software es importante aplicar una metodología que ayude a obtener calidad en los resultados, se ha creído conveniente utilizar la metodología Iconix, la cual ha proporcionado métodos para llevar el control del ciclo de vida del presente proyecto. A continuación se menciona como se ha desarrollado cada una de las fases de esta metodología:

- 1. Análisis de requerimientos:** En esta fase se aplicó el estándar IEEE 830 (ver anexo 3) el cual permitió realizar una especificación de requerimientos de forma más clara así como la identificación de los objetos del mundo real y las relaciones existentes entre ellos, para seguidamente construir los diagramas: modelo del dominio, modelo de casos de uso, prototipo de interfaz de usuario.
- 2. Análisis y diseño preliminar:** En esta fase se construyeron los diagramas de casos de uso y robustez, en los cuales se describió los flujos de la aplicación tanto normales como alternos.
- 3. Diseño:** En esta fase se verificó si los diagramas realizados satisfacen los requerimientos identificados, para lo cual se construyeron los diagrama de secuencia.
- 4. Implementación:** En esta fase se realizó en diagrama de componentes y se codificó todo lo que se ha especificado en los diagramas de las fases anteriores, así como también se realizaron las pruebas de funcionalidad y usabilidad.

F. Resultados

Fase 1: Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar.

El desarrollo de software es un proceso donde los componentes deben encajar y funcionar adecuadamente, para así maximizar la homogeneidad y calidad de los resultados, tomando en cuenta estos aspectos, la selección de la metodología es una decisión muy importante para el desarrollo del presente proyecto, para lo cual se realizó la descripción de las características más relevantes de las metodologías Scrum, Iconix, RUP, XP (ver anexo 1). A continuación se indicaran las métricas de comparación para adoptar la metodología más conveniente para el desarrollo del tipo de proyecto a implementar.

Cuadros comparativos.

- Comparativa entre metodología Ágiles y Tradicionales.

TABLA V.

METODOLOGÍAS ÁGILES VS TRADICIONALES [23].

Metodologías Ágiles	Metodologías Tradicionales
Poca documentación de modelos	Mucha documentación de modelos, el cual es esencial y se requiere su mantenimiento.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
La arquitectura del Software se va definiendo y mejorando a lo largo del proyecto.	La arquitectura se define previamente en el proyecto (análisis – diseño).

Pocos artefactos	Más artefactos
Actores con pocos roles, más genéricos y flexibles.	Actores con mayores roles específicos y funcionales.
Aplicable para proyecto pequeños, corta duración y menos de 10 integrantes.	Aplicable para proyectos de cualquier tamaño, proyectos grandes y tienen un equipo disperso.

De acuerdo a la Tabla V se puede identificar que la principal diferencia entre las metodologías tradicionales y ágiles es: la metodología tradicional se centran en un desarrollo a través del orden y la documentación extensa lo cual garantiza la calidad del producto de software en un tiempo y costo determinado, las metodologías ágiles en cambio tratan de mejorar la calidad del software a través de la comunicación directa con el cliente y de entregas tempranas de avances, utilizando poca documentación.

- Comparativa entre las metodología XP, Scrum, Iconix, RUP.

TABLA VI.
COMPARATIVA DE METODOLOGÍAS.

	Metodologías Ágiles		Metodologías Tradicionales	
	Scrum	XP	Iconix	RUP
Tamaño del Proceso.	Pequeño/ Medio	Pequeño/ Medio	Pequeño/ Medio	Medio/ Extenso
Tamaño del Equipo	Pequeño	Pequeño	Pequeño/ Medio	Medio/ Extenso
Obtención de requerimientos	Historias de usuario.	Historias de Usuario.	Casos de uso.	Casos de Uso.
Desarrollo incremental e iterativo	Sí	Sí	Sí	Sí
Documentación	Muy poca	Poca	Suficiente	Mucha

Entregas funcionales periódicas	Sí	Sí	No	No
Se basan en roles.	Sí	Sí	No	Sí
Utiliza diferentes artefactos	No	No	Sí	Sí
Curva de aprendizaje	Rápida	Rápida	Rápida	Lenta

Mediante las características que se indica en la tabla VI se ha creído conveniente justificar la elección de la metodología Iconix para el desarrollo del presente trabajo de titulación.

1.1. Descripción de la metodología seleccionada.

En base a estas comparativas se puede concluir que la metodología más adecuada a utilizar para el desarrollo de este tipo de software es Iconix, ya que la misma es utilizada para la construcción de sistemas de pequeña y mediana complejidad con la participación activa de los usuarios finales. Esta metodología en comparación con otros procesos más tradicionales es un proceso simplificado que permite unificar un conjunto de métodos orientados a objetos con el objetivo de tener un control estricto sobre todo el ciclo de vida del producto a realizar y se diferencia de las metodologías ágiles ya que mantiene todo el desarrollo del software constantemente documentado, permitiendo garantizar que los modelos empleados para las especificaciones sean comprensibles para personas no especializadas. De esta forma se irá obteniendo versiones funcionales del producto en intervalos de tiempos cortos, para ser evaluado y realizar cambios según se va desarrollando el software.

A continuación se menciona lo que se pretende conseguir utilizando esta metodología:

- Obtener la especificación de requerimientos y modelar el comportamiento de sistemas, utilizando el lenguaje de modelado unificado (UML).

- Se definirán los modelos de casos de uso, secuencia y robustez, con la finalidad de entender el flujo del sistema.
- Lograr retroactividad con el cliente, en la mitad de los procedimientos, comenzando con un prototipo en donde el analista y el cliente definirán pantallas, funcionalidades, en si lo que se espera obtener del programa.
- Relacionarse con el usuario final ya que el mismo es uno de los pilares fundamentales de las metodologías tradicionales que permite verificar el cumplimiento de los requisitos.

Fase 2: Implementar la metodología seleccionada que permita guiar la gestión del proyecto.

1. Análisis de requerimientos

La primera fase de la metodología escogida se centra en realizar un análisis del problema a resolver, se empieza obteniendo una lista de los requisitos que debe cumplir el sistema, el cual es la base principal para la construcción del diagrama de clases, modelo del dominio y el prototipo de interfaz de usuario. A continuación se detalla cada una de las actividades que se realizó en esta fase.

1.1 Definir requerimientos funcionales

Para definir los requerimientos del sistema se utilizaron entrevistas (ver anexo 7,9) con las personas interesadas en el software, además se realizó la especificación de requerimientos según el estándar IEEE 830 (ver anexo 3) el cual permitió realizar un análisis profundo de los requerimientos que se listan a continuación:

TABLA VII.
LISTA DE REQUERIMIENTOS

NÚMERO	DESCRIPCIÓN	PRIORIDAD
RF001	El sistema permitirá al administrador buscar, crear, editar vehículos .	Alta/Esencial
RF002	El sistema permitirá al administrador cambiar el estado de las partes mecánicas del vehículo .	Alta/Esencial
RF003	El sistema permitirá al administrador editar el estado de ubicación del vehículo (Equipo bueno inactivo, Equipo trabajando normalmente, Equipo trabajando con fallas, equipo en reparación, equipo para baja o remate), el lugar en donde se encuentra y la fecha por cada estado.	Alta/Esencial
RF004	El sistema permitirá al administrador asignar un conductor a un vehículo.	Alta/Esencial
RF005	El sistema permitirá al secretario buscar, crear una solicitud de reparación y mantenimiento de un vehículo, con sus respectivos, ítems y el vehículo a reparar; las puede editar y dar de baja siempre y cuando la solicitud no esté aprobada por el administrador caso contrario solamente podrá visualizarla.	Alta/Esencial
RF006	El sistema permitirá al secretario asignar requisición de reparación en una solicitud de reparación y mantenimiento.	Alta/Esencial
RF007	El sistema permitirá al secretario buscar y crear una requisición para las reparaciones de un vehículo, con sus respectivos ítems , vehículo y el número de partida que le corresponde; las podrá editar y dar de baja siempre y cuando no estén aprobadas por el	Alta/Esencial

	administrador caso contrario solo podrá visualizarlas.	
RF008	El sistema permitirá al secretario asignar solicitud de reparación y mantenimiento en una requisición de reparación.	Alta/Esencial
RF009	El sistema permitirá al secretario asignar un producto a un ítem de requisición de reparación.	Alta/Esencial
RF010	El sistema permitirá al secretario buscar, crear, editar, y desactivar/activar una partida de contabilidad.	Alta/Esencial
RF011	El sistema permitirá al secretario buscar, crear, editar, y dar de baja un producto de reparación del vehículo.	Alta/Esencial
RF012	El sistema permitirá al administrador buscar y editar kardex .	Alta/Esencial
RF013	El sistema permitirá al administrador aprobar solicitud de reparación y mantenimiento a una kardex .	Alta/Esencial
RF014	El sistema permitirá al administrador aprobar requisición de reparación a una kardex .	Alta/Esencial
RF015	El sistema permitirá al administrador buscar, crear, editar y activar planes de mantenimiento .	Alta/Esencial
RF016	El sistema permitirá al administrador conocer por medio de alertas las revisiones necesarias que debe hacer al vehículo según el plan de mantenimiento activo.	Alta/Esencial
RF017	El sistema permitirá al administrador imprimir reportes del informe mensual del estado de los vehículos , resumen del estado general de los vehículos .	Alta/Esencial
RF018	El sistema permitirá al administrador buscar, crear, editar el personal con sus diferentes tipos (administrador, secretario, conductor, mecánico, etc).	Alta/Esencial

RF019	El sistema permitirá al administrador crear, editar y desactivar las funcionalidades de acuerdo a requerimientos futuros.	Alta/Esencial
RF020	Iniciar sesión en el sistema mediante el ingreso del usuario y contraseña	Alta/Esencial

- Glosario de términos.

Antes de empezar la construcción de los diagramas primeramente procedemos a realizar una lista de los posibles objetos a formar parte del sistema como se muestra en la tabla VIII.

TABLA VIII.
GLOSARIO DE TÉRMINOS.

TÉRMINO	DEFINICIÓN
Entidad abstracta.	Una entidad abstracta es un objeto de la clase AbstractEntity [36]. Representa una visión global de los datos, la cual no va a ser representada en un sistema de base de datos.
Entidad General.	Una entidad general se describe en la estructura de la base de datos empleando un modelo de dato. [37]
Entidad.	Una entidad se compone por atributos y funciones que se diferencian de los demás.
Estructura.	Una estructura es un tipo de dato compuesto que permite almacenar un conjunto de datos de diferente tipo. [38].
Propiedad.	Es una unidad básica e indivisible de información que pertenece a una entidad, además sirve para identificar y describir a la misma.
Valor Atributo.	Valor específico que puede tomar una propiedad.

Persona.	Hace referencia al personal que labora dentro de la empresa que puede ser de tipo conductor, secretario administrador, etc.
Vehículo	Es una entidad que almacena información del vehículo.
Estado.	Describe el estado de ubicación actual del vehículo.
Plan mantenimiento.	Es un documento que contiene una lista de actividades de mantenimiento a realizar para un vehículo cada 5.000 kilómetros.
Actividades	Tareas a realizar para el mantenimiento vehicular según cada kilómetro.
Kardex	Es un documento de control de aprobación de requisiciones y solicitudes de reparación y mantenimiento.
Solicitud de reparación y mantenimiento.	Documento elaborado con el fin de informar que parte del vehículo necesita reparación o mantenimiento (ver anexo 13).
Ítem solicitud de reparación y mantenimiento	Cada uno de los puntos a solicitar, forma parte de una solicitud de reparación y mantenimiento
Requisición.	Documento elaborado con el fin de notificar la adquisición de un producto de reparación de un vehículo (ver anexo 14).
Ítem de requisición.	Cada uno de los puntos que forman parte de una requisición.
Partida contabilidad.	Serie numérica que sirve para identificar los productos, reportes e información y se utilizan para poder realizar una requisición (ver anexo 15).
Producto.	Entidad que hace referencia la información acerca de los posibles productos a agregar a un ítem de

	requisición.
Modelo dinámico	Es el modelo base a partir del cual se pueden crear nuevas propiedades y entidades, a las clases que extienden de este modelo [35].

1.2 Modelado del dominio.

- Diagrama del modelo de dominio

En la figura 2, se indica el modelo dinámico que permite al usuario realizar nuevos cambios que surjan en un futuro sin la necesidad de requerir los servicios de un programador.

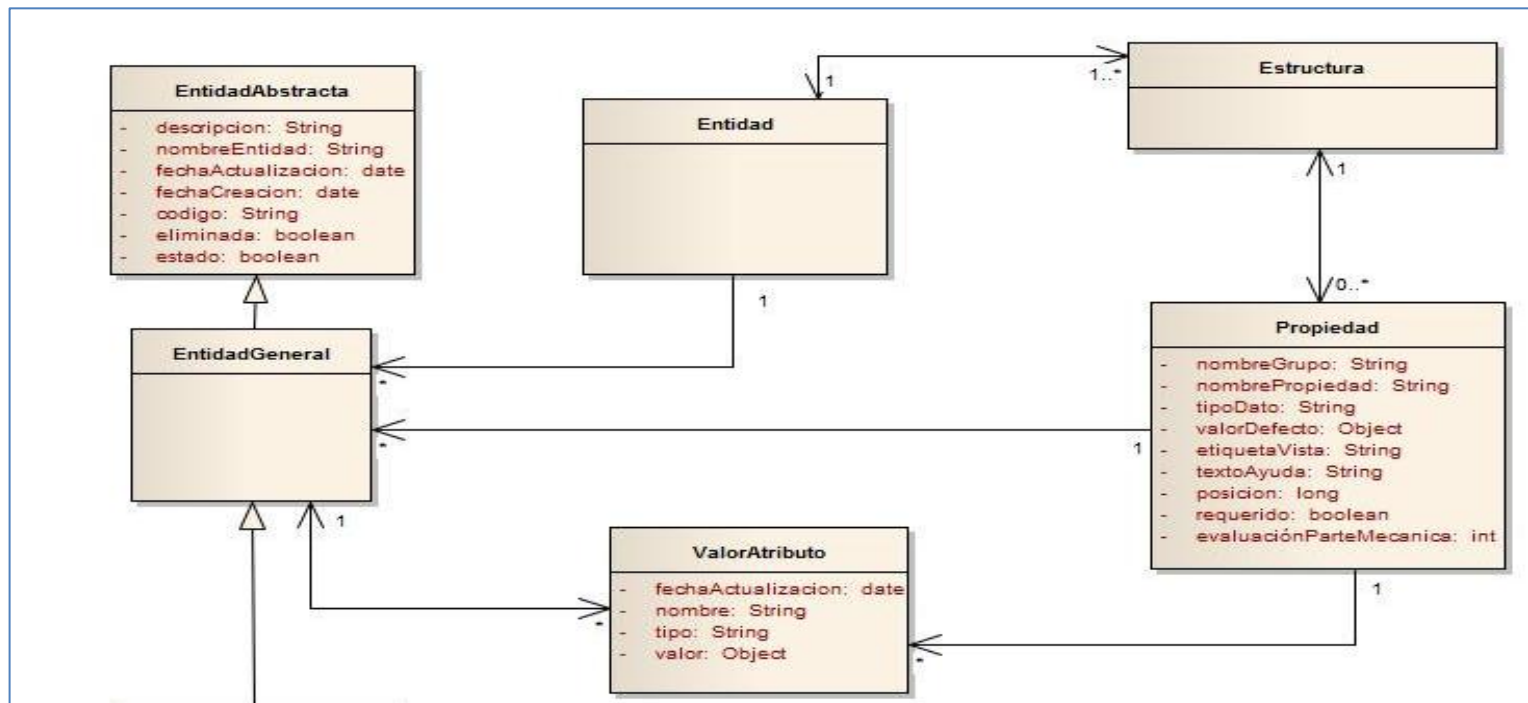


Figura 2. Modelo del dominio inicial (a) [35]

En la figura 3, se muestra las clases identificadas mediante el análisis realizado, las mismas que van a extender del modelo dinámico.

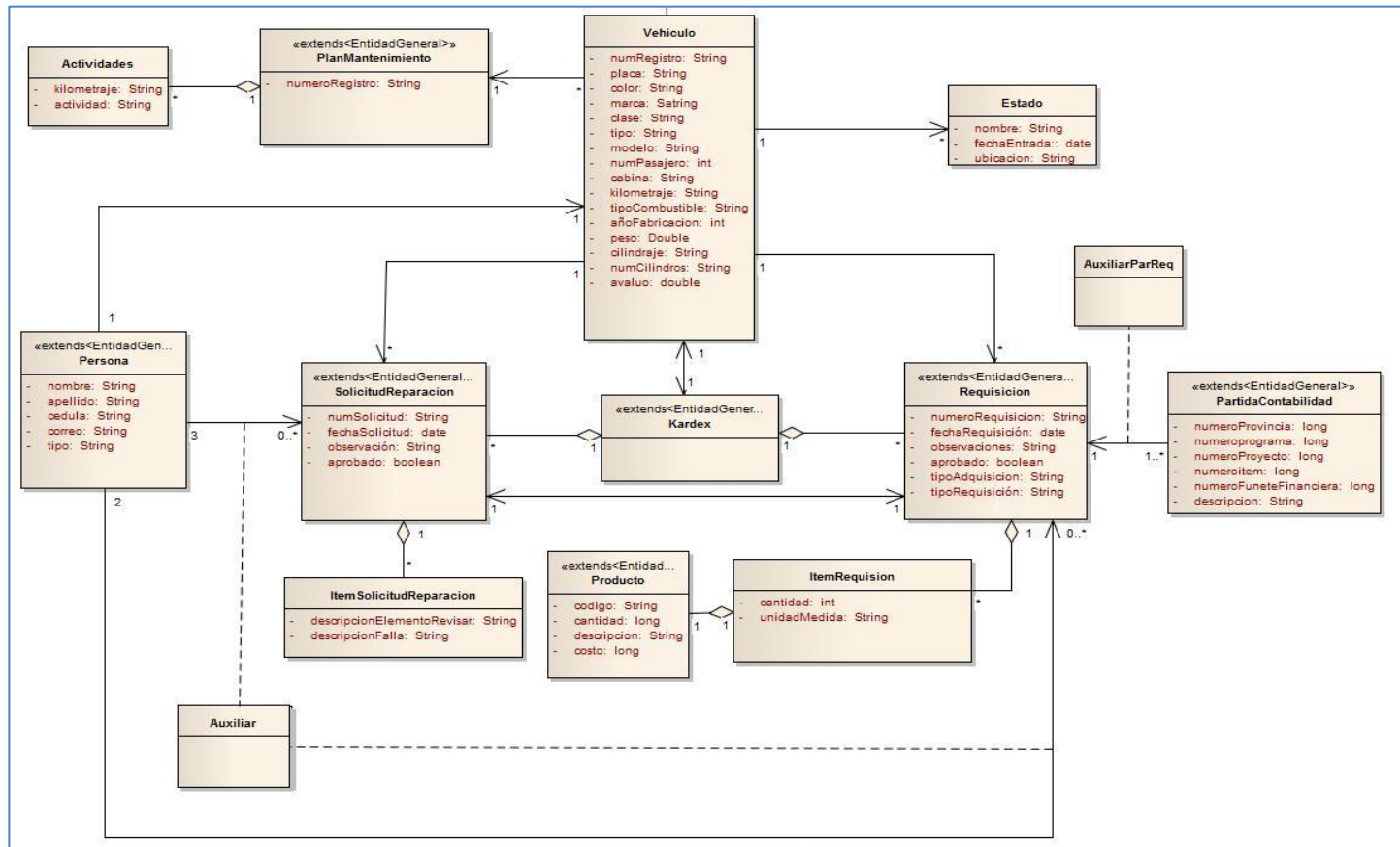


Figura 3. Modelo del dominio inicial (b)

- Diagrama de clases inicial.

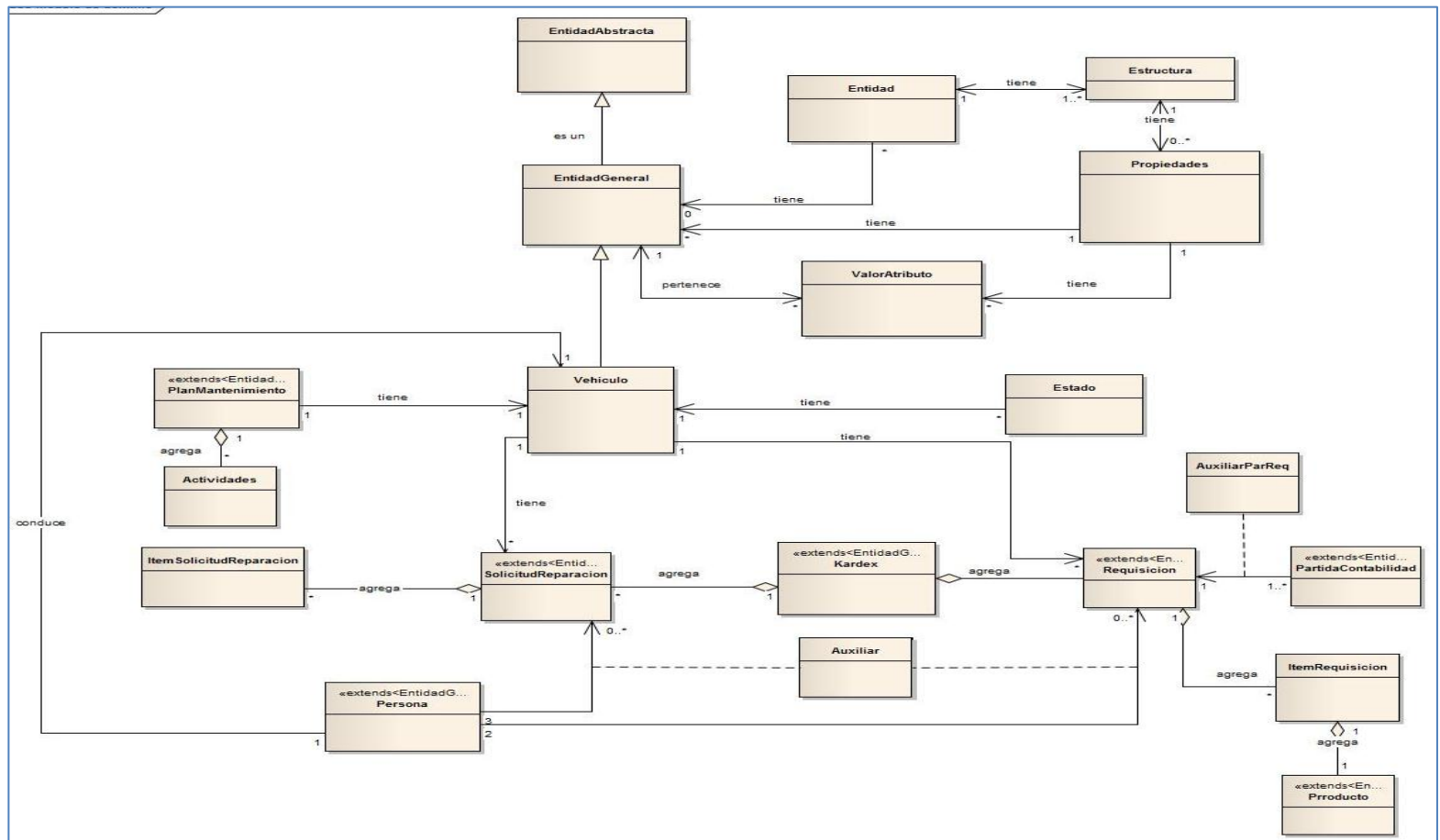


Figura 4. Diagrama de clases inicial

1.3 Modelado de casos de uso

Para el modelado de los casos de uso, se ha identificado los actores que intervienen en el sistema, a continuación se listan los casos de uso que se obtuvieron mediante la especificación de requerimientos según el estándar IEEE 830 (ver anexo 3).

- Identificación de actores

TABLA IX.
LISTA ACTORES DEL SISTEMA

ACTOR	ROL
Administrador del Sistema	<ul style="list-style-type: none">• Crear, editar, buscar vehículos.• Crear, editar, buscar personal.• Gestionar usuarios.• Registrar kardex, asignar solicitud de reparación y mantenimiento a kardex, asignar requisición a kardex.• Generar: informe mensual de estado de los vehículos, resumen de estado general por vehículo.• Registrar estado de ubicación actual y estado mecánico del vehículo.• Generar reporte de usuarios y personal.
Secretario	<ul style="list-style-type: none">• Crear solicitud de reparación y mantenimiento, asignar vehículo a solicitud, asignar persona que solicita la reparación, crear ítems a solicitud de reparación y mantenimiento.• Crear requisición, asignar vehículo

	<p>a requisición, asignar persona que solicita la requisición, crear ítems a requisición.</p> <ul style="list-style-type: none"> • Asignar solicitud a una requisición. • Asigna requisición a una solicitud de reparación y mantenimiento. • Gestionar partidas de contabilidad
--	---

- Identificación de Casos de Uso

TABLA X.
LISTA DE CASOS DE USOS

Caso de Uso	Actor	Ref. Requerimiento	Código
Gestionar vehículos.	Administrador.	RF001, RF004, RF017.	CU001.
Registrar estado mecánico del vehículo.	Administrador.	RF002.	CU002.
Registrar estado de ubicación del vehículo.	Administrador.	RF003.	CU003.
Gestionar solicitudes de reparación y mantenimiento.	Secretario.	RF005.	CU004.
Agregar ítem de solicitud de reparación y mantenimiento.	Secretario.	RF005.	CU005.
Visualizar	Secretario.	RF005.	CU006.

solicitudes de reparación y mantenimiento aprobadas			
Asignar vehículo a una solicitud de reparación y mantenimiento.	Secretario.	RF005.	CU007.
Asignar requisición.	Secretario.	RF006.	CU008.
Gestionar requisición.	Secretario.	RF007.	CU009.
Visualizar requisiciones aprobadas.	Secretario.	RF007.	CU010.
Asignar vehículo a requisición.	Secretario.	RF007.	CU011.
Agregar ítem de requisición.	Secretario.	RF007.	CU012.
Asignar solicitud reparación y mantenimiento.	Secretario.	RF008.	CU013.
Asignar producto.	Secretario.	RF009.	CU014.
Gestionar partida de contabilidad.	Secretario.	RF010.	CU015.
Gestionar productos.	Secretario.	RF011.	CU016.
Registrar kardex.	Administrador.	RF012.	CU017.
Aprobar solicitud reparación y	Administrador.	RF013.	CU018.

mantenimiento.			
Aprobar requisición.	Administrador.	RF014.	CU019.
Gestionar plan de mantenimiento.	Administrador.	RF015.	CU020.
Registrar mantenimiento del vehículo.	Administrador.	RF016.	CU021.
Gestionar personal.	Administrador.	RF018.	CU022.
Gestionar funcionalidades.	Administrador.	RF019.	CU023.
Ingresar al sistema.	Administrador/ Secretario.	RF020.	CU024.

- Diagrama de casos de uso

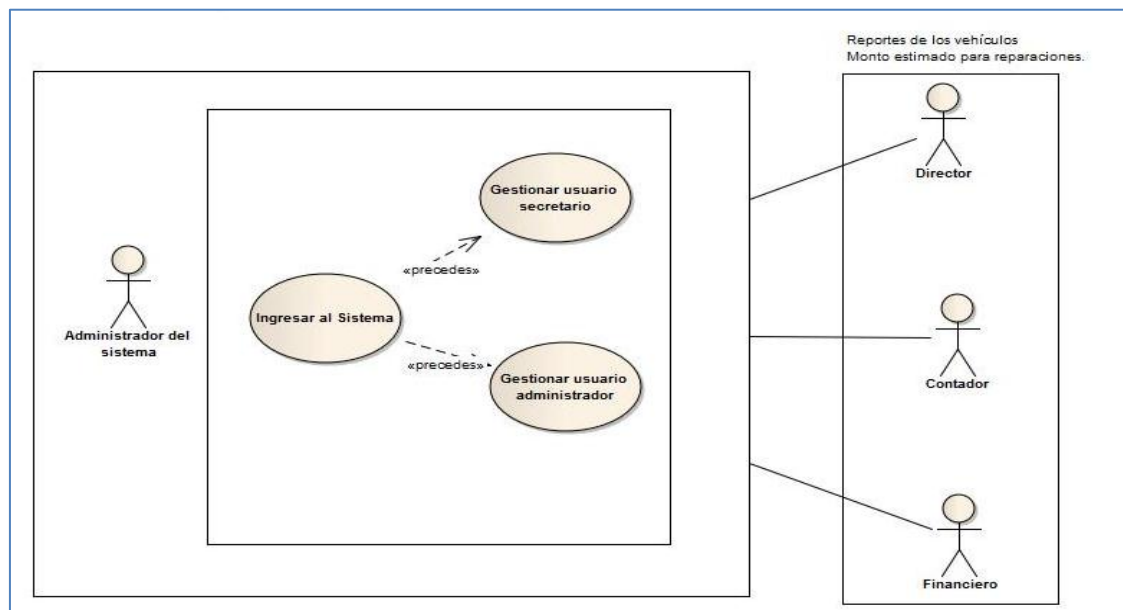


Figura 5. Diagrama de casos de uso general

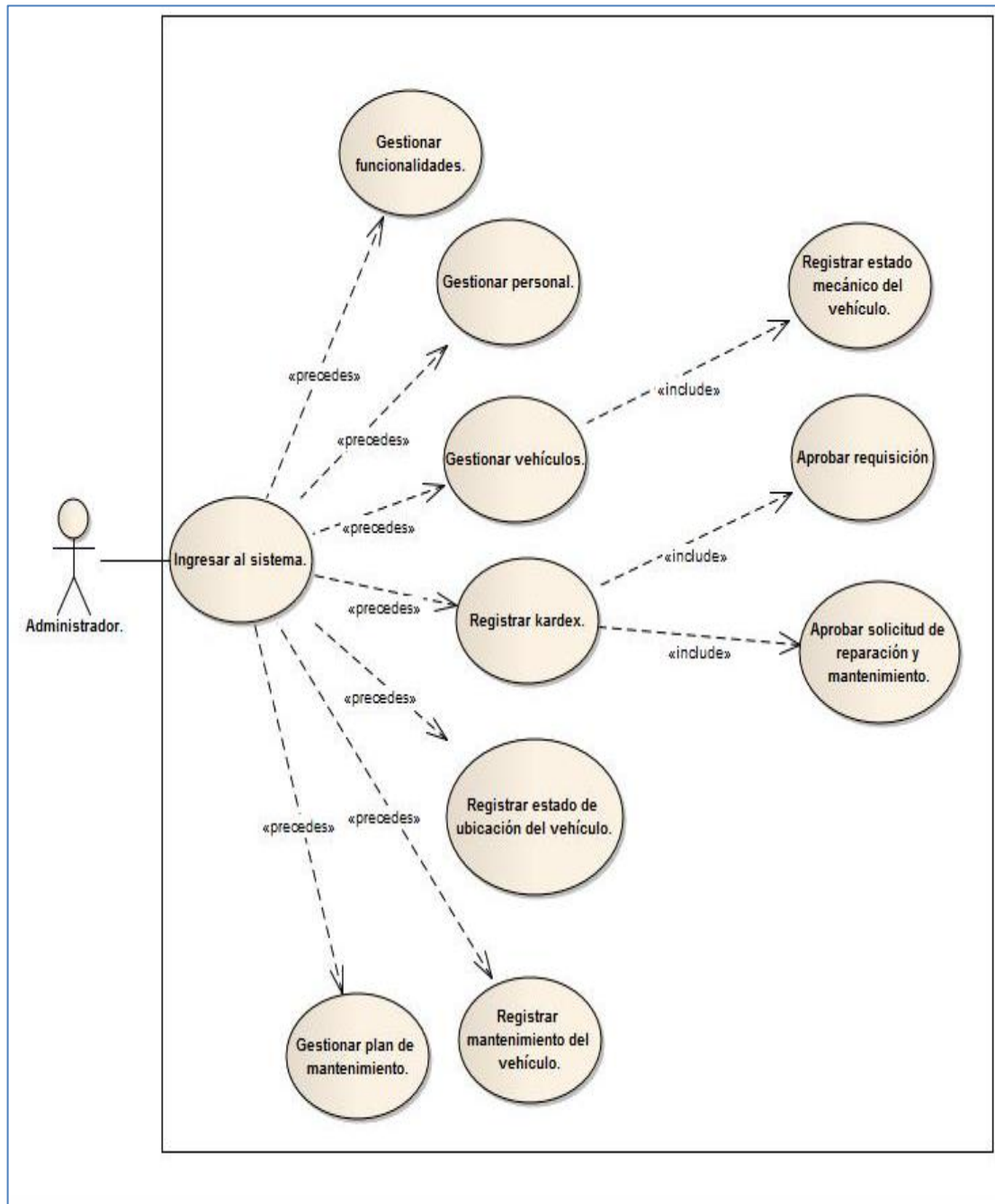


Figura 6. Diagrama de casos de uso del administrador

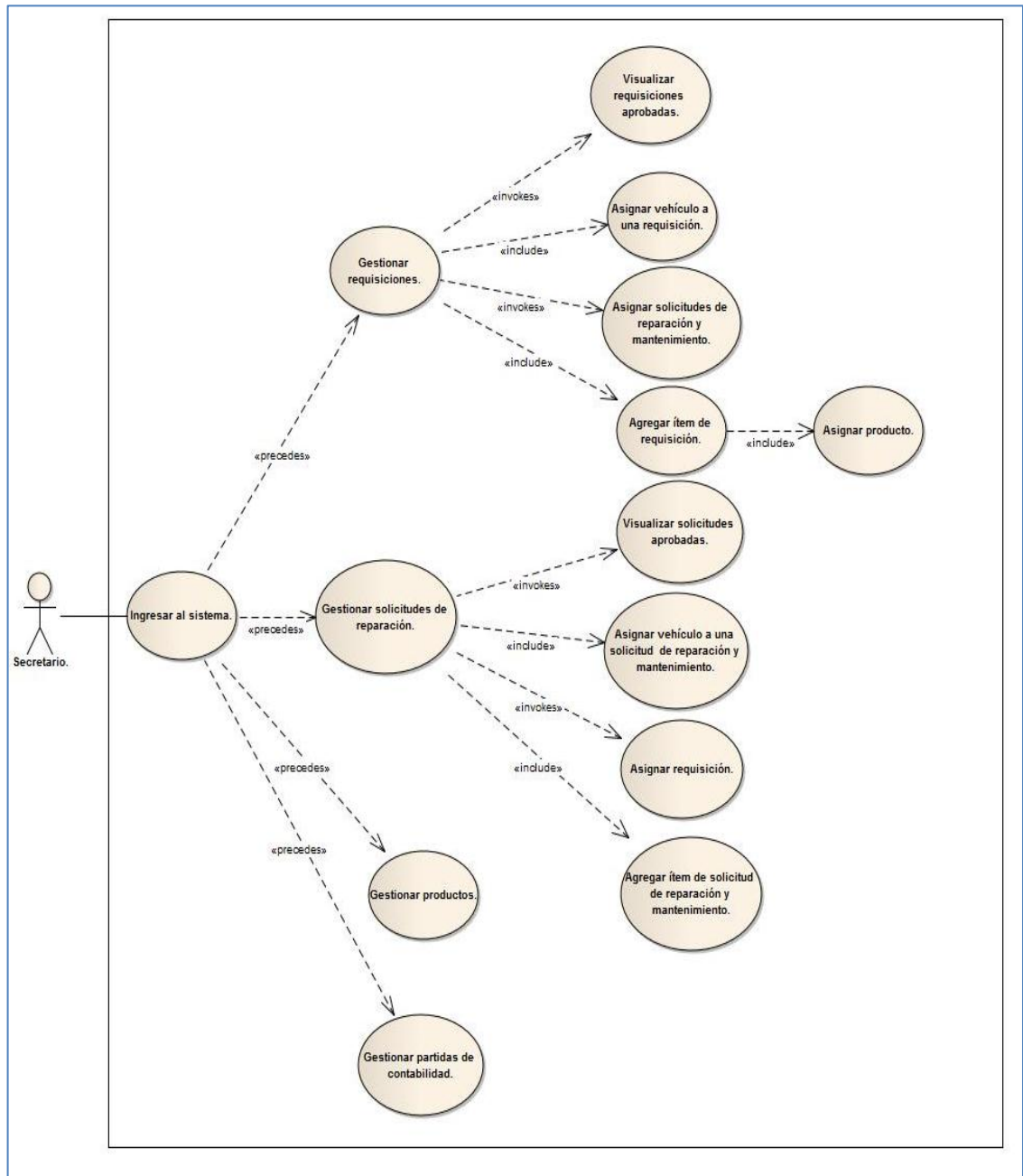


Figura 7. Diagrama de casos de uso del secretario

1.4. Prototipo de la interfaz gráfica.

Para un mayor entendimiento de los casos de uso, Iconix sugiere realizar el prototipo de interfaz de usuario, en esta parte se ha considerado indicar las pantallas de los siguientes casos de uso: gestionar funcionalidades, registrar mantenimiento del vehículo, registrar

estado mecánico del vehículo, registrar estado de ubicación y gestionar kardex. El resto del prototipo puede visualizarlos (ver anexo 4).

- Gestionar funcionalidades.

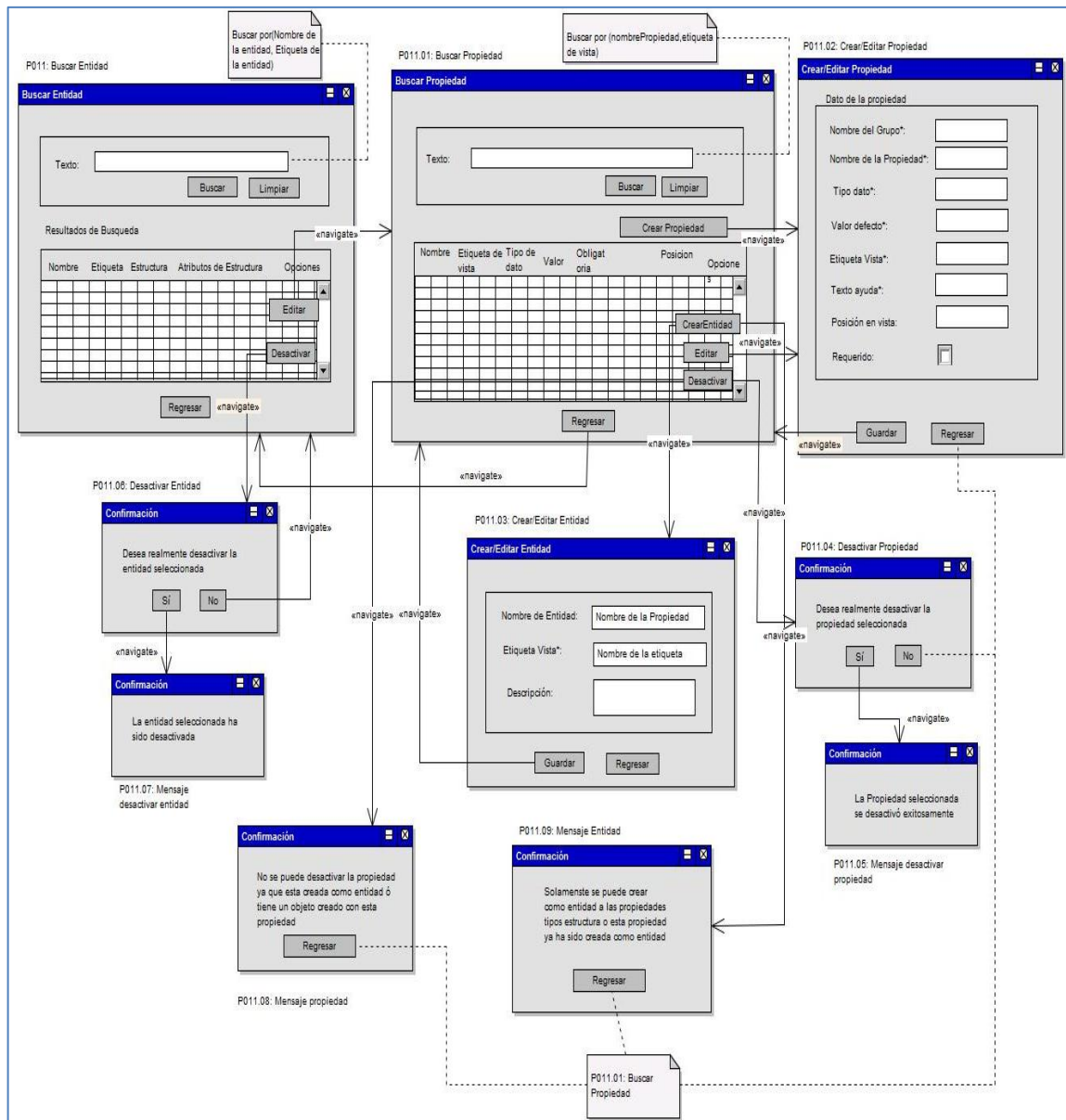


Figura 8. Prototipo del CU023

- Registrar mantenimiento del vehículo.

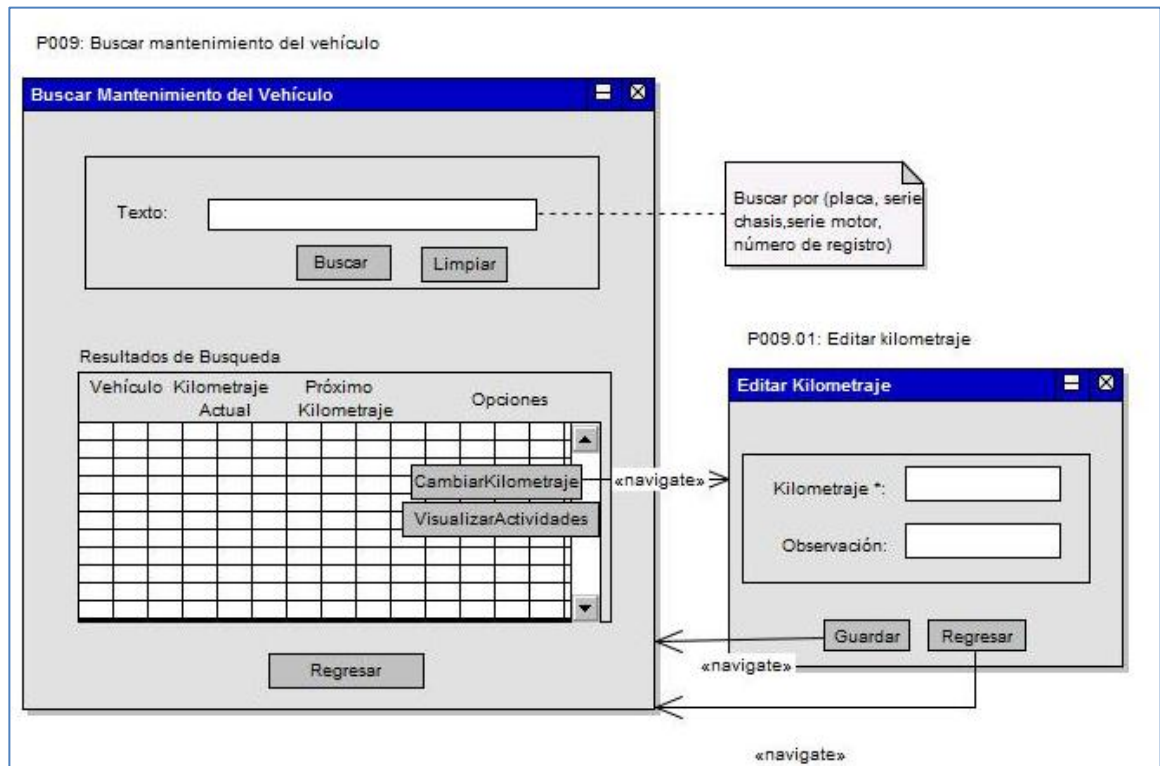


Figura 9. Prototipo del CU021

- Registrar estado de ubicación del vehículo.

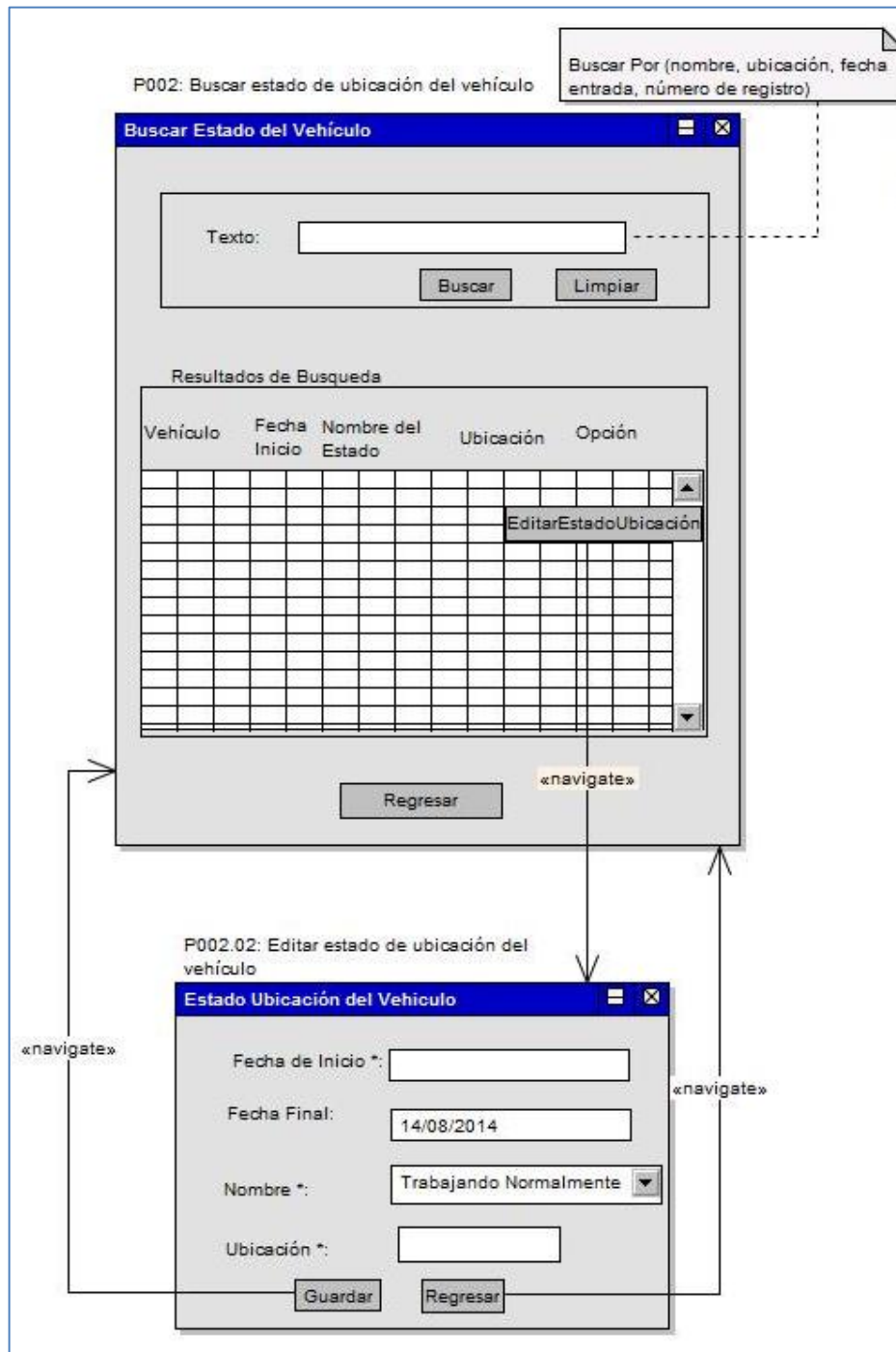


Figura 10. Prototipo del CU002.

- Registrar estado mecánico del vehículo.

P002.01: Editar Estado Mecánico

Editar Estado Mecánico

ESPECIFICACIONES TÉCNICAS

Registro: <input type="text"/>	Tipo: <input type="text"/>	Nro. Chasis: <input type="text"/>
Placa: <input type="text"/>	Marca: <input type="text"/>	Nro. Motor: <input type="text"/>
Modelo: <input type="text"/>	Color: <input type="text"/>	Tipo: <input type="text"/>

ESPECIFICACIONES MECÁNICAS

MOTOR

Sistema Combustible: <input type="text"/>	Sistema Refrigeración: <input type="text"/>
Potencia Motor: <input type="text"/>	Sistema Admisión y escape: <input type="text"/>
Sistema Lubricación: <input type="text"/>	Instrumentos de Control: <input type="text"/>

TRANSMISIÓN

Embrague:

Caja de cambios:

Diferenciales:

SUSPENSIÓN

Amortiguadores:

Eje delantero:

Eje posterior:

FRENOS

Cilindro principal:

Cilindro secundario:

Forros de freno:

SISTEMA ELÉCTRICO

Baterías:

Arranque:

Alternador:

NEUMÁTICO

Llantas:

CARROCERÍA

Forma:

DIRECCIÓN

Caja de Dirección: <input type="text"/>	Bomba dirección: <input type="text"/>
Articulaciones y terminales: <input type="text"/>	
Pines bodines y rotulas: <input type="text"/>	

CARROCERÍA

Forma:

Vizualizar Estado Mecánico

Estado Mecánico General:

RESUMEN

Guardar

Regresar

P001: Buscar Estado Mecánico del Vehículo

Figura 11. Prototipo del CU003.

- Registrar kardex.

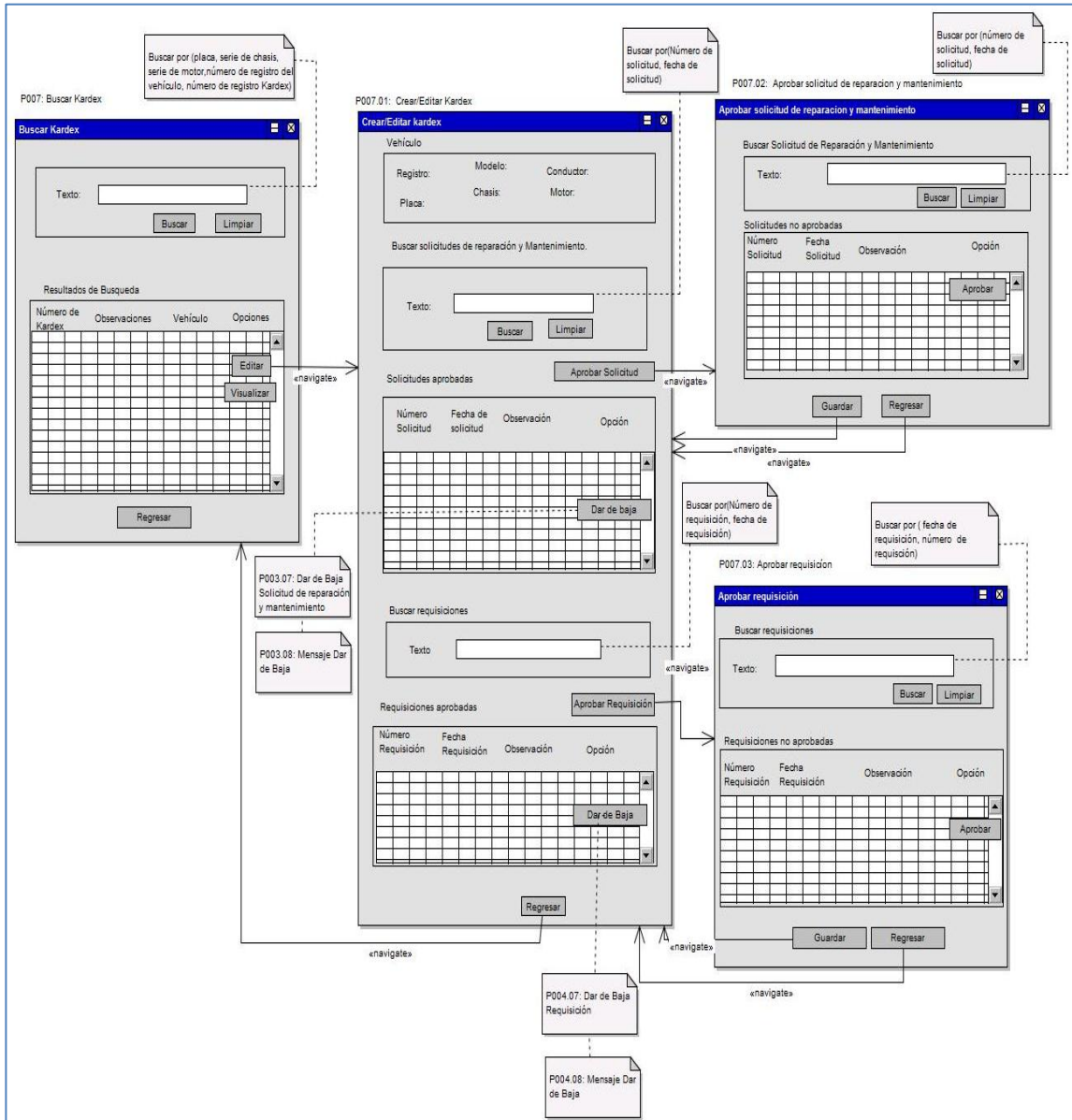


Figura 12. Prototipo del CU017

1.5 Diagrama de Paquetes

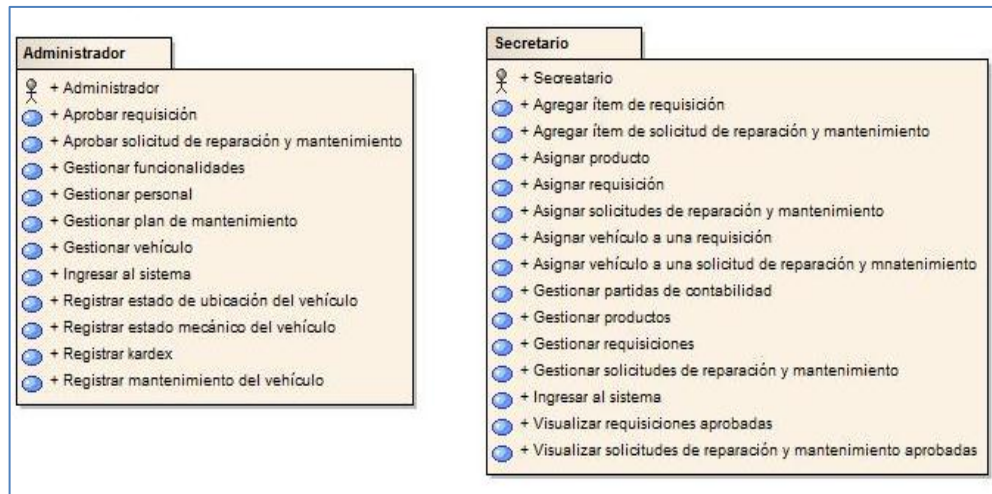


Figura 13. Diagrama de paquetes

2. Análisis y Diseño preliminar.

En esta fase de la metodología Iconix se debe realizar la descripción de casos de uso, ya que permite identificar las acciones que va hacer el sistema, así como la construcción de diagramas de robustez que facilita el reconocimiento e identificación de objetos que participan en cada caso de uso. A continuación se detalla cada una de las actividades realizadas en esta fase.

2.1 Descripción de los casos de uso.

En esta actividad se consideró describir algunos casos de uso como: gestionar funcionalidades, registrar mantenimiento del vehículo, registrar estado mecánico del vehículo, registrar estado de ubicación y gestionar kardex. Ya que la especificación de requerimientos según el estándar IEEE 830 (ver anexo 3) exige la descripción de cada uno de los casos de uso.

- Gestionar funcionalidades.

TABLA XI.
DESCRIPCIÓN DEL CASO DE USO GESTIONAR FUNCIONALIDAD.

Caso de Uso:	Gestionar funcionalidades.			Actor:	Administrador.
Código:	CU023.	Tipo:	Primario-Sistema.	Pantalla:	P011, P011.01, P011.02, P011.03, P011.04, P011.05, P011.06, P011.07, P011.08, P011.09.
Requerimientos:	RF019.				
Resumen:	El Administrador, visualiza las entidades para crear y visualiza las propiedades para poder desactivar, editar o crear en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar entidad y propiedad. • Crear una entidad. • Crear y editar una propiedad. 				
Precondiciones:	<ul style="list-style-type: none"> • El administrador se haya autenticado. 				
PosCondiciones:	<ul style="list-style-type: none"> • Entidad creada y propiedad creada o editada. 				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El Administrador ingresa los criterios de búsqueda (nombre de la entidad, etiqueta de la entidad) para una entidad y presiona el botón [Buscar] en la pantalla Buscar Entidad. 2. El sistema actualiza la tabla [Resultados de búsqueda], con la entidad que 					

<p>coincidan con el criterio de búsqueda.</p> <p>3. El Administrador selecciona la entidad de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla Buscar Entidad.</p> <p>4. El sistema invoca a la pantalla Buscar Propiedad.</p> <p>5. El Administrador ingresa los criterios de búsqueda (nombre de la propiedad, etiqueta de vista) para una propiedad y presiona el botón [Buscar] en la pantalla Buscar Propiedad.</p> <p>6. El sistema actualiza la tabla [Resultados de búsqueda], con la propiedad que coincidan con el criterio de búsqueda.</p> <p>7. El Administrador verifica que exista la propiedad en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear Propiedad] de la pantalla Buscar Propiedad.</p> <p>8. El sistema invoca a la pantalla Crear/Editar Propiedad.</p> <p>9. El Administrador registra los datos generales (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) de la propiedad y presiona el botón [Guardar] de la pantalla Crear/Editar Propiedad.</p> <p>10. Si el Administrador crear una propiedad de tipo de datos entidad, el sistema presenta en la pantalla Buscar Propiedad el botón [Crear Entidad].</p> <p>11. El caso de uso finaliza.</p>
Curso alterno de eventos.
<p>A. Limpiar búsqueda entidad.</p> <p>A.1. El Administrador presiona el botón [Limpiar] de la pantalla Buscar Entidad.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar Entidad.</p>
<p>B. Limpiar búsqueda propiedad.</p> <p>B.1. El Administrador presiona el botón [Limpiar] de la pantalla Buscar Propiedad.</p> <p>B.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar Propiedad.</p>
<p>C. Editar propiedad.</p> <p>C.7. El Administrador selecciona la propiedad a modificar de la tabla [Resultados</p>

de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Propiedad.**

C.8. El sistema recupera la información de la propiedad y la muestra en la pantalla **Crear/Editar Propiedad.**

C.9. El Administrador edita los datos correspondientes (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) de la propiedad y presiona el botón [Guardar] de la pantalla **Crear/Editar Propiedad.**

C.10. El sistema guarda la nueva propiedad con su respectiva información y presenta la pantalla **Buscar Propiedad.**

D. Crear entidad.

D.7. El administrador elige la propiedad a crear como entidad de la tabla [Resultados de Búsqueda] y presiona el botón [Crear entidad], en la pantalla **Buscar propiedad**

D.8. El sistema invoca la pantalla **Crear/Editar Entidad.**

D.9. El Administrador registra los datos correspondientes (descripción) de la Entidad y presiona el botón [Guardar] de la pantalla **Crear/Editar Entidad.**

D.10. El sistema guarda la entidad con su respectiva información y presenta la pantalla **Buscar Propiedad** para agregar propiedades.

E. Desactivar propiedad.

E.7. El administrador elige la propiedad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Propiedad.**

E.4. El sistema invoca pantalla **Desactivar propiedad.**

E.5. El administrador presiona el botón [Sí], de la pantalla **Desactivar propiedad.**

E.6. El sistema presenta un mensaje “La propiedad seleccionada ha sido desactivada exitosamente”.

F. Desactivar entidad.

F.3. El administrador elige la entidad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Entidad.**

F.4. El sistema invoca la pantalla **Desactivar entidad.**

F.5. El administrador presiona el botón [Sí], de la pantalla **Desactivar entidad.**

F.6. El sistema presenta un mensaje “La entidad seleccionada ha sido

desactivada”.
<p>G. Campos obligatorios y vacíos de propiedad.</p> <p>G.10.- (G.C.10) El sistema verifica que estén llenos los campos obligatorios (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla Crear/Editar Propiedad y continúa (9, C.9).</p>
<p>H. No puede crear entidad</p> <p>H.D.8. El sistema verifica que la propiedad escogida sea de tipo estructura y no haya sido creada como entidad, si no es así presenta el mensaje “Solamente se puede crear como entidad a las propiedad tipos estructura o esta propiedad ya ha sido creada como entidad”.</p>

- Registrar mantenimiento del vehículo.

TABLA XII.

DESCRIPCIÓN DEL CASO DE USO REGISTRAR MANTENIMIENTO DEL VEHÍCULO.

Caso de Uso:	Registrar mantenimiento del vehículo.			Actor:	Administrador.
Código:	CU021	Tipo:	Primario-Sistema.	Pantalla:	P009, P009.01.
Requerimientos:	RF016.				
Resumen:	El Administrador, visualiza el mantenimiento necesario por cada uno de los vehículos, para poder e Celular: 0985537193ditarlo.				
Objetivos:	<ul style="list-style-type: none"> • Buscar vehículos. • Editar el mantenimiento del vehículo y visualizar las actividades del mantenimiento a realizar según el kilometraje del vehículo. 				
Precondiciones:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. 				

PosCondiciones:	<ul style="list-style-type: none"> • kilometraje del vehículo actualizado.
Curso normal de eventos.	
<ol style="list-style-type: none"> 1. El Administrador del sistema ingresa los criterios de búsqueda (placa del vehículo, número de registro del vehículo, serie de motor, serie de chasis) para un vehículo y presiona el botón [Buscar] en la pantalla Buscar Mantenimiento del Vehículo. 2. El sistema actualiza la tabla [Resultados de Búsqueda] con los vehículos que coincidan con el criterio de búsqueda. 3. El administrador elige el vehículo a modificar su kilometraje de la tabla [Resultados de Búsqueda] y presiona el botón [Cambiar Kilometraje] de la pantalla Buscar Mantenimiento del Vehículo. 4. El sistema invoca la pantalla Editar Kilometraje. 5. El administrador ingresa el kilometraje actual con su respectiva observación del vehículo y presiona el botón [Guardar] de la pantalla Editar Kilometraje. 6. El sistema guarda el nuevo kilometraje del vehículo con su respectiva observación y presenta la pantalla Buscar Mantenimiento del Vehículo. 7. El caso de uso finaliza. 	
Curso alterno de eventos.	
A. Limpiar búsqueda.	
<ol style="list-style-type: none"> A.1. El Administrador presiona el botón [Limpiar] de la pantalla Buscar Mantenimiento del Vehículo. A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar Mantenimiento del Vehículo. 	
B. Campos obligatorios vacíos.	
<ol style="list-style-type: none"> B.6. El sistema verifica que esté lleno el campo obligatorio (kilometraje y observación) y presenta un mensaje “Campos obligatorios vacío” en la pantalla Crear/Editar Kilometraje, y continúa (5). 	
C. Visualizar actividades.	
<ol style="list-style-type: none"> C.3. El administrador presiona el botón [Visualizar Actividades] de la pantalla Buscar Mantenimiento del Vehículo. 	

C.4 El sistema muestra las próximas actividades del mantenimiento a realizar del vehículo seleccionado.

- Registrar estado mecánico del vehículo.

TABLA XIII.
DESCRIPCIÓN DEL CASO DE USO REGISTRAR ESTADO MECÁNICO DEL VEHÍCULO

Caso de Uso:	Registrar estado mecánico del vehículo.		Actor:	Administrador
Código:	CU00 2	Tipo:	Primario - Sistema	Pantallas: P001, P002.01.
Ref. Req:	RF002.			
Resumen:	El Administrador, podrá editar, evaluar y visualizar el estado mecánico del vehículo.			
Objetivos:	<ul style="list-style-type: none"> • Mantener actualizado el estado mecánico del vehículo. 			
PreCondiciones:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. 			
PosCondiciones:	<ul style="list-style-type: none"> • Estado mecánico del vehículo actualizado y evaluado. 			
Curso normal de eventos				
<ol style="list-style-type: none"> 1. El sistema recupera el estado actual del vehículo y lo presenta en la pantalla Editar Estado Mecánico. 2. El administrador cambia los estados de los vehículos (Motor: sistema combustible, potencia motor, sistema lubricación, sistema refrigeración, sistema admisión escape, instrumentos de control. Transmisión: embrague, caja de cambios, diferenciales. Suspensión: amortiguadores, eje delantero, eje posterior. Frenos: cilindro principal, cilindro secundario, forros de freno. 				

Sistema Eléctrico: baterías, arranque, alternador. Neumático: llantas. Carrocería: forma, Dirección: caja de dirección, articulaciones y terminales, pines bocines y rótulas) y presiona el botón [Guardar] de la pantalla **Editar Estado Mecánico**.

3. El sistema evalúa el estado general del vehículo en base a cada uno de los estados de las partes mecánicas y lo presenta en la caja de texto [Estado general] de la pantalla **Editar Estado Mecánico**.

4. El caso de uso finaliza.

Curso alterno de eventos

A. Visualizar estado mecánico

A.1. El administrador presiona el botón [Visualizar estado mecánico], de la pantalla **Editar Estado Mecánico**.

A.2. EL sistema presenta la información del estado mecánico del vehículo con su evaluación.

- Registrar estado de ubicación del vehículo.

TABLA XIV.

DESCRIPCIÓN DEL CASO DE USO REGISTRAR ESTADO DE UBICACIÓN DEL VEHÍCULO

Caso de Uso:	Gestionar estado de ubicación del vehículo.		Actor:	Administrador
Código:	CU003	Tipo:	Primario-Sistema	Pantallas: P002, P002.02.
Ref. Req:	RF003.			
Resumen:	El Administrador, podrá editar el estado de ubicación del vehículo.			
Objetivos:	<ul style="list-style-type: none"> • Mantener actualizado el estado de ubicación del vehículo. 			

Pre- Condiciones:	<ul style="list-style-type: none"> • El administrador se haya autenticado.
PosCondicion es:	<ul style="list-style-type: none"> • Estado de ubicación actualizado.
Curso normal de eventos	
<ol style="list-style-type: none"> 1. El Administrador del sistema ingresa los criterios de búsqueda (nombre, ubicación, fecha entrada, número de registro) para un vehículo y presiona el botón [Buscar] en la pantalla Buscar estado de ubicación del vehículo. 2. El sistema actualiza la tabla [Resultados de Búsqueda] con los vehículos que coincidan con el criterio de búsqueda. 3. El Administrador verifica que exista el vehículo en la tabla [Resultados de búsqueda] y presiona el botón [Editar Estado Ubicación] de la pantalla Buscar Estado del Vehículo. 4. El sistema recupera el estado actual de ubicación del vehículo y lo presenta en la pantalla Editar Estado Ubicación del Vehículo. 5. El administrador edita el estado de ubicación del vehículo (nombre, fecha entrada, ubicación) y presiona el botón [Guardar] de la pantalla Editar Estado Ubicación del Vehículo. 6. El sistema guarda el estado de ubicación actual del vehículo y presenta la pantalla Buscar Estado del Vehículo. 	
Curso alterno de eventos	
A. Limpiar Búsqueda.	
<ol style="list-style-type: none"> A.1. El Administrador presiona el botón [Limpiar] de la pantalla Buscar Estado del Vehículo. A.2. El sistema limpia el criterio de búsqueda y actualiza en la tabla [Resultados de búsqueda] de la pantalla Buscar Estado de Ubicación del Vehículo. 	
B. Campos Obligatorios Vacíos.	
<ol style="list-style-type: none"> B.4. El sistema verifica que estén llenos los campos obligatorios (nombre, fecha entrada, ubicación) y continúa en el paso (5). 	

- Registrar kardex.

TABLA XV.

DESCRIPCIÓN DEL CASO DE USO REGISTRAR KARDEX.

Caso de Uso:	Registrar Kardex			Actor:	Administrador.
Código:	CU017.	Tipo:	Primario-Sistema.	Pantalla:	P003.07, P003.08, P004.07, P004.08, P007, P007.01, P007.02, P007.03.
Requerimientos:	RF013.				
Resumen:	El Administrador, visualiza las Kardex, para poder editarlas.				
Objetivos:	<ul style="list-style-type: none"> • Buscar Kardex. • Editar Kardex. 				
Precondiciones:	El Administrador se haya autenticado.				
PosCondiciones:	Kardex por vehículo registrada.				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El Administrador ingresa los criterios de búsqueda (placa, serie de motor, serie de chasis, registro del vehículo y kardex) para una Kardex y presiona el botón [Buscar] en la pantalla Buscar Kardex. 2. El sistema actualiza la tabla [Resultados de búsqueda], con la kardex que coincidan con el criterio de búsqueda. 3. El Administrador verifica que exista la kardex en la tabla [Resultados de búsqueda] en la pantalla Buscar Kardex, selecciona la kardex a modificar y presiona el botón [Editar] de la pantalla Buscar Kardex. 4. El sistema recupera la información (vehículo, requisición solicitud de reparación y mantenimiento aprobadas) de la kardex y los muestra en la pantalla Crear/Editar Kardex. 					

5. El Administrador presiona el botón [Aprobar Solicitud] e invoca al caso de uso CU018 o presiona el botón [Aprobar Requisición] e invoca al caso de uso CU019 y presiona el botón [Guardar] de la pantalla **Crear/Editar Kardex**.
6. El caso de uso finaliza.

Curso alterno de eventos.

1. Limpiar Búsqueda.

- A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar kardex**.
- A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar kardex**.

2. Buscar solicitudes de reparación y mantenimiento aprobadas.

- E.5. El administrador ingresa los criterios de búsqueda (número de solicitud, fecha de solicitud) presiona el botón [**Buscar**], de la pantalla **Crear/Editar Kardex**.
- E.6. El sistema actualiza la tabla [Solicitudes aprobadas], con la solicitud de reparación y mantenimiento que coincidan con el criterio de búsqueda.
- E.7. El Administrador verifica que exista la solicitud de reparación y mantenimiento en la tabla [Solicitudes aprobadas] en la pantalla **Crear/Editar Kardex** y continúa (5)

3. Buscar requisiciones aprobadas.

- C.5. El administrador ingresa los criterios de búsqueda (número de requisición, fecha de requisición) presiona el botón [**Buscar**], de la pantalla **Crear/Editar Kardex**.
- C.6. El sistema actualiza la tabla [Requisiciones aprobadas], con la requisición que coincidan con el criterio de búsqueda.
- C.7. El Administrador verifica que exista la requisición en la tabla [Requisiciones aprobadas] en la pantalla **Crear/Editar Kardex** y continúa (5)

4. Dar de Baja a una solicitud

- D.E.7. El administrador selecciona la solicitud de reparación a dar de baja de la tabla [Solicitudes aprobadas] y presiona el botón [Dar de baja] de la pantalla **Crear/Editar Kardex**.
- D.E.8 El sistema invoca la pantalla **Dar de baja Solicitud de Reparación y Mantenimiento**.
- D.E.9. El administrador presiona el botón [Sí] de la pantalla **Dar de baja Solicitud de Reparación y Mantenimiento**.

D.E.10. El sistema presenta un mensaje “Solicitud de reparación y mantenimiento seleccionada ha sido dada de baja”.

5. Dar de baja Requisición.

E.C.7. El administrador selecciona la requisición a dar de baja de la tabla [Requisiciones Aprobadas] y presiona el botón [dar de baja] de la pantalla

Crear/Editar Kardex.

E.C.8. El sistema invoca la pantalla **Dar de baja Requisición.**

E.C.9. El administrador presiona el botón [Sí], de la pantalla **Dar de baja Requisición.**

E.C.10. El sistema presenta un mensaje “Requisición seleccionada ha sido dado de baja”.

2.2 Realiza diagrama de robustez

Por el motivo de la extensa documentación de diagramas de robustez se consideró indicar los diagramas de los siguientes casos de uso: gestionar funcionalidades, registrar mantenimiento del vehículo, registrar estado mecánico del vehículo, registrar estado de ubicación y gestionar kardex. El resto de diagramas puede visualizarlos (ver anexo 5).

- Gestionar funcionalidades.

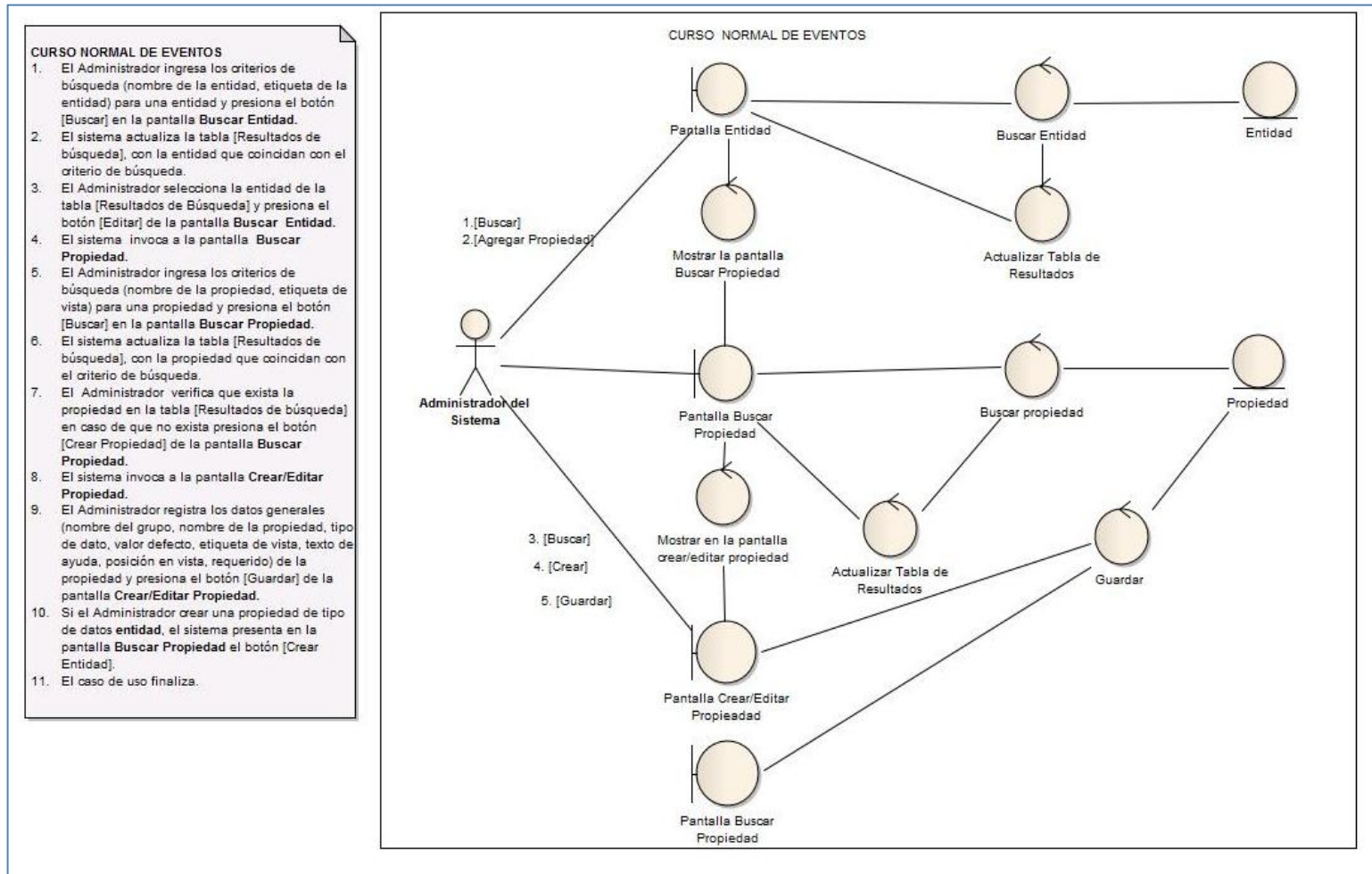


Figura 14. Diagrama de robustez CU023 curso normal eventos

CURSO ALTERNO DE EVENTOS

CURSO ALTERNO DE EVENTOS

A. Limpiar búsqueda entidad.
 A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Entidad**.
 A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Entidad**.

B. Limpiar búsqueda propiedad.
 B.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Propiedad**.
 B.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Propiedad**.

C. Editar propiedad.
 C.7. El Administrador selecciona la propiedad a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Propiedad**.
 C.8. El sistema recupera la información de la propiedad y la muestra en la pantalla **Crear/Editar Propiedad**.
 C.9. El Administrador edita los datos correspondientes (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) de la propiedad y presiona el botón [Guardar] de la pantalla **Crear/Editar Propiedad**.
 C.10. El sistema guarda la nueva propiedad con su respectiva información y presenta la pantalla **Buscar Propiedad**.
 D.7. El administrador elige la propiedad a crear como entidad de la tabla [Resultados de Búsqueda] y presiona el botón [Crear entidad], en la pantalla **Buscar propiedad**.
 D.8. El sistema invoca la pantalla **Crear/Editar Entidad**.
 D.9. El Administrador registra los datos correspondientes (descripción) de la Entidad y presiona el botón [Guardar] de la pantalla **Crear/Editar Entidad**.
 D.10. El sistema guarda la entidad con su respectiva información y presenta la pantalla **Buscar Propiedad** para agregar propiedades.

E. Desactivar propiedad.
 E.7. El administrador elige la propiedad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Propiedad**.
 E.4. El sistema invoca pantalla **Desactivar propiedad**.
 E.5. El administrador presiona el botón [Si], de la pantalla **Desactivar propiedad**.
 E.6. El sistema presenta un mensaje "La propiedad seleccionada ha sido desactivada exitosamente".

F. Desactivar entidad.
 F.3. El administrador elige la entidad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Entidad**.
 F.4. El sistema invoca la pantalla **Desactivar entidad**.
 F.5. El administrador presiona el botón [Si], de la pantalla **Desactivar entidad**.
 F.6. El sistema presenta un mensaje "La entidad seleccionada ha sido desactivada".

G. Campos obligatorios y vacíos de propiedad.
 G.10. (G.C.10) El sistema verifica que estén llenos los campos obligatorios (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Propiedad** y continúa (9, C.9).

H. No puede crear entidad
 H.9. El sistema verifica que la propiedad escogida sea de tipo estructura y no haya sido creada como entidad, si no es así presenta el mensaje "Solamente se puede crear como entidad a las propiedad tipos estructura o esta propiedad ya ha sido creada como entidad".

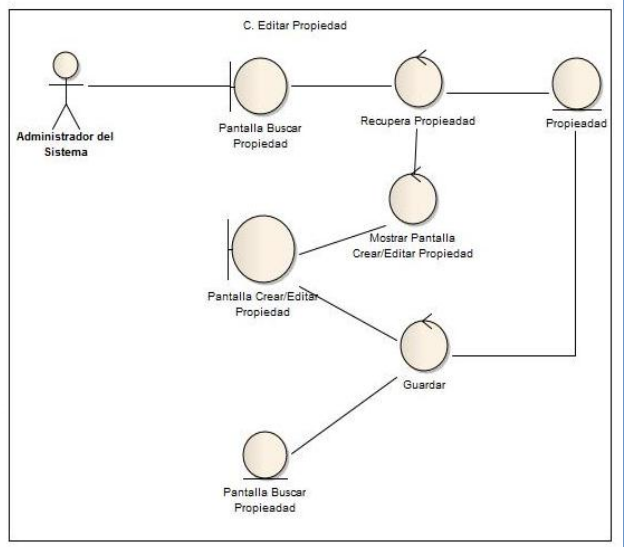
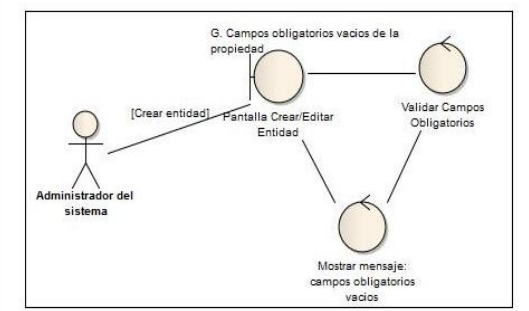
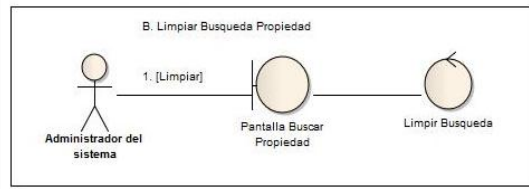
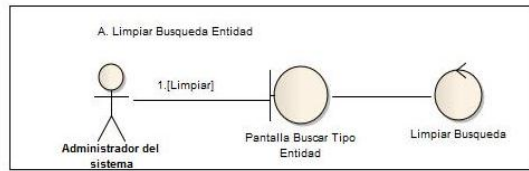


Figura 15. Diagrama de robustez CU023 curso alterno de eventos (a).

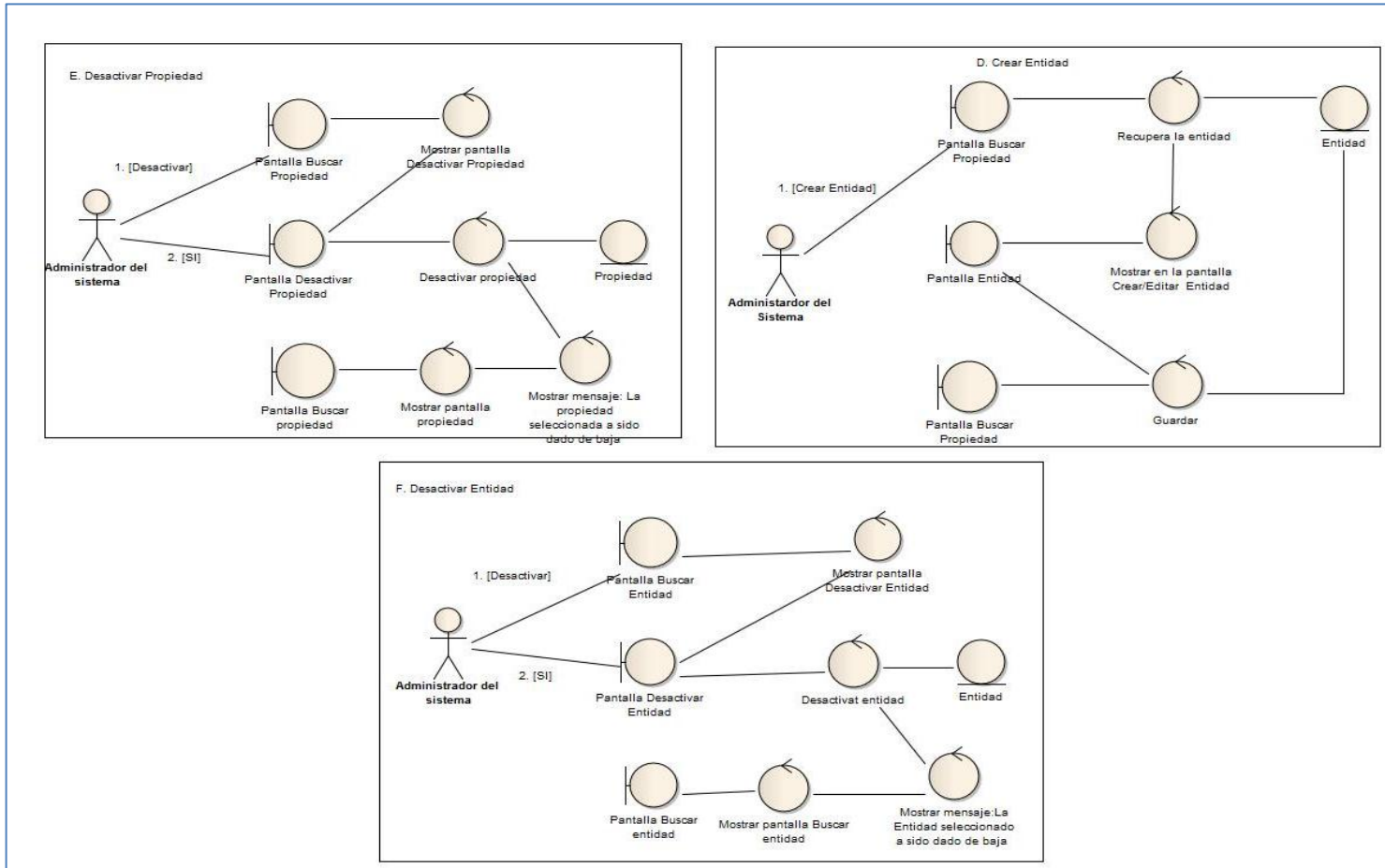


Figura 16. Diagrama de robustez CU023 curso alternativo de eventos (b).

- Registrar mantenimiento del vehículo.

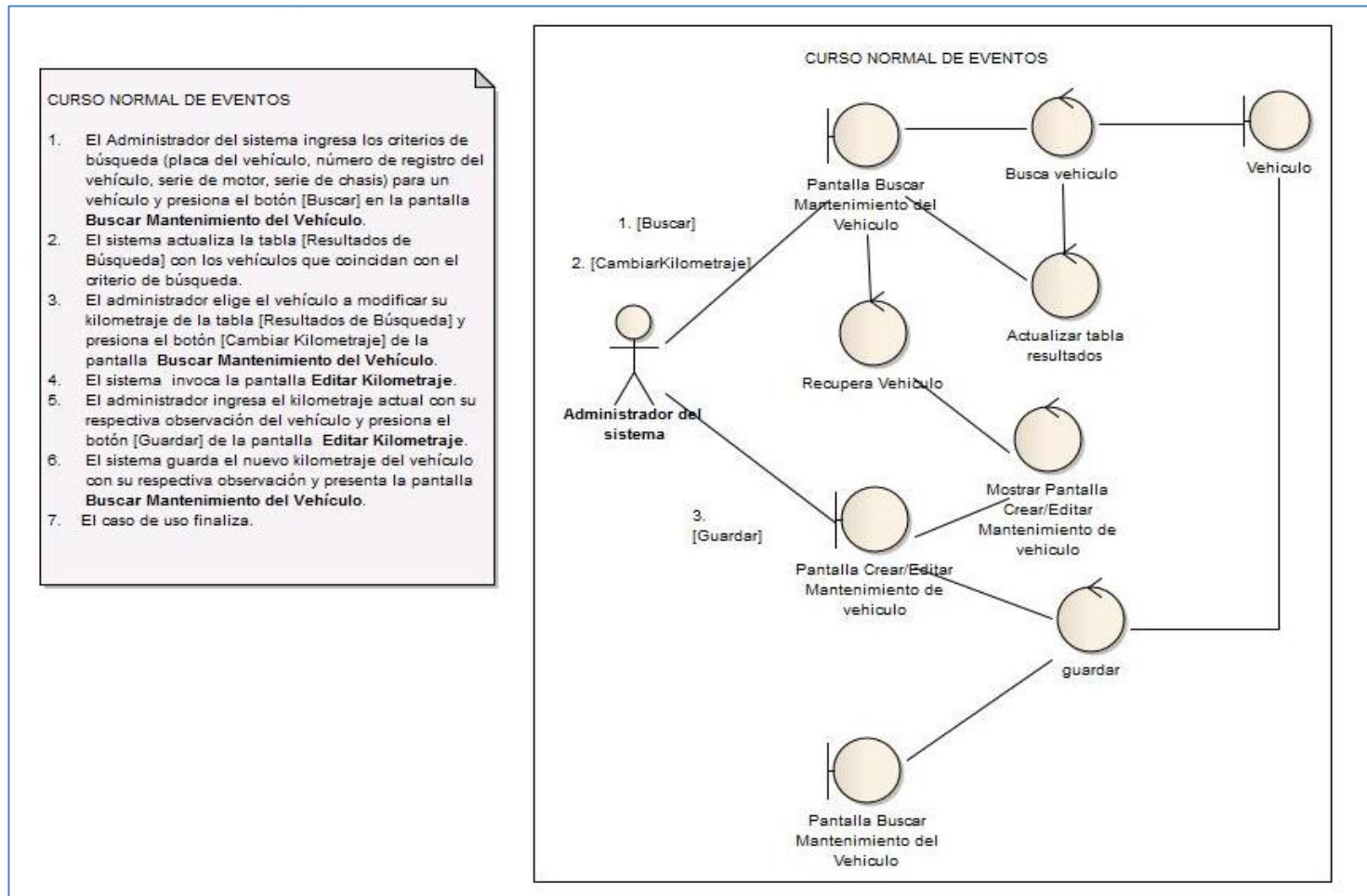


Figura 17. Diagrama de robustez CU021 curso normal de eventos.

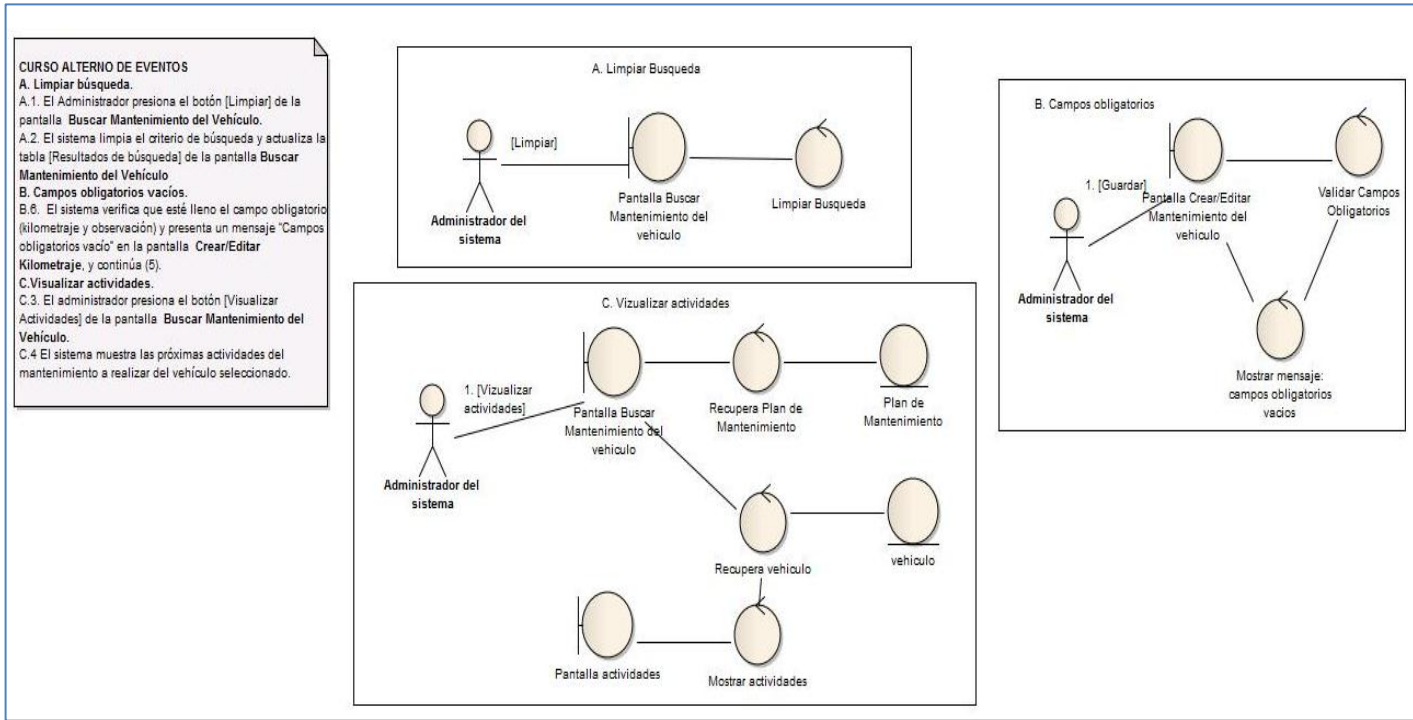


Figura 18. Diagrama de robustez CU021 curso alterno de eventos.

- Registrar estado mecánico del vehículo.

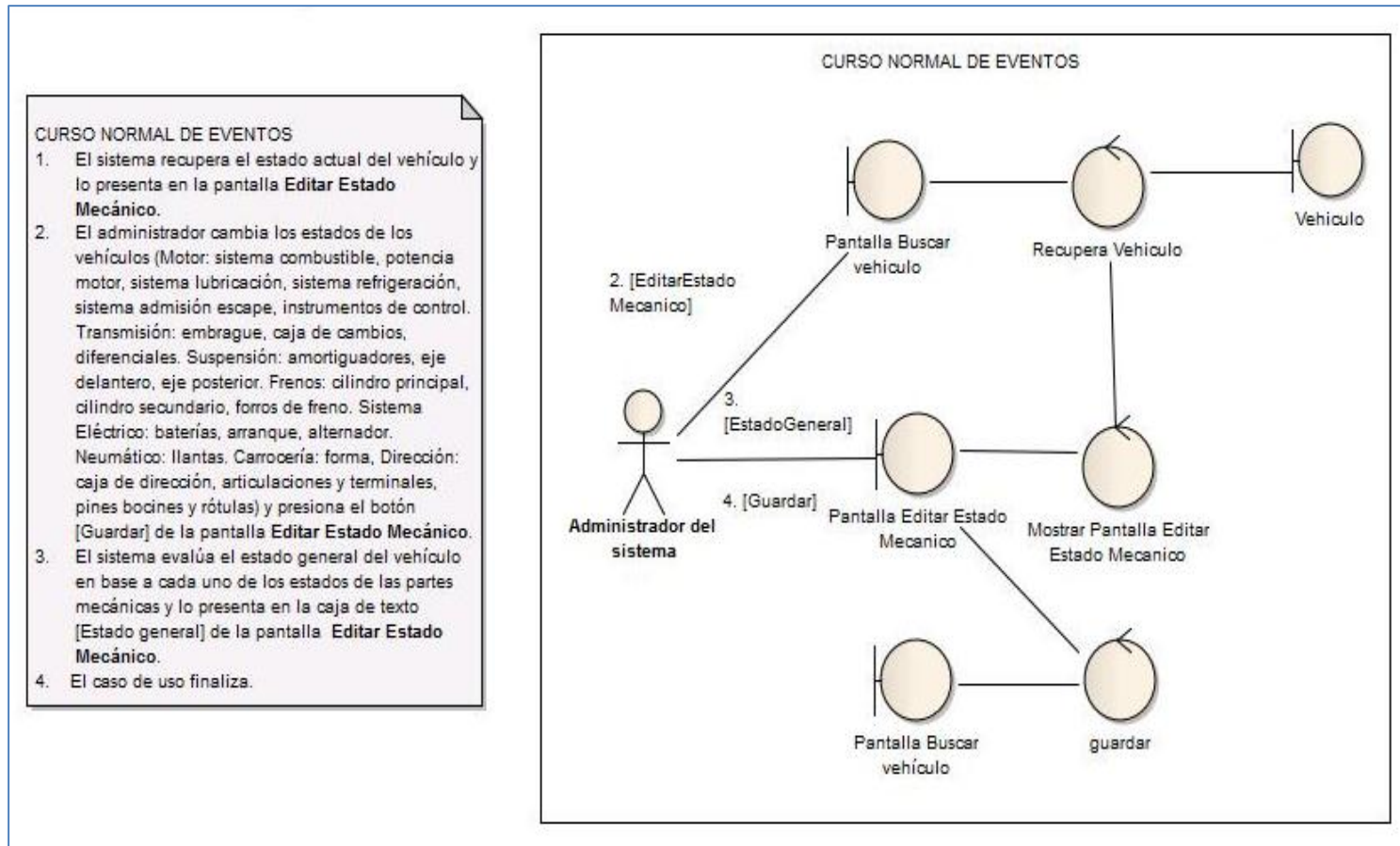


Figura 19. Diagrama de robustez CU002 curso normal de eventos.

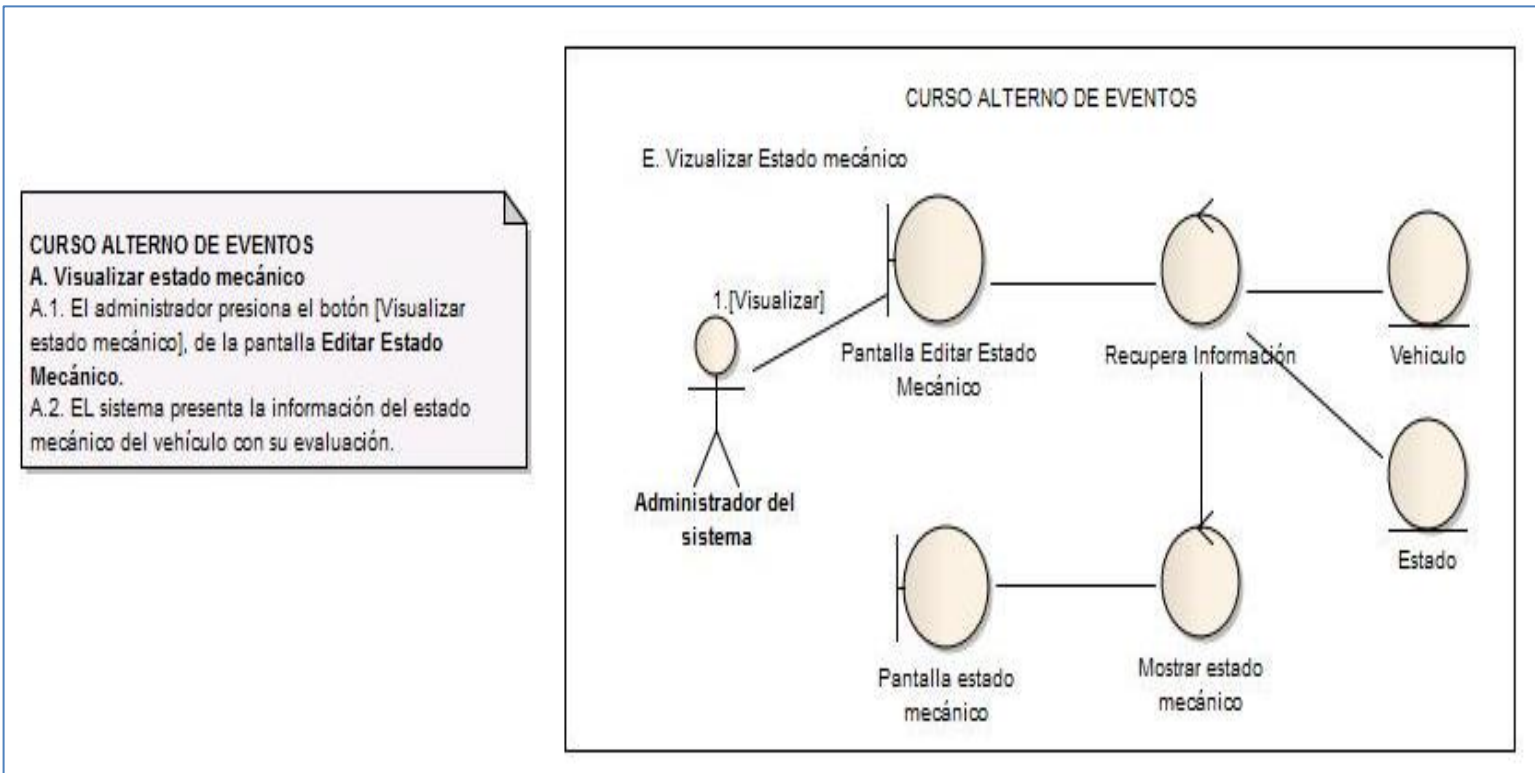


Figura 20. Diagrama de robustez CU002 curso alterno de eventos.

- Registrar estado de ubicación del vehículo.

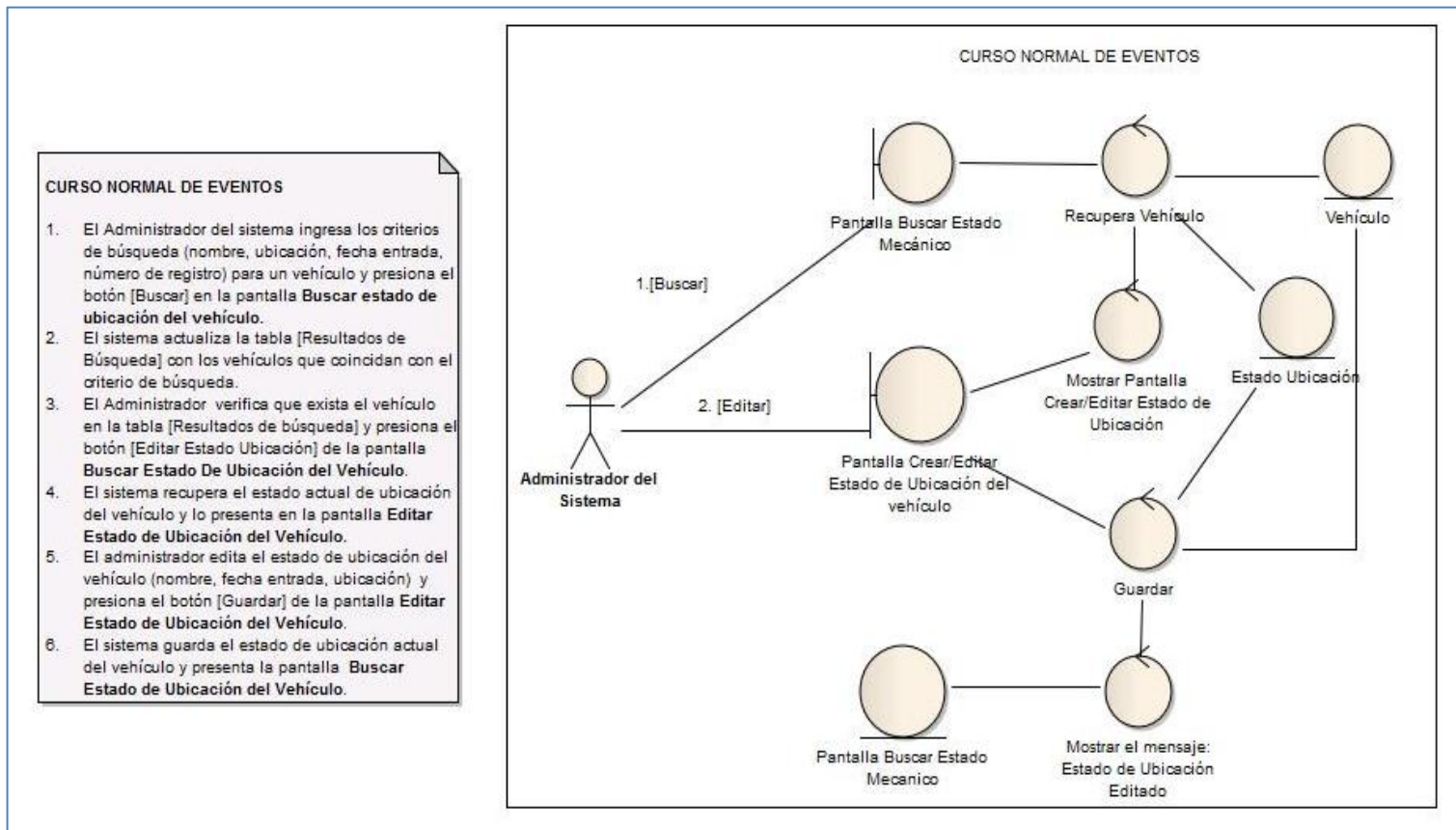


Figura 21. Diagrama de robustez CU03 curso normal de eventos.

CURSO ALTERNO DE EVENTOS
A. Limpiar Búsqueda.
 A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Estado de Ubicación del Vehículo**.
 A.2. El sistema limpia el criterio de búsqueda y actualiza en la tabla [Resultados de búsqueda] de la pantalla **Buscar Estado de Ubicación del Vehículo**.
B. Campos Obligatorios Vacíos.
 B.4. El sistema verifica que estén llenos los campos obligatorios (nombre, fecha entrada, ubicación) y continúa en el paso (5).

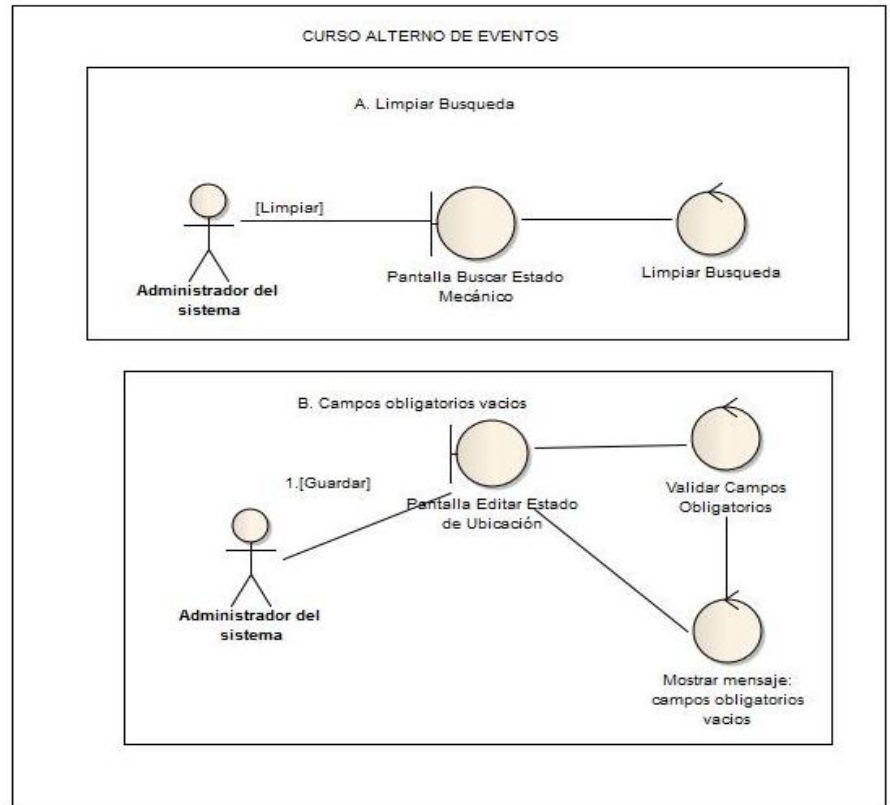


Figura 22. Diagrama de robustez CU003 curso alterno de eventos.

- Registrar kardex.

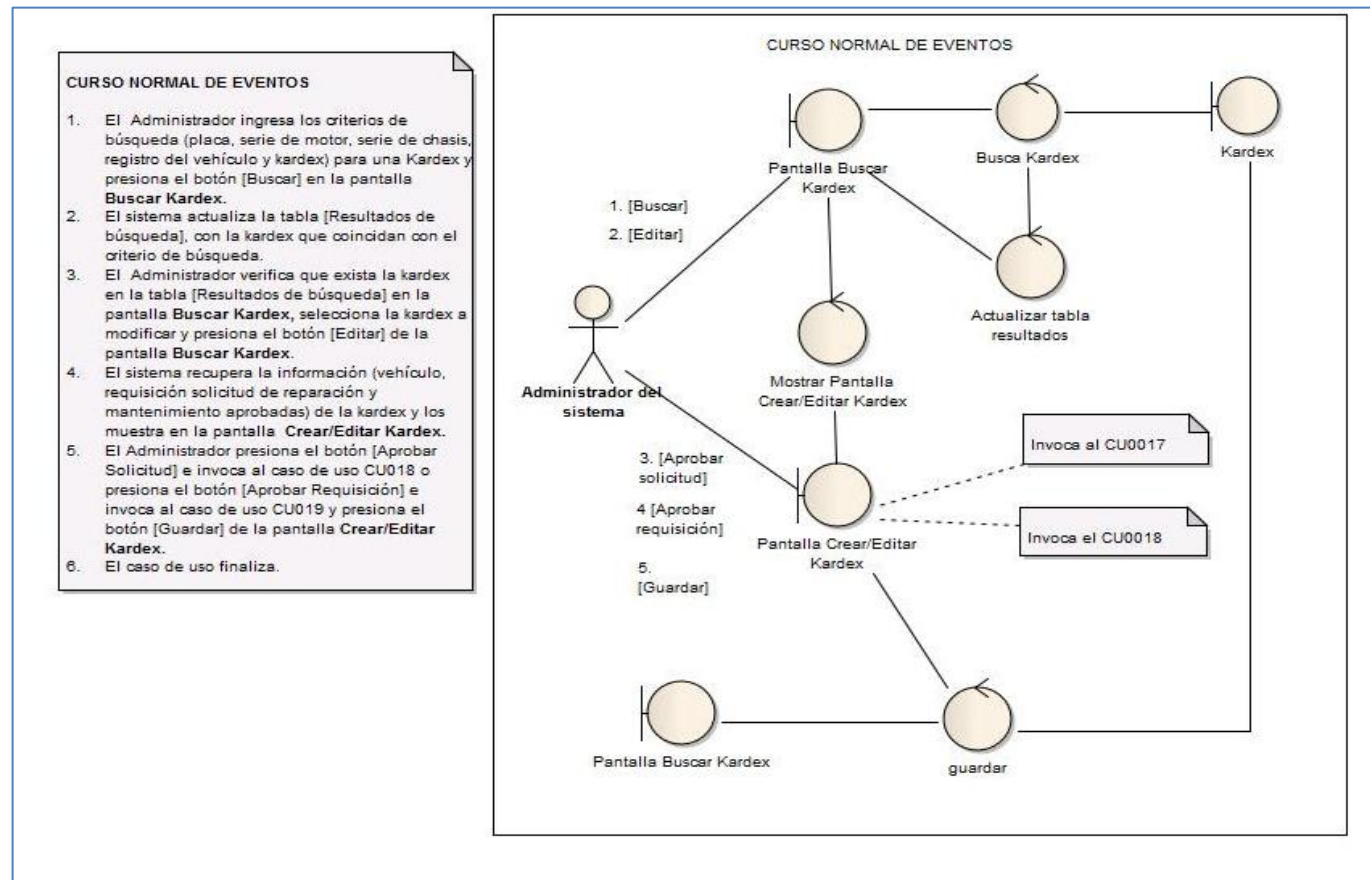


Figura 23. Diagrama de robustez CU017 curso normal de eventos.

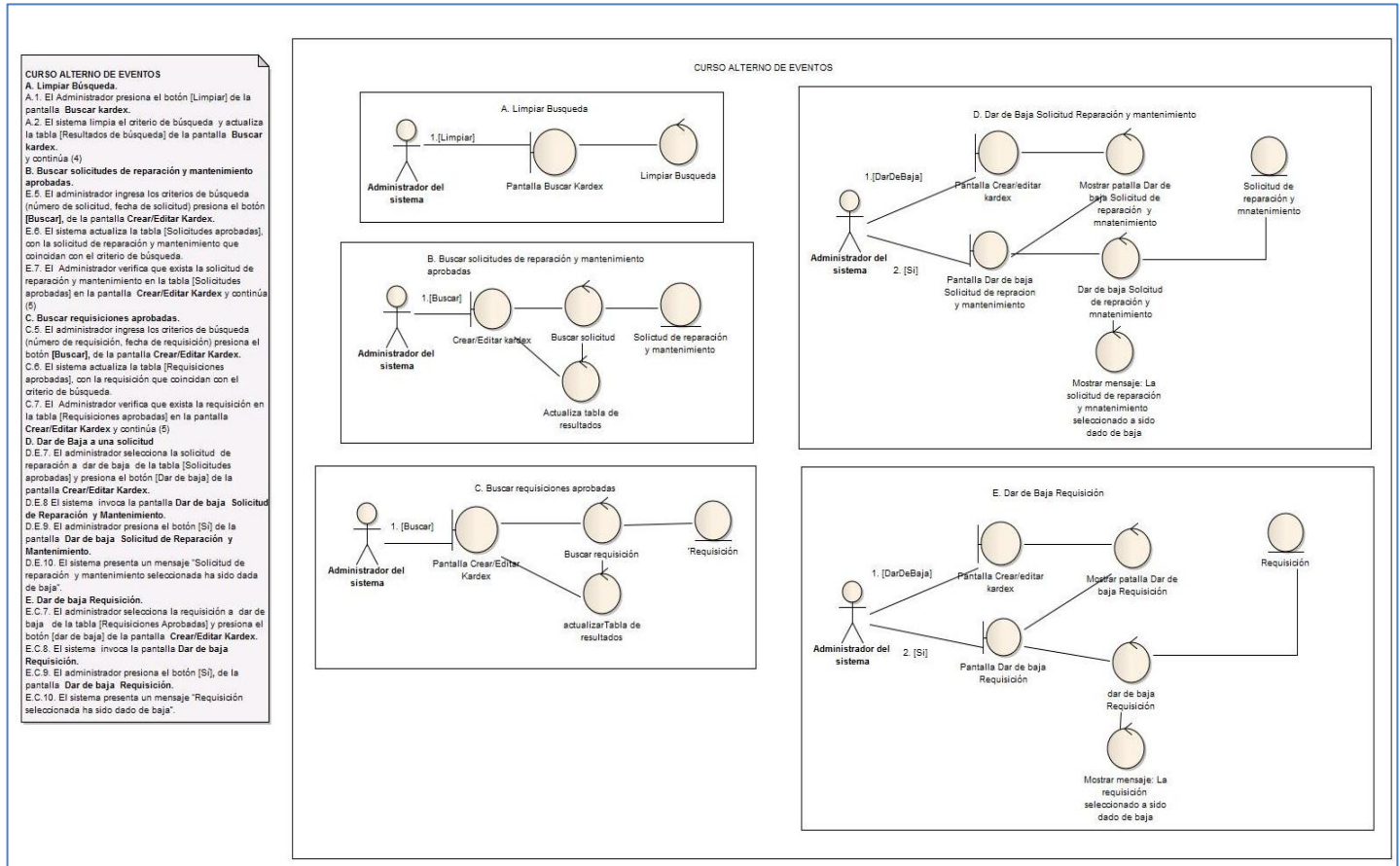


Figura 24. Diagrama de robustez CU017 curso alterno de eventos.

3. Diseño detallado

En esta fase se realizó la construcción de los diagramas de secuencia en los cuales se indican los métodos que llevarán las clases del sistema, así como también se obtuvo la actualización del modelo del dominio y del diagrama de clases.

3.1. Realizar diagrama de secuencia

En esta actividad se indica los diagramas de secuencia de acuerdo a los siguientes casos de uso: gestionar funcionalidades, registrar mantenimiento del vehículo, registrar estado mecánico del vehículo, registrar estado de ubicación y gestionar kardex. El resto de diagramas puede visualizarlos (ver anexo 6).

- Gestionar funcionalidades.

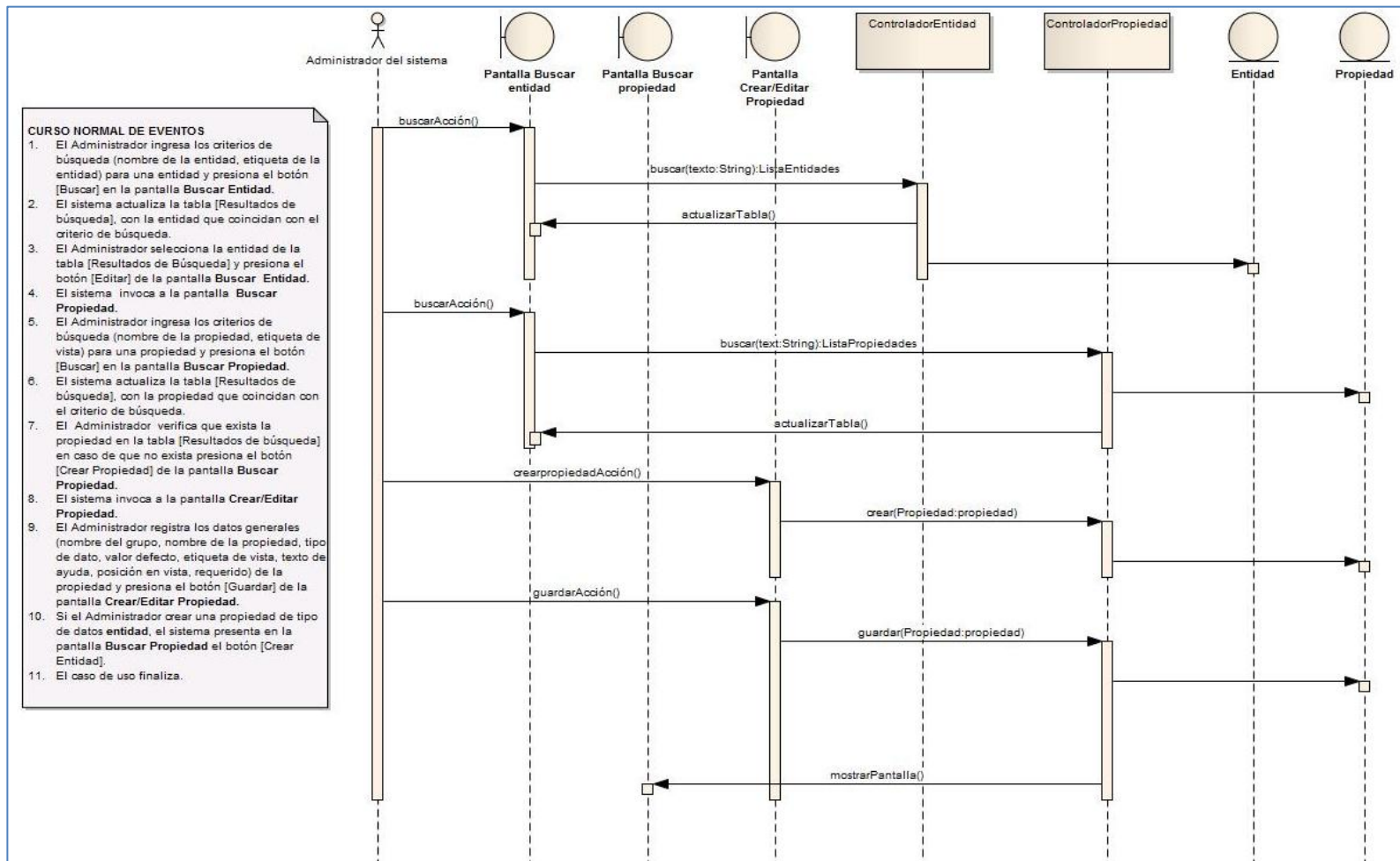


Figura 25. Diagrama de secuencia CU023 curso normal de eventos.

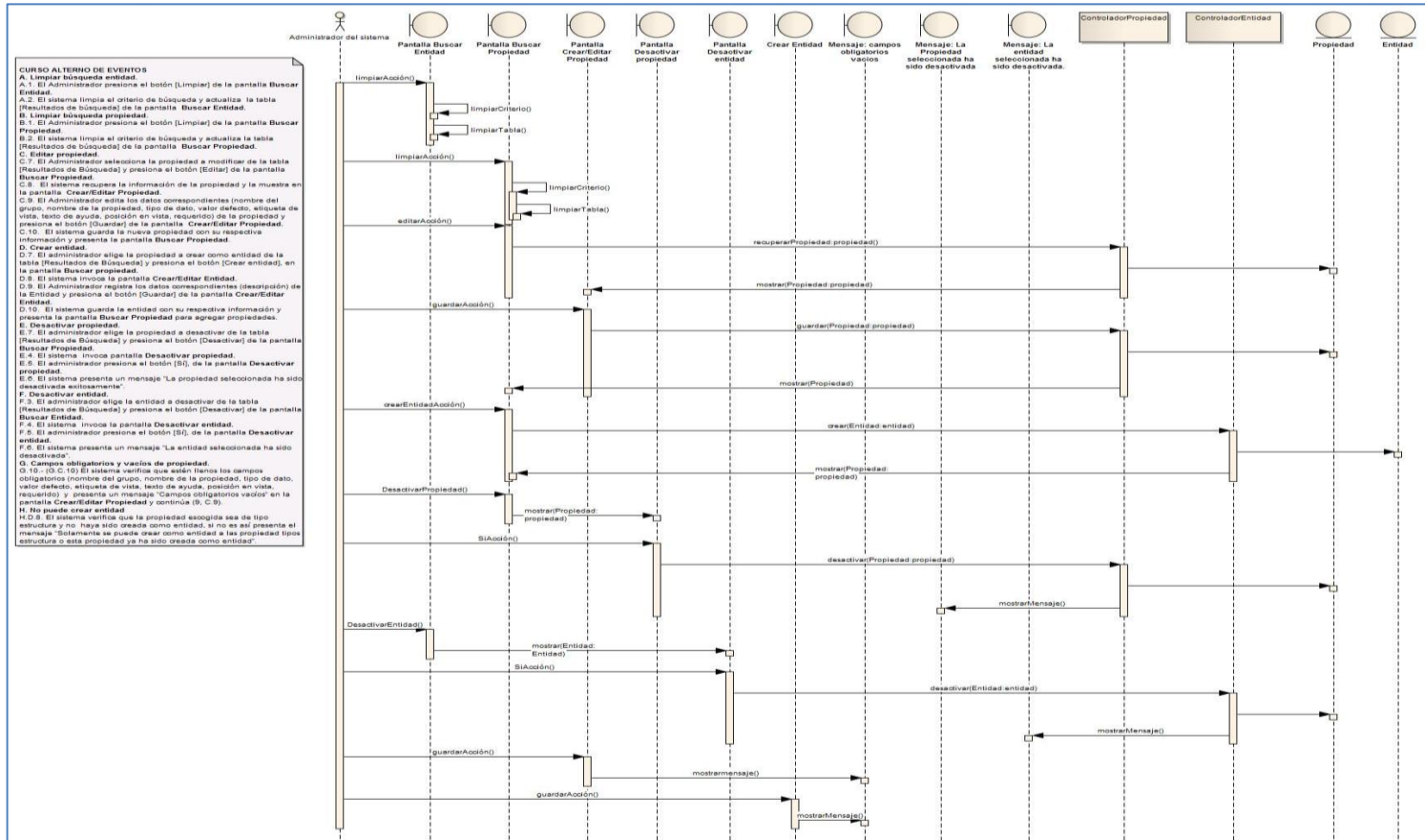


Figura 26. Diagrama de secuencia CU023 curso alternativo de eventos.

- Registrar mantenimiento del vehículo.

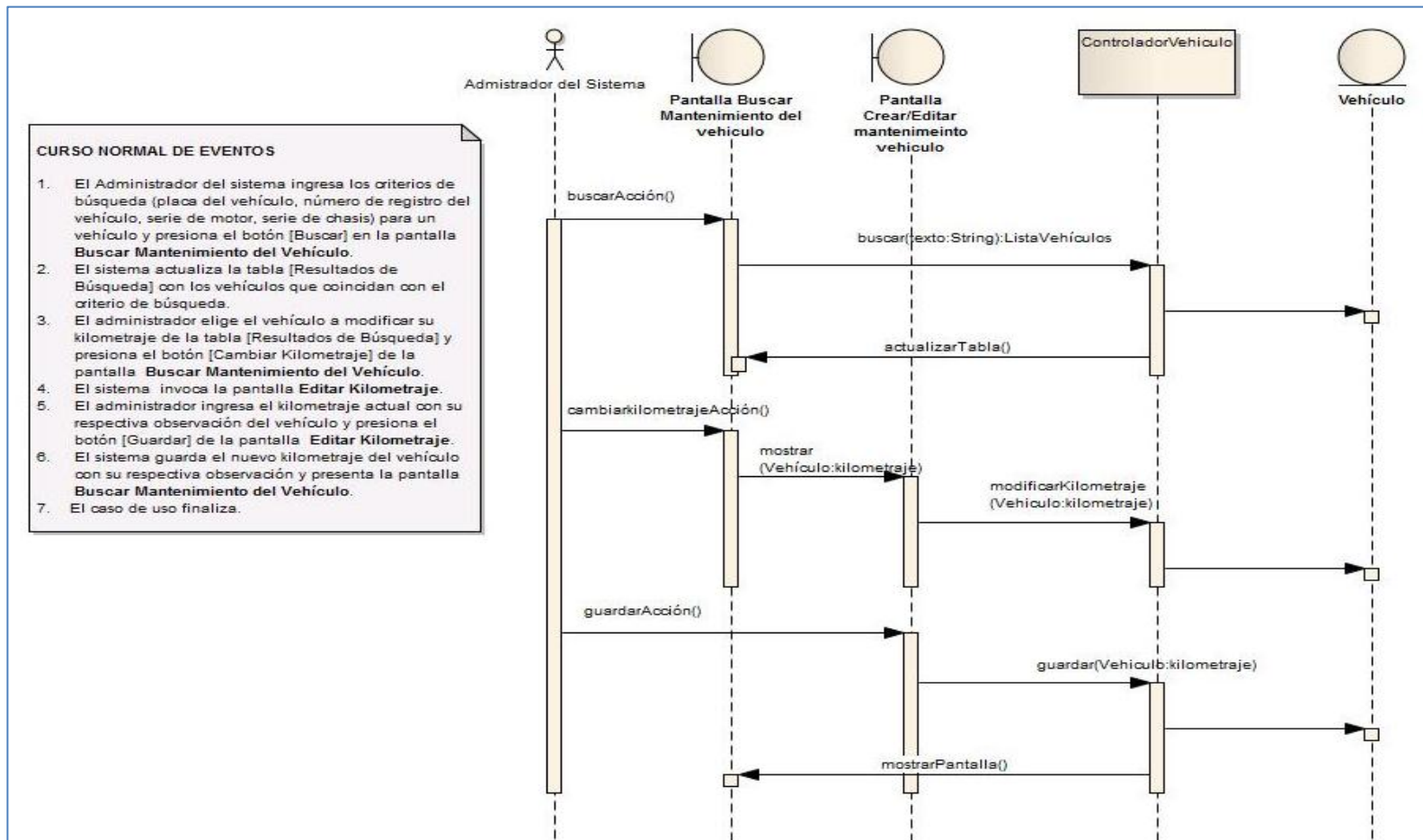


Figura 27. Diagrama de secuencia CU021 curso normal de eventos.

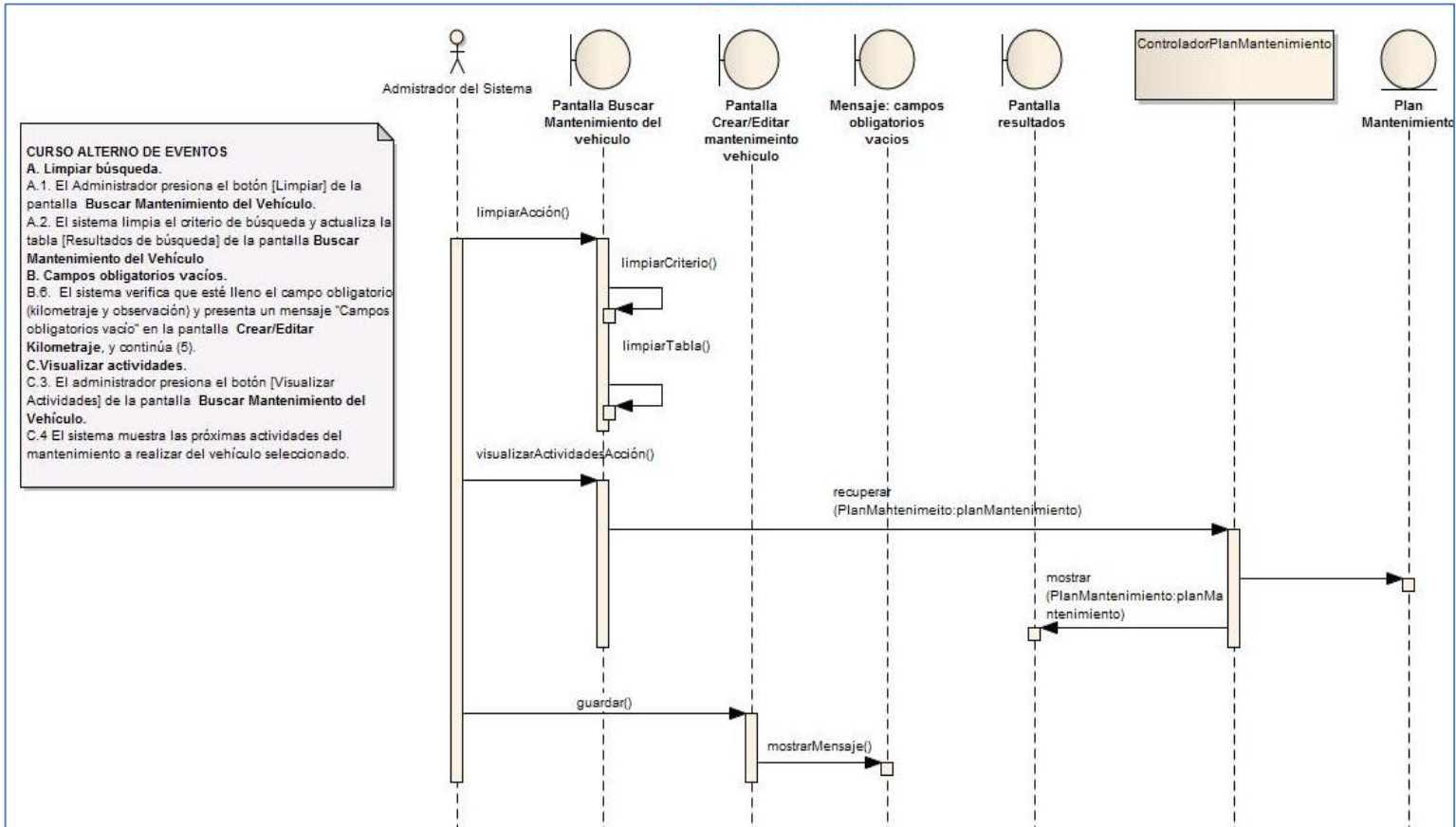


Figura 28. Diagrama de secuencia CU021 curso alternativo de eventos.

- Registrar estado mecánico del vehículo.

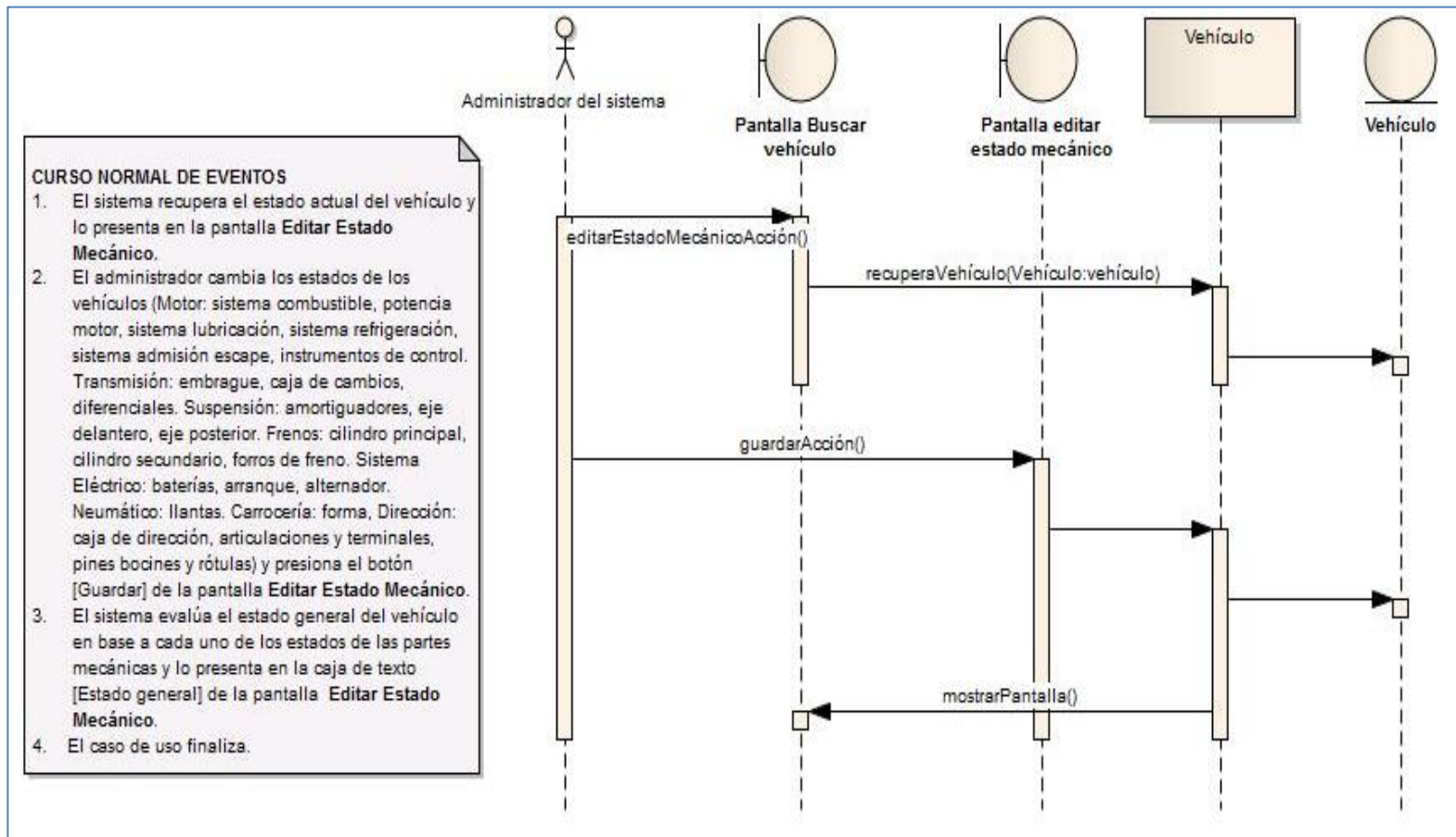


Figura 29. Diagrama de secuencia CU002 curso normal de eventos.

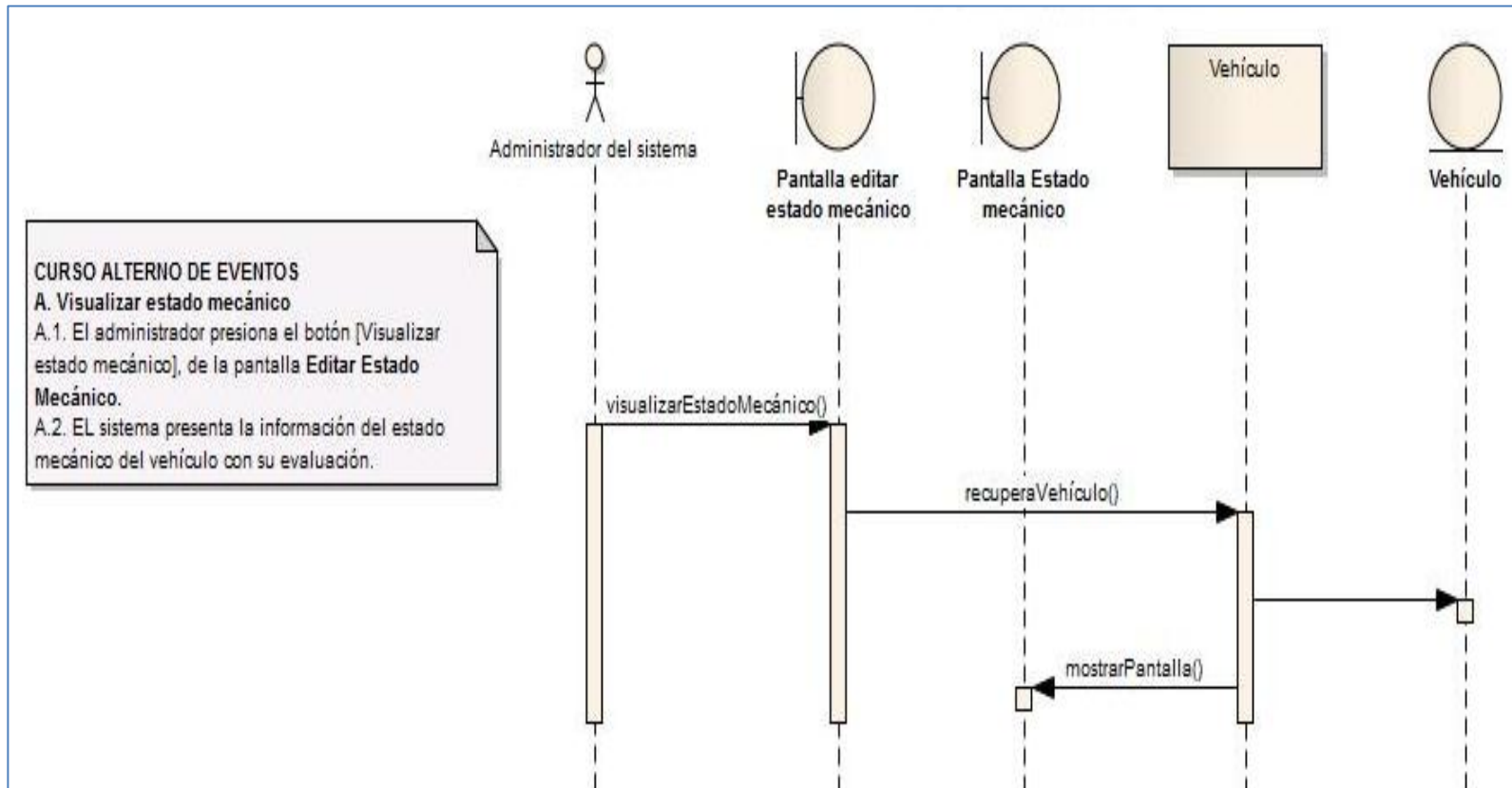


Figura 30. Diagrama de secuencia CU002 curso alternativo de eventos.

- Registrar estado de ubicación del vehículo.

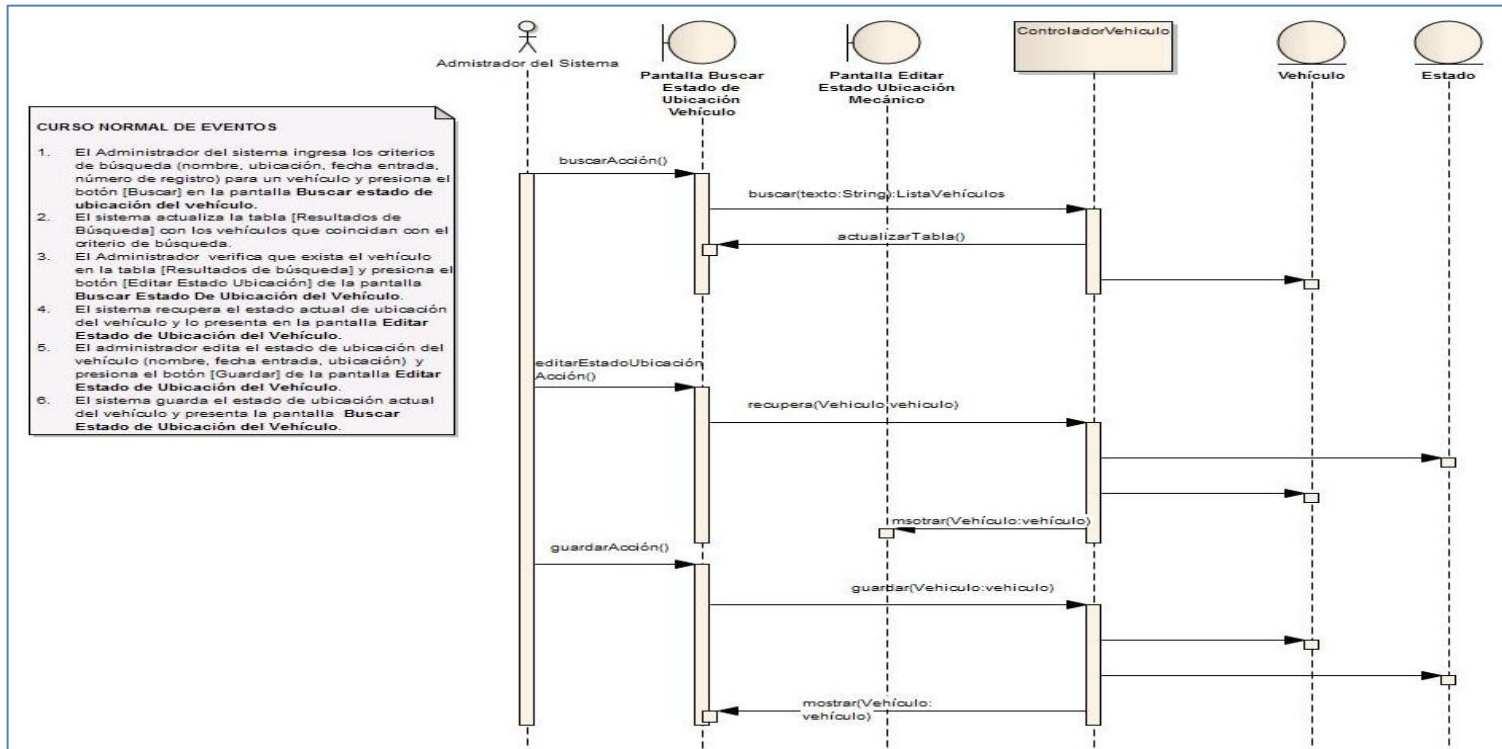


Figura 31. Diagrama de secuencia CU003 curso normal de eventos.

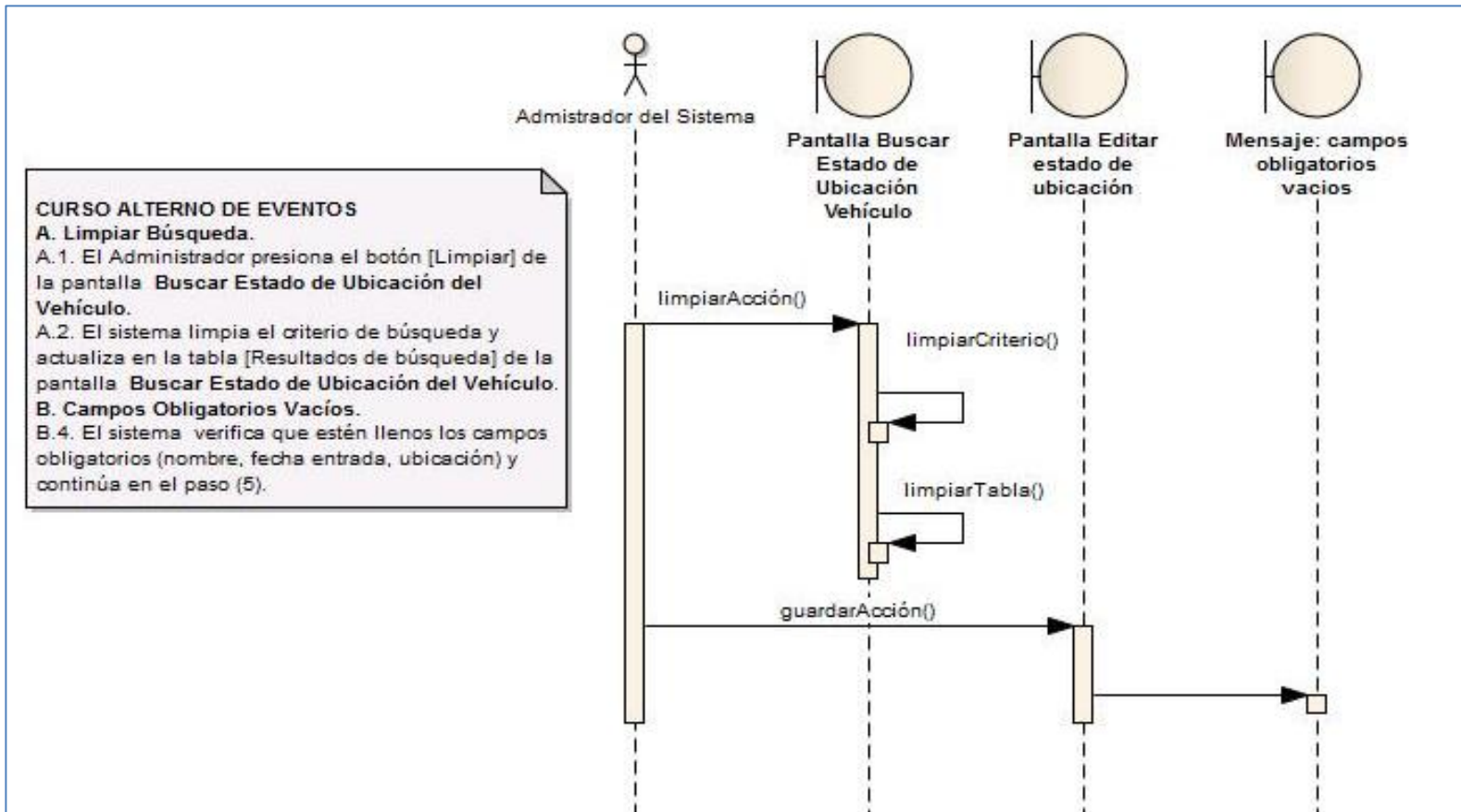


Figura 32. Diagrama de secuencia CU003 curso alterno de eventos.

- Registrar kardex.

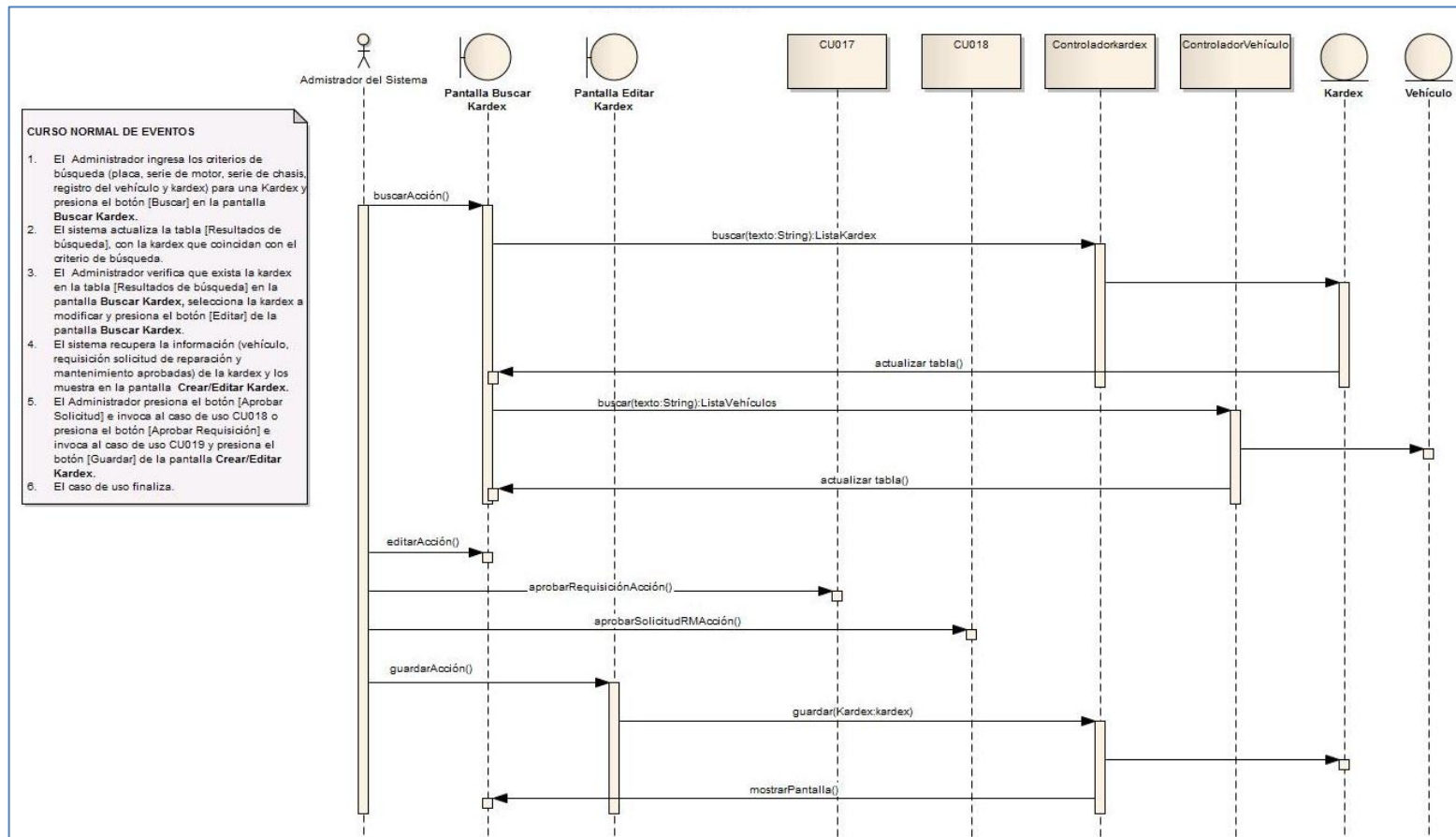


Figura 33. Diagrama de secuencia CU017 curso normal de eventos.

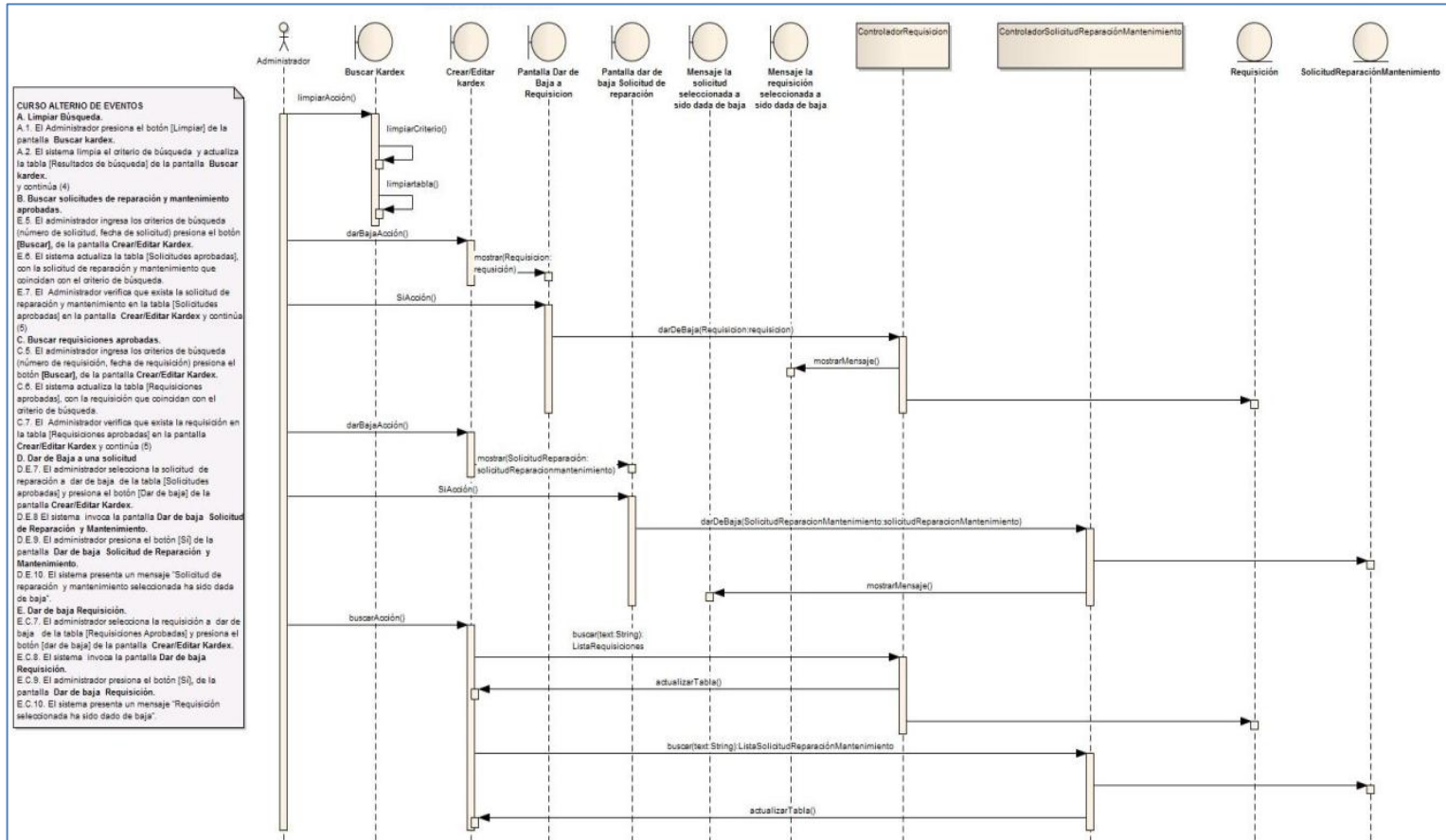


Figura 34. Diagrama de secuencia CU017 curso alterno de eventos.

3.2. Diagrama del modelo del dominio final

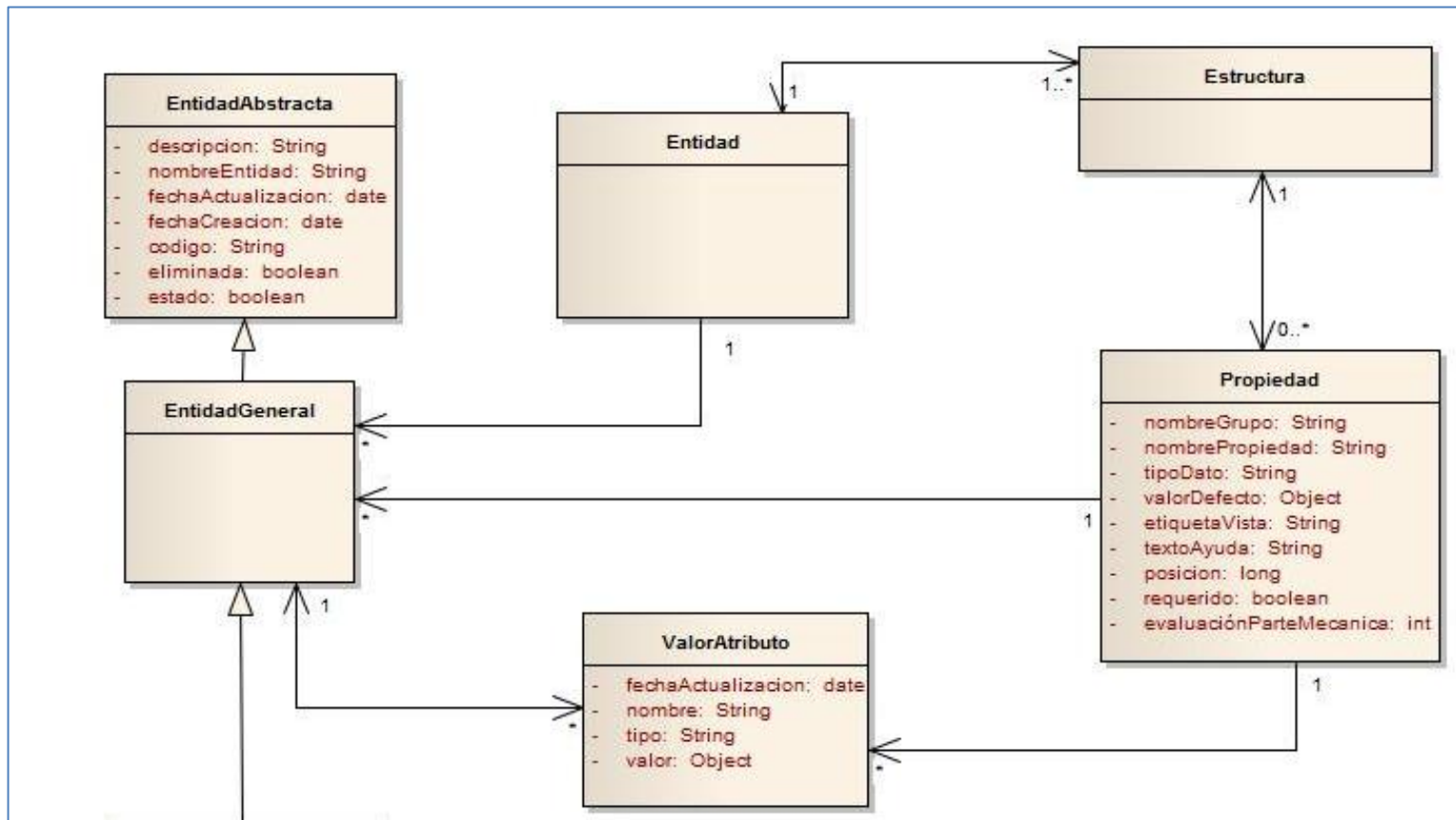


Figura 35. Modelo del dominio final (a)

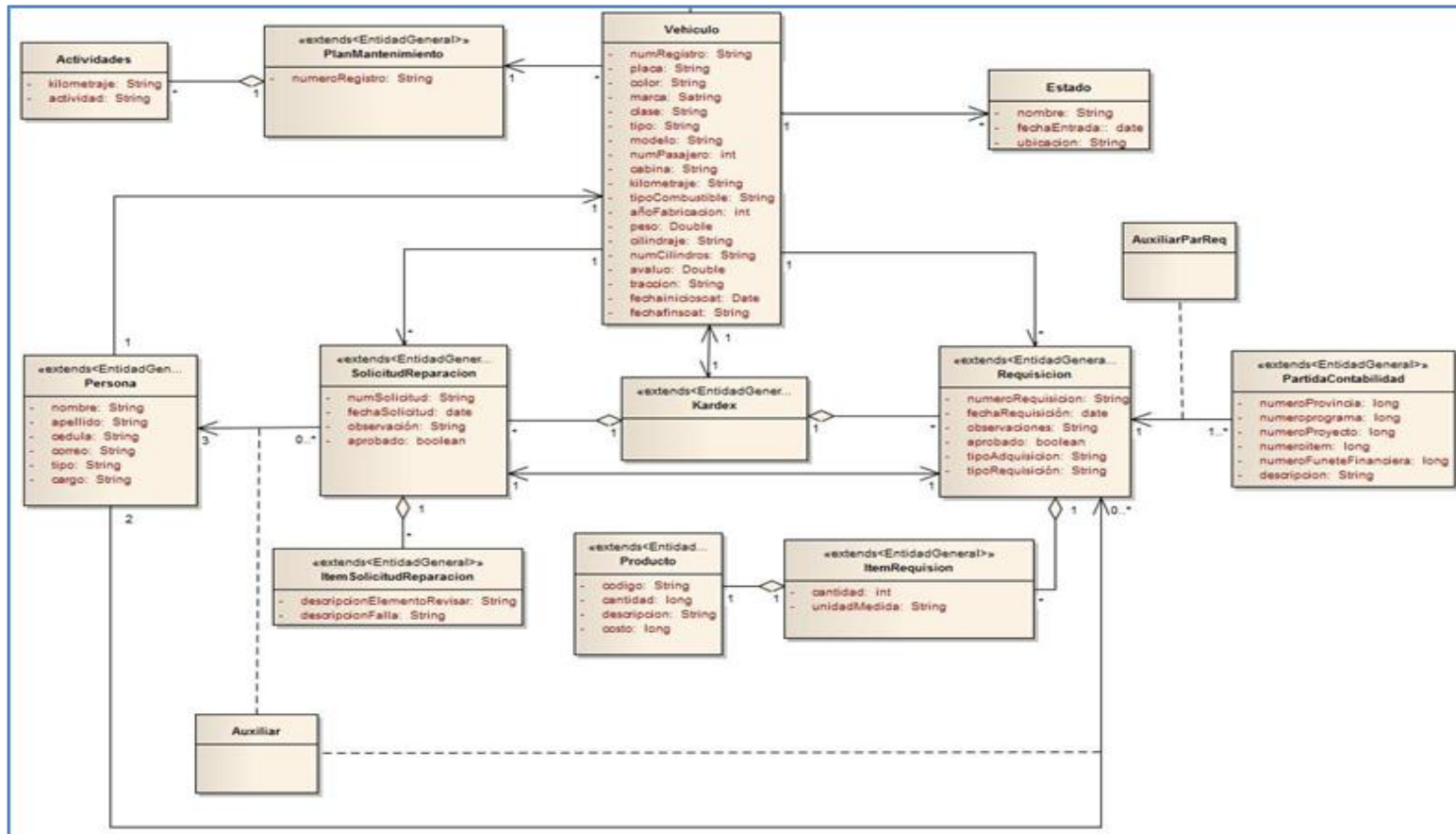


Figura 36. Modelo del dominio final (b)

3.3. Diagrama de clases final

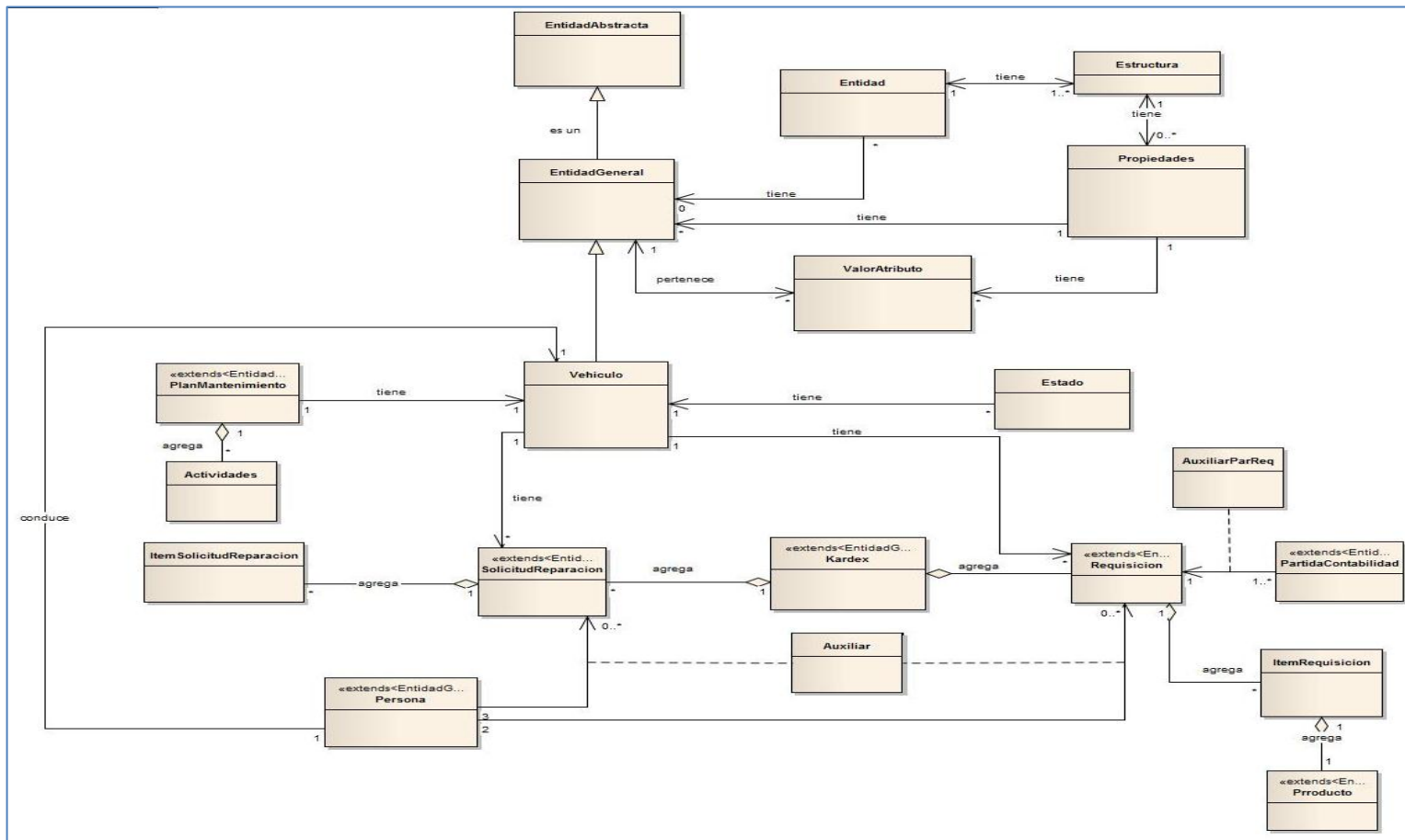


Figura 37. Diagrama de clases final

4. Implementación

4.1 Diagrama de componentes

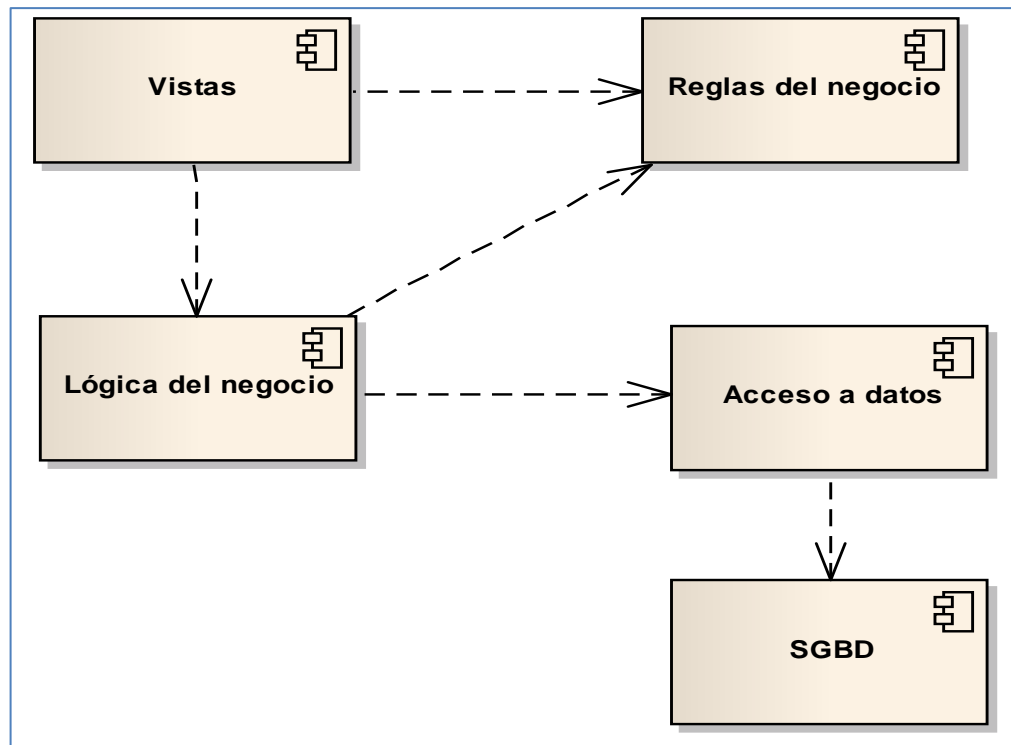


Figura 38. Diagrama de componentes

4.2. Arquitectura de la aplicación

En la figura 39, se muestra la Arquitectura de la Aplicación Web en la cual se observa que los usuarios pueden ingresar desde un computador o dispositivo conectado a la misma red, desde un navegador Web puede acceder a la Aplicación Web.

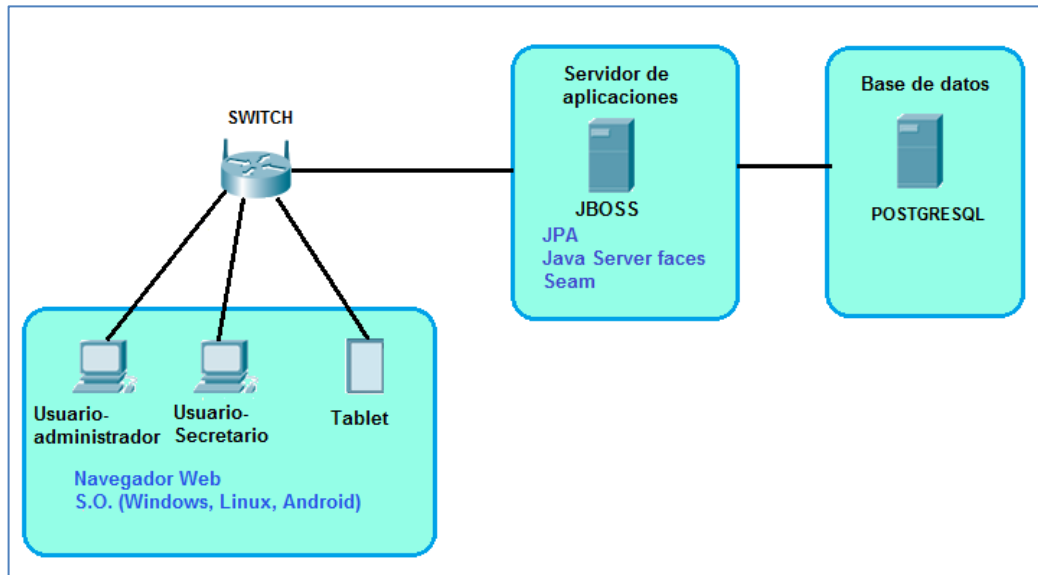


Figura 39. Arquitectura de la Aplicación.

En la tabla XVI se indica la tecnología a utilizar en la Aplicación Web.

TABLA XVI.

RESTRICCIONES DE LA TECNOLOGÍA DE LA APLICACIÓN

	Servidor aplicación	Servidor BD	Cliente
Software	JBoss v2	PostgreSQL v9.1	Navegador Web (Windows-Linux-Android)
Hardware	Dell (Optiplex 780)	Dell (Optiplex 780)	Dell (Optiplex 780)

4.3. Código fuente

A continuación se describirá parte del código utilizado para el módulo de gestión de funcionalidades, el cual fue realizado utilizando como modelo el software empresarial para gestión de proyectos, que se basa en reglas de negocio con configuraciones parametrizables acoplándose a cualquier módulo de una empresa [35], el cual sirvió de base para resolver el problema planteado y de esta forma adaptar la Aplicación Web con dos niveles de personalización permitiendo al usuario agregar en un futuro nuevos

cambios en el sistema y así evitar grandes inversiones que demandan la implementación de nuevos requerimientos.

Clases a implementar

PersistentObject --→ Entidad Abstracta

Esta clase se implementa de la interfaz serializable, los atributos iniciales son: activationtime, createOn, description, lastupdate, name, versión, deleted. Y el atributo que se agregó fue estado para saber si las entidades que extiendan de esta clase están dadas de baja o no.

BussinesEntity -→ Entidad General

En esta entidad se crea las siguientes relaciones:

- ManyToOne con la clase BussinesEntityType
- OneToMany con la clase BussinesEntityAttribute
- ManyToOne con la clase Property

BussinesEntityType-→ Entidad

En esta entidad se crea las siguientes relaciones:

- OneToMany con la clase Structure

Structure -→ Estructura

En esta entidad se crea las siguientes relaciones:

- ManyToOne con la clase BussinesEntityType
- OneToMany con la clase Property

Property--→ Propiedad

En esta entidad se crea las siguientes relaciones:

- ManyToOne con la clase Structure

BussinesEntityAttribute --→ Valor atributo

En esta entidad se crea las siguientes relaciones:

- ManyToOne con la clase BussinesEntity
- ManyToOne con la clase Property

Creación de propiedades y entidades mediante código.

Como se puede observar en el modelo del dominio existen clases que heredan de la Entidad General, por lo tanto estas deben inicializarse siguiendo el flujo del modelo dinámico (ver fase 2 de la sección de resultados) y de esta forma crear sus **Propiedades**. A continuación se explica cómo está estructurado el modelo dinámico dentro del código, tomando como ejemplo la clase Vehículo.

- Creación del Vehículo como entidad.

Primeramente como se puede observar en la tabla XVII se debe hacer una consulta a la base datos en donde encuentre una **Entidad** con el atributo “name” que será el nombre de la clase Vehículo.

TABLA XVII.
BUSQUEDA ENTIDAD EN LA BASE DE DATOS.

```
TypedQuery<BussinesEntityType> query =
entityManager.createQuery("from BussinesEntityType b where
b.name=:name", BussinesEntityType.class);
query.setParameter("name", Vehiculo.class.getName());
bussinesEntityType = query.getSingleResult();
```

Si no se encuentra la **Entidad** que está buscando, procede a crear una nueva **Entidad** como se muestra en la tabla XVIII, a la cual se fija el nombre de la clase Vehículo y la etiqueta.

TABLA XVIII.
CREACIÓN DE LA ENTIDAD.

```
bussinesEntityType = new BussinesEntityType();
bussinesEntityType.setName(Vehiculo.class.getName());
bussinesEntityType.setLabel("Vehículo");
```

Luego se crea una **Estructura** en la cual se fija de nombre la clase Vehículo, la fecha de creación y la fecha de actualización ver tabla XIX.

TABLA XIX.
CREACIÓN DE LA ESTRUCTURA

```
Date now = Calendar.getInstance().getTime();
Structure structure = null;
structure = new Structure();
structure.setName(Vehiculo.class.getName());
structure.setCreatedOn(now);
structure.setLastUpdate(now);
```

Seguidamente como se observa en la tabla XX se crea una lista de **Propiedades** en la cual se va construyendo las **Propiedades** tipo estructura de la clase Vehículo utilizando el método **builStructureTypeProperty**.

TABLA XX.
CREACIÓN DE LISTA DE PROPIEDADES

```
List<Property> attributes = new ArrayList<Property>();
attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Chasis", "Chasis", "Información de Chasis",
    "/paginas/vehiculo/crear", 1L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Motor", "Motor", "Información del Motor",
    "/paginas/vehiculo/crear", 2L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Historial", "Historial", "Información del Historial",
    "/paginas/vehiculo/crear", 3L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "SistemaElectrico", "Sistema Eléctrico", "Información del Sistema Eléctrico",
    "/paginas/vehiculo/crear", 4L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Lubricantes", "Lubricantes", "Información de Lubricantes",
    "/paginas/vehiculo/crear", 5L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Carroceria", "Carrocería", "Información de Carrocería",
    "/paginas/vehiculo/crear", 6L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Direccion", "Dirección", "Información de Dirección",
    "/paginas/vehiculo/crear", 7L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo",
    "Frenos", "Frenos", "Información de Frenos",
    "/paginas/vehiculo/crear", 8L));
```

```

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo
", "Neumaticos", "Neumáticos", "Información de los Neumáticos",
"/paginas/vehiculo/crear", 9L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo
", "Transmision", "Transmisión", "Información de la transmisión",
"/paginas/vehiculo/crear", 10L));

attributes.add(buildStructureTypeProperty("org.mtop.modelo.Vehiculo
", "Suspension", "Suspensión", "Información de la suspensión",
"/paginas/vehiculo/crear", 11L));

```

Luego de haber creado la lista de **Propiedades** tipo “estructura”, las mismas son fijadas a la **Estructura** y esta añadida a la **Entidad** creada anteriormente ver tabla XXI.

TABLA XXI.

FIJAR PROPIEDADES Y ESTRUCTURA A LA ENTIDAD

```

structure.setProperties(attributes);
bussinesEntityType.addStructure(structure);
entityManager.persist(bussinesEntityType);
entityManager.flush();

```

El método **buildStructureTypeProperty** se indica en la tabla XXII, el cual es utilizado para construir **Propiedades** tipo “estructura”.

TABLA XXII.

CONSTRUCCIÓN DE PROPIEDADES TIPO ESTRUCTURA

```

private Property buildStructureTypeProperty(String groupName,
String name, String label, String helpinline, String customForm,
Long sequence) {
    Property property = new Property();
    property.setName(name);
    property.setType(Structure.class.getName());
    property.setValue(null);
    property.setRequired(true);
    property.setGroupName(groupName);
    property.setLabel(label);
    property.setHelpInline(helpinline);
    property.setCustomForm(customForm);
    property.setShowDefaultBussinesEntityProperties(false);
    property.setGeneratorName(null);
    property.setMaximumMembers(null);
    property.setSequence(sequence);
    return property;
}

```


- Construcción de propiedades para Chasis.

Como se indica anteriormente “Motor” fue creada como una **Propiedad** de tipo “estructura” de la clase Vehículo, seguidamente en la tabla XXIII procedemos a crear “Motor” como una **Entidad**, a la cual le agregamos sus **Propiedades** utilizando el método **buildProperty** y el método **builPropertyParteMecánica**.

TABLA XXIII.
CREAR PROPIEDAD TIPO ESTRUCTURA COMO ENTIDAD

```
private void validarEstructuraParaMotorV() {
    BussinesEntityType bussinesEntityType = null;
    String name = "Motor";
    try {
        TypedQuery<BussinesEntityType> query =
            entityManager.createQuery("from BussinesEntityType b where
                b.name=:name",
                    BussinesEntityType.class);
        query.setParameter("name", name);
        bussinesEntityType = query.getSingleResult();
    } catch (NoResultException e) {
        bussinesEntityType = new BussinesEntityType();
        bussinesEntityType.setName(name);
        bussinesEntityType.setLabel(name);

        Date now = Calendar.getInstance().getTime();
        Structure structure = null;
        structure = new Structure();
        structure.setName(name);
        structure.setCreatedOn(now);
        structure.setLastUpdate(now);
        //Lista de atributos de entidad de negocios
        List<Property> attributes = new ArrayList<Property>();

        attributes.add(buildProperty("Motor", "serieMotor",
            String.class.getName(), null, true, "Serie", "Ingrese la serie para
            el Motor", false, 1L));
        attributes.add(buildProperty("Motor", "marca",
            String.class.getName(), null, false, "Marca", " Ingrese la marca del
            Motor", false, 2L));
        attributes.add(buildProperty("Motor", "modelo",
            String.class.getName(), null, false, "Modelo Tolva", "Ingrese el
            modelo del Motor", false, 3L));
        attributes.add(buildProperty("Motor", "tipoMotor",
            String.class.getName(), null, false, "Tipo", "Ingrese el tipo de
            Motor", false, 4L));
        attributes.add(buildPropertyParteMecanica("Motor",
            "soporte", "org.mtop.modelo.EstadoParteMecanica", "Malo,Bueno*",
            false, "Soporte", " Ingrese el valor para soporte del Motor", false,
            5L, 100));
    }
}
```

```

        attributes.add(buildProperty("Motor", "tanque",
String.class.getName(), null, false, "Tanque", "Ingrese el valor
para tanque del Motor", false, 6L));

        attributes.add(buildPropertyParteMecanica("Motor",
"sistemaCombustible", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Sistema de Combustible", "Ingrese el Sistema
de Combustible para el Motor", false, 7L, 100));
        attributes.add(buildProperty("Motor", "potencia",
String.class.getName(), null, false, "Potencia", " Ingrese la
potencia del Motor", false, 8L));
        attributes.add(buildProperty("Motor",
"sistemaRefrigeracion", String.class.getName(), null, false,
"Sistema de Refrigeración", "Ingrese el valor para sistema de
refrigeración de Motor", false, 10L));
        attributes.add(buildProperty("Motor", "admicionEscape",
String.class.getName(), null, false, "Admisión de Escape", " Ingrese
la admición de escape del Motor", false, 11L));
        attributes.add(buildPropertyParteMecanica("Motor",
"instrumentosControl", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Instrumentos de control", "Ingrese
instrumentos de control del Motor", false, 12L, 100));
        attributes.add(buildPropertyParteMecanica("Motor",
"estadoPotencia", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Potencia", "Escoja el estado para potencia
del motor", false, 13L, 100));
        attributes.add(buildPropertyParteMecanica("Motor",
"sistemaLubricacion", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Sistema de Lubricación", "Escoja el estado
para sistema de lubricación del motor", false, 13L, 100));
        attributes.add(buildPropertyParteMecanica("Motor",
"estadoSistemaRefrigeracion", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Sistema de Refrigeración", "Escoja el estado
para sistema de refrigeración del motor", false, 14L, 100));
        attributes.add(buildPropertyParteMecanica("Motor",
"sistemaAdmEsc", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Sistema de Admisión y Escape", "Escoja el
estado para sistema de admición y escape del motor", false, 15L,
100));
        attributes.add(buildPropertyParteMecanica("Motor",
"sistemaInsCon", "org.mtop.modelo.EstadoParteMecanica",
"Malo,Bueno*", false, "Sistema de Instrumentos de Control", "Escoja
el estado para sistema de instrumentos de control y escape del
motor", false, 16L, 100));
        //Agregar atributos
        structure.setProperties(attributes);
        bussinesEntityType.addStructure(structure);
        entityManager.persist(bussinesEntityType);
        entityManager.flush();
    }
}

```

El método **builProperty** se indica en la tabla XXIV, el cual fue utilizado para construir **Propiedades**.

TABLA XXIV.
CONSTRUCCIÓN DE PROPIEDADES

```
private Property buildProperty(String groupName, String name, String
type, Serializable value, boolean required, String label, String
helpinline, boolean showInColumns, Long sequence) {
    Property property = new Property();
    property.setGeneratorName(null);
    property.setGroupName(groupName);
    property.setName(name);
    property.setType(type);
    property.setValue(value);
    property.setRequired(required);
    property.setLabel(label);
    property.setHelpInline(helpinline);
    property.setRender(null);
    property.setCustomForm(null);
    property.setShowInColumns(showInColumns);
    property.setSequence(sequence);
    return property;
}
```

En el método **buildPropertyParteMecanica** que se muestra en la tabla XXIV se puede observar el uso del atributo “evaluaciónparte” que fue agregado al modelo original de la clase **Propiedad**, el cual fue utilizado para realizar la evaluación de cada una partes mecánicas del vehículo (ver anexo 20.5)

TABLA XXV.
CONSTRUCCION DE PROPIEDADES TIPO PARTE MECÁNICA

```
private Property buildPropertyParteMecanica(String groupName, String
name, String type, Serializable value, boolean required, String label,
String helpinline, boolean showInColumns, Long sequence, Integer
evaluacionParte) {
    Property property = new Property();
    property.setGeneratorName(null);
    property.setGroupName(groupName);
    property.setName(name);
    property.setType(type);
    property.setValue(value);
    property.setRequired(required);
    property.setLabel(label);
    property.setHelpInline(helpinline);
    property.setRender(null);
    property.setCustomForm(null);
}
```

```

property.setShowInColumns (showInColumns) ;
property.setSequence (sequence) ;
property.setEvaluacionParte (evaluacionParte) ;
return property;
}

```

Creación de propiedades y entidades mediante la interfaz.

En la figura 40, se indica la lista de **Propiedades** de la entidad Vehículo que se crearon en la sección anterior, en esta lista se puede crear una nueva **Propiedad**, **Entidad** y desactivar una **Propiedad** siempre y cuando no exista ningún objeto creado, se consideró que para poder desactivar una **Entidad** primero se debe desactivar sus propiedades.

Nombre	Etiqueta de vista	Tipo de dato	Valor	Obligatoria	Posición	Crear entidad	Editar	Desactivar
Chasis	Chasis	Estructura		<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>		
motor2	motor2	Estructura		<input type="checkbox"/>	1	<input checked="" type="checkbox"/>		
Motor	Motor	Estructura		<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>		
Historial	Historial	Estructura		<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>		
SistemaElectrico	Sistema Eléctrico	Estructura		<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>		

Figura 40. Lista de propiedades de la entidad Vehículo

- Creación de una propiedad tipo estructura.

Los tipos de datos que se ha considerado para crear una **Propiedad** son: **estructura (Structure)**, **fecha (Date)**, **texto (String)**, **real (Double)**, **entero (Integer)**, **entero mayor (Long)**, **Booelano (Boolean)**, **Lista (String [])**, **área de texto (MultiLineString)**, y el tipo **EstadoParteMecanica** se mostrará cuando la **Entidad** a la que se le está agregando la **Propiedad** sea una propiedad del clase Vehículo.

Para evitar conflictos se ha considerado fijar el nombre del grupo de la **Propiedad** con el mismo nombre de la **Entidad** a la que se está agregando dicha **Propiedad**. En la figura 41 se indica lo parámetros para crear una **Propiedad**.

Datos generales de la propiedad

Nombre del grupo
org.mtop.modelo.Vehiculo
✓ Nombre del Grupo

Nombre de propiedad *
MotorVehiculo
✓ Ingrese el nombre de propiedad

Tipo de dato*
Estructura
✓ Tipo de dato de la propiedad, ejemplo (String, Date, Double, etc...)

Valor
✓ Valor por defecto de tipo de dato

Propiedad requerida
 Seleccione si es requerida esta propiedad

Posición en la vista
1
✓ Valor de secuencia, ejemplo (0,1,2)

Etiqueta de vista*
Motor
✓ Valor que se mostrará en la vista, ejemplo (Nombre, Datos personales)

Línea de ayuda*
motor del vehiculo
✓ Ingrese una descripción de la línea

Guardar

Figura 41. Crear propiedad.

- Crear entidad

La **Propiedad** que fue creada como tipo “estructura” se la podrá crear como **Entidad**, para ello se consideró fijar como nombre de la **Entidad** y etiqueta de vista el nombre y etiqueta respectivamente de la **Propiedad** y de esta forma evitar cualquier inconveniente. En la figura 42, se indica los parámetros para crear de una **Entidad**.

Datos generales

Nombre de la Entidad de Negocio
MotorVehiculo
✓ Nombre de la entidad

Etiqueta
MotorVehiculo
✓ Nombre de etiqueta de vista

Descripción
entidad que identifica el motor del vehiculo
✓ Ingrese una descripción de la entidad

Guardar

Figura 42. Crear entidad

En la figura 43, se muestra la **Propiedad** creada anteriormente como **Entidad** y sus **Propiedades** dentro de la **Entidad** Vehículo.



The image shows a web application interface with a blue header bar containing the text "DATOS ADICIONALES". Below the header is a navigation menu with tabs: "Chasis", "Motor" (which is highlighted in blue), "Historial", "Sistema Eléctrico", "Lubricantes", and "Neumáticos". The main content area contains three input fields, each with a label and a placeholder instruction:

- Serie***: A text input field with the instruction "Ingrese la serie para el Motor" below it.
- Marca**: A text input field with the instruction "Ingrese la marca del Motor" below it.
- Modelo Tolva**: A text input field with the instruction "Ingrese el modelo del Motor" below it.

Figura 43. Propiedades dentro de la clase Vehículo.

Evaluación del estado mecánico del vehículo

Para realizar la evaluación general de un vehículo en base al estado de sus partes mecánicas se lo ha realizado basándose a la ficha técnica de vehículos y estado mecánico (ver anexo 20.5) y también con la ayuda de una entrevista realizada (ver anexo 16), para obtener información de la importancia que tiene cada parte del vehículo y de esta forma poder realizar la evaluación en base a los estado BUENO y MALO.

Para ello se consideró que cuando se cree una **Propiedad** dentro de una **Entidad** que sea **Propiedad** de la **Entidad** Vehículo se mostrará la opción tipo "EstadoParteMecanica" en la cual se debe asignar un porcentaje inicial (ver tabla XXIV) 100% muy importante y 80% importante, como se indica en la figura 44.

localhost:8080/mtop/paginas/admin/bussinesentitytype/property.xhtml?propertyId=62&bussinesEntityType=56&structureId=57&backView=bussinesentit

Bienvenido: Mtop Software de Vehiculos

Ministerio de Transporte y Obras Públicas

Datos generales de la propiedad soporte

Nombre del grupo

 ✓ Nombre del Grupo

Nombre de propiedad *

 ✓ Ingrese el nombre de propiedad

Tipo de dato*

 ✓ Tipo de dato de la propiedad, ejemplo (String, Date, Double, etc...)

Valor

 ✓ Valor por defecto de tipo de dato

Propiedad requerida
 Seleccione si es requerida esta propiedad

Posición en la vista

 ✓ Valor de secuencia, ejemplo (0,1,2)

Etiqueta de vista*

 ✓ Valor que se mostrará en la vista, ejemplo (Nombre, Datos personales)

Línea de ayuda*

 ✓ Ingrese una descripción de la línea

Porcentaje de evaluación *
 Muy importante Importante

Figura 44. Crear propiedades tipo estado parte mecánica.

Para realizar la evaluación en la figura 45, se muestra todas las **Propiedades** que son de tipo “estado parte mecánica” de la **Entidad** Vehículo en el cual se pueda realizar la evaluación de cada parte mecánica y de esta forma obtener la evaluación general del vehículo.

localhost:8080/mtop/paginas/admin/vehiculo/estadoMecanico

Bienvenido: Mtop Software de Vehículos

ESTADO MECÁNICO

Motor Sistema Eléctrico Carrocería Dirección Frenos Neumáticos Transmisión Suspensión

Soporte

Bueno

Seleccione uno...

Malo

Bueno

✓ Ingrese el Sistema de Combustible para el Motor

Instrumentos de control

Bueno

✓ Ingrese instrumentos de control del Motor

Potencia

Bueno

✓ Escija el estado para potencia del motor

Sistema de Lubricación

Bueno

✓ Escija el estado para sistema de lubricación del motor

Sistema de Refrigeración

Bueno

✓ Escija el estado para sistema de refrigeración del motor

Sistema de Admisión y Escape

Bueno

✓ Escija el estado para sistema de admisión y escape del motor

Sistema de Instrumentos de Control

Bueno

✓ Escija el estado para sistema de instrumentos de control y escape del motor

Evaluación del estado General

Evaluar

Evaluación*

Resultado de la evaluación del vehículo

Guardar Ver Estado Mecánico

Figura 45. Evaluación del vehículo.

La evaluación se la ha realizado utilizando el método **obtenerEvaluacion** que se muestra en la tabla XXVI.

TABLA XXVI.
OBTENER EVALUACIÓN

```

public String obtenerEvaluacion() {

    List<Property> propiedades =
servgen.buscarTodoscoincidencia(Property.class,
Property.class.getSimpleName(), Property_.type.getName(),
"org.mtop.modelo.dinamico.Structure");
    List<BussinesEntityAttribute> bea;
    Integer suma = 0;
    Integer contador = 0;
    evaluacion = "Bueno";
    //rompe el bucle del primer for
    fuera:
    for (Property p : propiedades) {
        System.out.println("propiedadess " + p.getName());
        bea =
getInstance().findBussinesEntityAttribute(p.getName());
        for (BussinesEntityAttribute a : bea) {
            if
(a.getProperty().getType().equals("org.mtop.modelo.EstadoParteMecanic
a")) {
                System.out.println("nombre propiedad " +
p.getName());
                System.out.println("nombre de la propiedad de la
propiedad " + a.getName());
                System.out.println("valor" + a.getValue());
                //si la evaluacion parte es muy importante =100 y
la evaluacion es mala
                //inmediatamente se evalúa como malo y sale del
ciclo

if(a.getProperty().getEvaluacionParte().equals(100) &&
a.getValue().equals("Malo")){
                    evaluacion = "Malo";
                    //para romper los bucles anidados
                    break fuera;
                }
                //muelles-amortiguador 80
                //forros 80
                //todos parte motor 100
                //todos parte SistemaElectrico 100
                //todos parte transmisión 100
                //articulacionesTerminales 100
                //ejeDelantero 100
                //ejePosterior 100
                //pinesBocines 100
                // bombaDireccion 100
                System.out.println("nombre " + a.getName() +
"valor: " + a.getValue());
                if (a.getValue().equals("Bueno")) {
                    suma += 100;

```

```

        contador++;
    }
    if (a.getValue().equals("Malo")) {
        suma += 80;
        contador++;
    }
}
}
if (!evaluacion.equals("Malo")) {
    suma = suma / contador;
    if (suma >= 90) {
        evaluacion = "Bueno";
    } else {
        evaluacion = "Malo";
    }
}
return evaluacion;
}

```

Licencia de la aplicación.

La licencia de esta Aplicación Web es tipo GPL (Licencia Pública General) es decir que este software puede ser utilizado de forma libre, redistribuirlo o modificarlo para futuros proyectos ver <http://www.gnu.org/licenses/gpl.html>.

Funcionamiento de la Aplicación Web.

Para tener conocimiento más a fondo sobre el software realizado puede visitar el siguiente link: <https://www.youtube.com/watch?v=z5pgkIAM0Bo>, en el cual se demuestra el funcionamiento del mismo.

Fase 3: Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la Aplicación Web.

En el presente proyecto se aplicaron las pruebas de rendimiento que garantizan que el sistema funcione correctamente y de usabilidad en el cual se ha verificado el ingreso correcto de datos y manipulación de la información por parte del usuario, y de esta

manera cumplir con la verificación de funciones incorrectas o ausentes del sistema, errores en la interfaz de usuario y realizar cambios resultantes.

1. Pruebas de funcionalidad

Para realizar las pruebas de funcionalidad y rendimiento del sistema se ha creído conveniente hacer uso de la herramienta JMeter, la cual permitió simular la concurrencia de un gran número de usuarios haciendo peticiones al servidor y mediante esto dar a conocer los tiempos exactos de respuesta de la aplicación.

Según el artículo “Usando JMeter para pruebas de rendimiento” [21], se menciona que el autor del libro “Usability Engineering” hace referencia a tres límites de tiempos de respuesta de una aplicación de software.

- 0,1 Segundo: es el límite en el cual el usuario siente que está manipulando los objetos desde la interfaz del usuario [21].
- 1 segundo: es el límite en el cual el usuario siente que está navegando libremente sin esperar demasiado una respuesta del servidor [21].
- 10 segundos: es el límite en el cual se pierde la atención del usuario, si la respuesta tarda más de 10 segundos deberá indicar algún mecanismo por el cual el usuario pueda interrumpir la operación [21].

Para la evaluación de los resultados se ha hecho uso de 2 componentes que incluye la herramienta JMeter en su versión 2.11:

1. Agregate Graph: obtiene de forma precisa los datos de la mediana y la línea al 90% [21]:

- URL: etiqueta de la muestra.
- #Muestras: cantidad de thread utilizados para la URL.
- Media: tiempo promedio en milisegundos para un conjunto de resultados.
- Mediana: tiempo promedio del percentil 50.
- Línea del 90%: máximo tiempo utilizado por el 90% de la muestra.
- Min: tiempo mínimo de la muestra de una determinada URL.
- Max: tiempo máximo de la muestra de una determinada URL.

- %Error: porcentaje de requerimientos con errores.
- Rendimiento: rendimiento medio en los requerimientos por segundo/minuto/hora.
- KB/sec: rendimiento medido en Kbytes por segundo.

2. Gráfico de resultados: se muestra de forma gráfica la media, mediana, dispersión y el rendimiento de la aplicación [21].

A continuación se indicará los resultados obtenidos con la herramienta JMeter en la cual se ha definido 3 tests de 50, 100, 150 números de hilos cada uno, los cuales simulan 50, 100, 150 accediendo respectivamente al servidor.

Además se utilizó una lista de enlaces para realizar una simulación de forma aleatoria, los mismos que permitieron recolectar los datos necesarios para la interpretación.

Prueba Nro 1.

Esta prueba fue realizada el día 11/10/14 a las 15:00 hs, se configuraron 50 usuarios, cada 50 segundos. A continuación en la figura 46 se indican los resultados obtenidos.

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/mtop/paginas/a...	50	621	614	921	244	970	0,00%	47,0/min	,4
/mtop/paginas/s...	100	2968	2985	4404	833	5228	0,00%	1,5/sec	,9
/mtop/paginas/s...	100	2590	2865	4086	183	4762	0,00%	1,7/sec	,9
/mtop/paginas/s...	50	565	490	981	47	1366	0,00%	56,5/min	,6
Total	650	2712	1048	7066	12	19299	7,69%	8,6/sec	32,6

Figura 46. Tabla de resultados prueba 1.

Como puede observarse en la figura 46, el tiempo promedio para acceder a una página es 2,712 segundos, realizándose un total de 650 requerimientos al servidor.

El tiempo total utilizado para los 50 usuarios se puede calcular con la siguiente fórmula:

Tiempo Total (TT)= #muestras * media

$$TT = 650 * 2712 = 1762800 \text{ milisegundos}$$

El tiempo promedio total requerido por cada usuario, se puede calcular de la siguiente manera:

Tiempo promedio (TP)= ((Tiempo Total /1000)/60)/cantidad de usuarios

$$TP= ((1762800/1000)/60)/ 50$$

$$TP=29,38/50=0,5876 \text{ minutos}$$

Análisis realizado

En primer lugar se evaluaron los resultados obtenidos a través de un intervalo de confianza para una distribución Normal al 95%. La fórmula del mismo es la siguiente:

$$[TP - Z_{0.95} * D/\sqrt{n}, TP + Z_{0.95} * D/\sqrt{n}]$$

Donde:

- Tiempo promedio (TP) de respuesta es: 2712
- Desviación (D) es: 3963.
- Tamaño de la muestra (n) es de: 650
- $Z_{0.95}$: 1,96

El intervalo resultante es el siguiente:

$$[2712- 1,96 * 3963/\sqrt{650}, 2712+ 1,96 * 3963/\sqrt{650}]$$

$$= [2700,05; 2723,94] \text{ en milisegundos.}$$

$$= [2,70005; 2,72394] \text{ en segundos.}$$

Por lo tanto, se puede esperar que el tiempo de respuesta promedio este entre 2,70 y 2,72 segundos para una cantidad de 50 usuarios simultáneos realizando 650 solicitudes.

Prueba Nro.2

Esta prueba fue realizada el día 11/10/14 a las 15:30 hs, se configuro 100 usuarios en cada 100 segundos. A continuación en la figura 47, se indican los resultados obtenidos.

Etiqueta	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/mtop/paginas/a...	100	398	344	762	45	1260	0,00%	50,0/min	,5
/mtop/paginas/s...	200	1964	1640	3807	166	4837	0,00%	1,6/sec	,9
/mtop/paginas/s...	200	1880	1429	3723	164	4812	0,00%	1,6/sec	,9
/mtop/paginas/s...	100	426	359	844	38	1288	0,00%	49,1/min	,5
Total	1300	1686	551	3891	12	18326	7,69%	10,4/sec	39,3

Figura 47. Tabla de resultados prueba 2

Como puede observarse en la figura 47, el tiempo promedio para acceder a una página es 1,686 segundos, realizándose un total de 1300 requerimientos al servidor.

El tiempo total utilizado para los 100 usuarios se puede calcular con la siguiente fórmula:

Tiempo Total (TT)= #muestras *media

$$TT = 1300 * 1686 = 2191800 \text{ milisegundos}$$

El tiempo promedio total requerido por cada usuario, se puede calcular de la siguiente manera:

Tiempo promedio (TP)= ((Tiempo Total /1000)/60)/cantidad de usuarios

$$TP = ((2191800/1000)/60) / 100$$

$$TP = 36,53 / 100 = 0,3653 \text{ minutos}$$

Análisis realizado

En primer lugar se evaluaron los resultados obtenidos a través de un intervalo de confianza para una distribución Normal al 95%. La fórmula del mismo es la siguiente:

$$[TP - Z_{0,95} * D/\sqrt{n}, TP + Z_{0,95} * D/\sqrt{n}]$$

El intervalo resultante es el siguiente:

$$[1686 - 1,96 * 3039/\sqrt{1300}, 1686 + 1,96 * 3039/\sqrt{1300}]$$

$$= [1520,79; 1851,20] \text{ en milisegundos.}$$

= [1,52079; 1,85120] en segundos.

Por lo tanto, se puede esperar que el tiempo de respuesta promedio este entre 1,5 y 1,8 segundos para una cantidad de 100 usuarios simultáneos realizando 1300 solicitudes.

Prueba Nro.3

Esta prueba fue realizada el día 11/10/14 a las 14:00 hs, se configuro 150 usuarios en cada 150 segundos. En la figura 48 se indican los resultados obtenidos.

Etiqueta	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/mtop/paginas/a...	150	512	495	925	49	1232	0,00%	51,5/min	,5
/mtop/paginas/s...	300	2737	2622	4778	134	5832	0,00%	1,7/sec	1,0
/mtop/paginas/s...	300	2557	2452	4659	241	5782	0,00%	1,7/sec	1,0
/mtop/paginas/s...	150	536	480	965	38	1292	0,00%	50,8/min	,5
Total	1950	2366	875	5680	11	23553	7,69%	10,9/sec	41,0

Figura 48. Tabla de resultado prueba 3

Como puede observarse en la figura 48, el tiempo promedio para acceder a una página es 1,950 segundos, realizándose un total de 2366 requerimientos al servidor.

El tiempo total utilizado para los 150 usuarios se puede calcular con la siguiente fórmula:

Tiempo Total (TT)= #muestras * media

$$TT= 1950*2366= 4613700 \text{ milisegundos}$$

El tiempo promedio total requerido por cada usuario, se puede calcular de la siguiente manera:

Tiempo promedio (TP)= ((Tiempo Total /1000)/60)/cantidad de usuarios

$$TP= ((4613700/1000)/60)/ 150$$

$$TP=76,895/150=0,51263 \text{ minutos}$$

Análisis realizado

En primer lugar se evaluaron los resultados obtenidos a través de un intervalo de confianza para una distribución Normal al 95%. La fórmula del mismo es la siguiente:

$$[TP - Z_{0.95} * D/\sqrt{n}, TP + Z_{0.95} * D/\sqrt{n}]$$

El intervalo resultante es el siguiente:

$$[2366 - 1,96 * 3854/\sqrt{1950}, 2366 + 1,96 * 3854/\sqrt{1950}]$$

$$= [2194,94; 2537,06] \text{ en milisegundos.}$$

$$= [2,19494; 2,53706] \text{ en segundos.}$$

Por lo tanto, se puede esperar que el tiempo de respuesta promedio este entre 2,19 y 2,53 segundos para una cantidad de 150 usuarios simultáneos realizando 1950 solicitudes.

Gráficas obtenidas en JMeter

Las Figuras 48, 49, y 50 muestran los datos obtenidos para la prueba 1, 2 y 3 respectivamente.

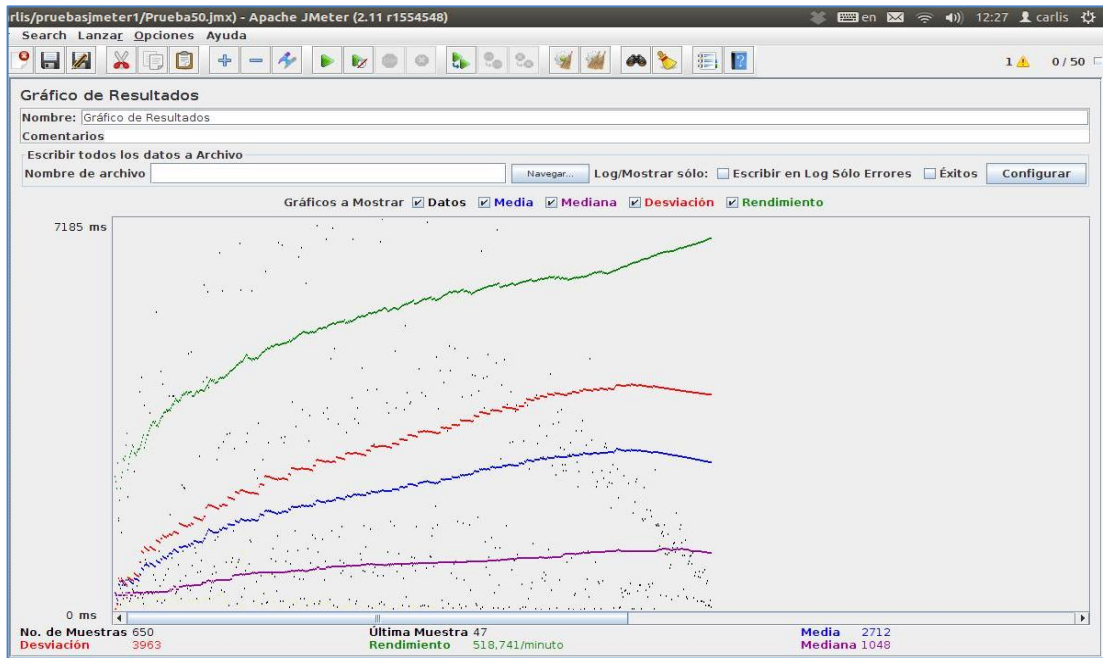


Figura 49. Gráfica de prueba 1

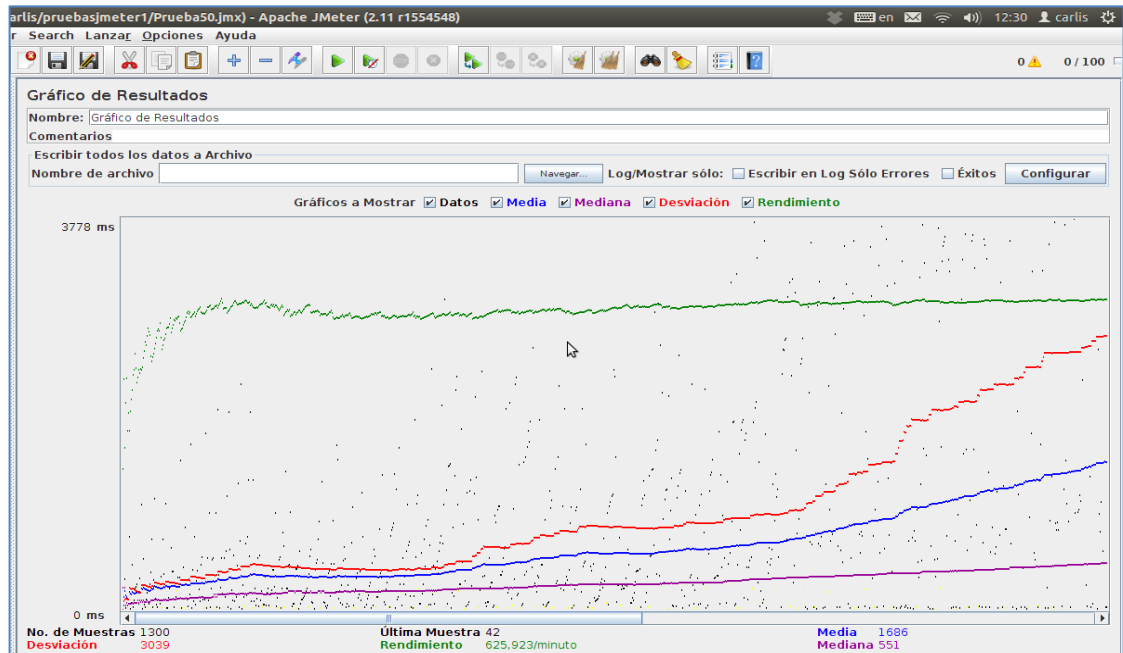


Figura 50. Gráfica de prueba 2

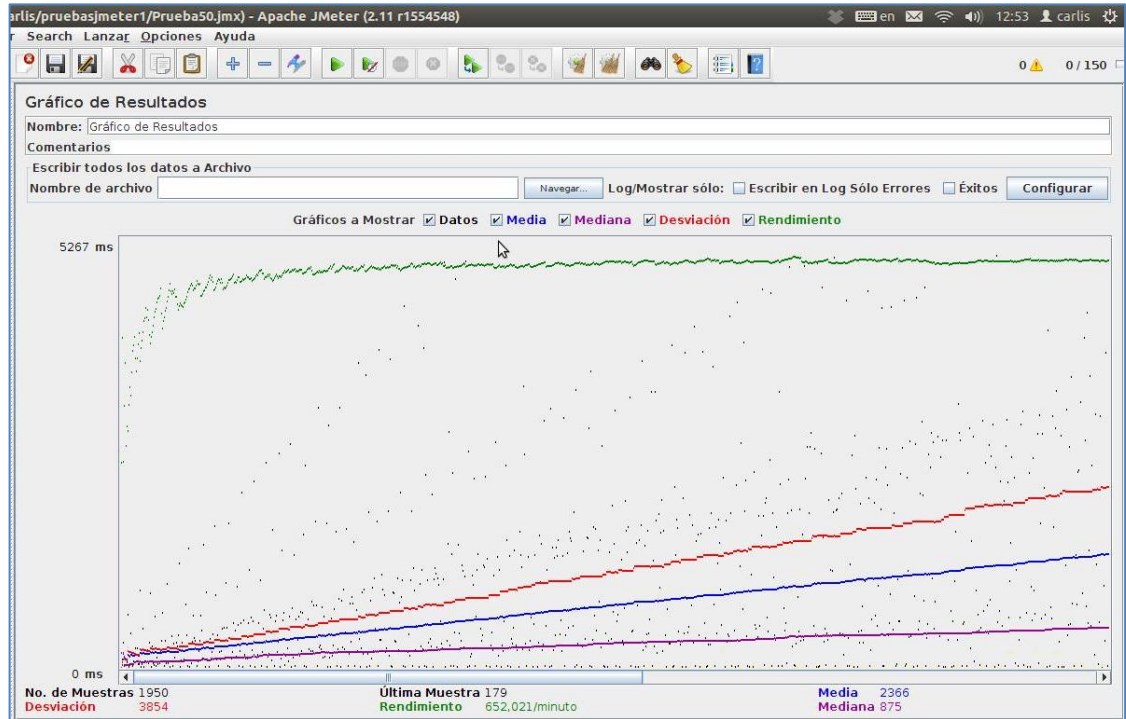


Figura 51. Gráfica de prueba 3

En la tabla XXVII se muestra un resumen de la interpretación de las gráficas indicadas anteriormente, en la cual se puede concluir que los tiempos de respuesta son aceptables con 50 usuarios 2,712 segundos, con 100 usuarios 1,686 segundos y con 150 usuarios 1,950 segundos; al tratarse de una aplicación que consta de dos tipos de usuarios los resultados son los esperados.

TABLA XXVII.

RESUMEN DE LOS DATOS OBTENIDOS EN LAS TRES PRUEBAS.

#Usuarios	#Muestras (tareas)	Rendimiento (peticiones)	Tiempo medio de Respuesta	Dispersión
50	650	518,741/min	2,712 seg	3963
100	1300	625,923/min	1,686 seg	3039
150	1950	652,021/min	2,366 seg	3854

2. Pruebas de usabilidad

Las pruebas de usabilidad indican la utilización de cada uno de los módulos por parte del usuario, describiendo la respuesta de la aplicación, así como mensajes ya sea de información o de validación, verificando si cada requerimiento se ha cumplido o no (ver anexo 10). También se realizó una encuesta para conocer el impacto que el sistema tuvo frente a los usuarios (ver anexo 11).

Módulo de gestión de vehículos

Fecha: 6 de Octubre del 2014

En la tabla XXVIII se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de vehículos.

TABLA XXVIII.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE VEHÍCULOS

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
Es sistema permitirá buscar vehículos.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF001, RF004	Sí
El sistema permitirá crear/editar vehículos.	El sistema muestra los parámetros del vehículo y realiza cada una de las respectivas validaciones.	RF001, RF004	Sí

El sistema permitirá editar el estado mecánico de los vehículos.	El sistema presenta los parámetros para editar el estado mecánico y realiza cada una de las respectivas validaciones.	RF002	Sí
El sistema permitirá editar el estado de ubicación de los vehículos.	El sistema presenta los parámetros para editar el estado de ubicación del vehículo y realiza cada una de las respectivas validaciones.	RF003	Sí
El sistema permitirá editar el mantenimiento de los vehículos.	El sistema presenta los parámetros para editar el mantenimiento del vehículo, y realiza cada una de las respectivas validaciones.	RF016	Sí
El sistema permitirá visualizar las actividades del próximo kilometraje, así como presentar alertas sobre la próxima revisión del vehículo, según el plan activo.	El sistema presenta una tabla del próximo kilometraje con sus respectivas actividades. El sistema presenta en la tabla de resultados de las alertas disponibles de cada vehículo.	RF016	Sí
El sistema permitirá	El sistema presenta opciones para seleccionar	RF017	Sí

imprimir reportes del informe mensual del estado de los vehículos.	el tipo del vehículo, el mes y año para generar los reportes mensuales.		
--	---	--	--

Módulo de gestión de personal

Fecha: 6 de Octubre del 2014

En la tabla XXIX se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión del personal.

TABLA XXIX.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PERSONAL

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
El sistema permitirá buscar el personal.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF0018	Sí
El sistema permitirá crear y editar una persona.	El sistema muestra los parámetros para ingresar los datos de la persona y realiza cada una de las	RF0018	Sí

	respectivas validaciones.		
El sistema permitirá crear un usuario, así como editar las contraseñas del mismo.	El sistema presenta los parámetros del usuario y realiza cada una de las respectivas validaciones.	RF0018	Sí
El sistema permitirá asignar roles (Secretario, administrador) a un usuario.	El sistema presenta una pestaña del usuario en el cual aparece un botón para ir asignar roles, el mismo redirecciona a otra pantalla en la cual se presenta una opción para escoger el rol y un botón para asignar el deseado.	RF0018	Sí
El sistema permitirá dar de baja un rol asignado a un usuario.	El sistema presenta una lista de usuarios y botón para dar de baja un rol asignado al usuario, al momento de presiona el botón presenta un mensaje que al usuario seleccionado se le asignado un rol.	RF0018	Sí
El sistema permitirá desactivar y activar un usuario.	El sistema presenta una lista de usuarios y un botón para desactivar o activar el mismo, al momento de presionar el botón presenta un mensaje de	RF0018	Sí

	confirmación y si elige Sí se muestra un mensaje que el usuario se activado/desactivado con éxito.		
--	--	--	--

Módulo de gestión de solicitudes de reparación y mantenimiento

Fecha: 7 de Octubre del 2014

En la tabla XXX se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de solicitudes de reparación y mantenimiento.

TABLA XXX.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE SOLICITUDES DE REPARACIÓN Y MANTENIMIENTO

Resultado esperado	Resultado verificado	Requerimiento o que cubre	Cumple
Es sistema permitirá buscar solicitudes de reparación y mantenimiento	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF005	Sí
El sistema permitirá crear/editar solicitudes de	El sistema muestra las opciones de ingreso de los parámetros de la solicitud, tiene las respectivas	RF005	Sí

reparación y mantenimiento	validaciones.		
El sistema permitirá asignar ítems de solicitud de reparación y mantenimiento .	El sistema muestra las opciones de ingreso de los parámetros de los ítems de la solicitud, debe ingresar al menos un ítem o se mostrará el mensaje de error. Al eliminar un ítem se presenta un mensaje de confirmación si realmente se desea eliminar y se presenta el mensaje se ha eliminado el ítem con éxito	RF005	Sí
El sistema permitirá asignar un vehículo a la solicitud de la	El sistema muestra la opción de búsqueda de vehículo para poder asignarlo a la solicitud si no se asigna se mostrara un mensaje que se debe asignar un vehículo	RF005	Sí
El sistema permitirá asignar una requisición a la solicitud de reparación y mantenimiento	El sistema presenta una opción para ingresar y autocompletar el criterio para buscar la requisición para asignar a una solicitud si no se encuentra se muestra también la opción de crear la requisición a asignar.	RF006	Sí
El sistema permitirá la búsqueda de	El sistema presenta una opción para ingresar y autocompletar el criterio a	RF005	Sí

solicitudes de reparación y mantenimiento aprobadas.	buscar, de la solicitud aprobada si se encuentra se actualizará la tabla de resultados y además se la puede visualizar		
--	--	--	--

Módulo de gestión de requisiciones.

Fecha: 7 de Octubre del 2014

En la tabla XXXI se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de requisiciones.

TABLA XXXI.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE REQUISICIÓN

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
Es sistema permitirá buscar requisiciones.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF007	Sí
El sistema permitirá crear/editar requisiciones.	El sistema muestra las opciones de ingreso de los parámetros de la solicitud, tiene las respectivas validaciones.	RF007	Sí

<p>El sistema permitirá asignar de los ítems de requisición</p>	<p>El sistema muestra las opciones de ingreso de los parámetros de los ítems de la requisición, debe ingresar al menos un ítem o se mostrará el mensaje de error.</p> <p>Al eliminar un ítem se presenta un mensaje de confirmación si realmente se desea eliminar y se presenta el mensaje se ha eliminado el ítem con éxito</p> <p>Si al elegir el tipo de adquisición de la requisición eligió bodega también se presentara la opción de elegir un producto previamente ingresado y la cantidad deseada a un ítem</p>	<p>RF007,RF009</p>	<p>Sí</p>
<p>El sistema permitirá asignar un vehículo a la requisición de la</p>	<p>El sistema muestra la opción de búsqueda de vehículo para poder asignarlo a la requisición, si el tipo de requisición es de tipo reparación la asignación es obligatoria y si no se asigna se mostrara un mensaje que se debe asignar un vehículo y si la requisición es de tipo bienes y servicios la asignación es opcional</p>	<p>RF007</p>	<p>Sí</p>

El sistema permitirá asignar una solicitud de reparación y mantenimiento a una requisición	El sistema presenta una opción para ingresar y autocompletar el criterio para buscar la solicitud para asignar a una requisición si no se encuentra se muestra también la opción de crear la solicitud a asignar.	RF008	Sí
El sistema permitirá la búsqueda de requisiciones aprobadas.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, de la requisición aprobada si se encuentra se actualizará la tabla de resultados y además se la puede visualizar	RF007	Sí

Módulo de gestión de kardex.

Fecha: 8 de Octubre del 2014

En la tabla XXXII se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de kardex.

TABLA XXXII.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE KARDEX

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
El sistema permitirá buscar kardex por medio de un vehículo, así como buscar	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la	RF012	Sí

requisiciones y solicitudes de reparación y mantenimiento aprobadas y no aprobadas.	tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.		
El sistema permitirá editar kardex.	El sistema presenta un parámetro para ingresa alguna observación si lo desea y un botón para aprobar solicitudes de reparación y mantenimiento así como un botón para aprobar requisiciones, las cuales se listan en cada tabla de resultado correspondiente.	RF012, RF013, RF014.	Sí

Módulo de gestión de planes de mantenimiento.

Fecha: 8 de Octubre del 2014

En la tabla XXXIII se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de planes de mantenimiento.

TABLA XXXIII.
RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PLANES DE MANTENIMIENTO

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
El sistema permitirá buscar planes de mantenimiento.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla	RF0015	Sí

	de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.		
El sistema permitirá crear y editar planes de mantenimiento.	El sistema muestra los parámetros del plan de mantenimiento y realiza cada una de las respectivas validaciones.	RF015	Sí
El sistema permitirá activar un solo plan de mantenimiento.	El sistema presenta una lista de los planes de mantenimiento creados y un botón para activar el plan de mantenimiento deseado, al presionar el botón se muestra un mensaje de confirmación si desea realmente activar el plan y si elige Sí se presenta un mensaje que el plan se activado con éxito.	RF015	Sí
El sistema permitirá visualizar el plan de mantenimiento activo.	El sistema presenta un botón para visualizar el plan de mantenimiento que se encuentra activo.	RF015	Sí

Módulo de gestión de partida de contabilidad.

Fecha: 13 de Octubre del 2014

En la tabla XXXIV se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de partidas de contabilidad.

TABLA XXXIV.
 RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PARTIDAS DE
 CONTABILIDAD

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
Es sistema permitirá buscar partidas de contabilidad	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados, sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF010	Sí
El sistema permitirá crear/editar partida de contabilidad.	El sistema muestra las opciones de ingreso de los parámetros de una partida, tiene las respectivas validaciones.	RF010	Sí
El sistema permitirá dar de baja una partida de contabilidad	El sistema muestra la lista de las partidas de contabilidad, y se presenta la opción de inactivar, al presionar el botón se muestra un mensaje de confirmación si realmente se desea inactivar y si es así se presenta el mensaje se ha inactivado una partida de contabilidad con éxito.	RF010	Sí

Módulo de gestión de productos.

Fecha: 14 de Octubre del 2014

En la tabla XXXV se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de productos.

TABLA XXXV.
RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE PRODUCTOS

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
Es sistema permitirá buscar productos.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF011	Sí
El sistema permitirá crear/editar productos.	El sistema muestra las opciones de ingreso de los parámetros de un producto, tiene las respectivas validaciones.	RF011	Sí
El sistema permitirá dar de baja un producto	El sistema muestra la lista de los productos, y se presenta la opción de dar de baja al presionar el botón se muestra un mensaje de confirmación si realmente se desea dar de baja y si es así se presenta el mensaje se ha dado de baja un producto con éxito.	RF011	Sí

Módulo de gestión de funcionalidades.

Fecha: 14 de Octubre del 2014

En la tabla XXXVI se puede apreciar los resultados de la evaluación de las pruebas de usabilidad realizadas del módulo de gestión de funcionalidades.

TABLA XXXVI.

RESULTADO DE LAS PRUEBAS DE USABILIDAD: GESTIÓN DE FUNCIONALIDADES

Resultado esperado	Resultado verificado	Requerimiento que cubre	Cumple
Es sistema permitirá buscar entidades.	El sistema presenta una opción para ingresar y autocompletar el criterio a buscar, si se encuentra coincidencias se actualiza la tabla de resultados sino se presenta los respectivos mensajes de información que no se ha encontrado o que ingrese el criterio.	RF019	Sí
El sistema permitirá crear/editar propiedades para una entidad.	El sistema muestra todas las entidades, se elige la opción editar a la entidad que se desee agregar propiedades, y se presenta la opción de crear la propiedad con los parámetros de una propiedad, tiene las respectivas validaciones.	RF019	Sí
El sistema permitirá dar de baja una propiedad.	El sistema muestra la lista de las propiedades, y se presenta la opción de inactivar al presionar el botón se muestra un mensaje de confirmación si realmente se desea inactivar y si es así se presenta el mensaje se ha inactivado la propiedad con éxito.	RF019	Sí

	Solamente se puede inactivar las propiedades que aún no tenga información.		
El sistema permitirá crear/editar una entidad.	El sistema muestra la opción de crear una propiedad tipo entidad una vez creada como propiedad se la puede crear como entidad. En la lista de las propiedades se muestra la opción para crear la entidad a una propiedad luego se muestra los parámetros de una entidad con las respectivas validaciones.		Sí
El sistema permitirá dar de baja una entidad.	El sistema muestra la lista de las entidades, y se presenta la opción de inactivar al presionar el botón se muestra un mensaje de confirmación si realmente se desea inactivar y si es así se presenta el mensaje se ha inactivado la entidad con éxito. Solamente se puede inactivar las entidades que aún no tengan propiedades, por lo tanto que no contengan información.	RF019	Sí

G. Discusión

1. Desarrollo de la propuesta alternativa

El presente trabajo de titulación se basa en los 3 objetivos planteados los cuales permitieron dar cumplimiento con el objetivo general, a continuación se describe cada una de las fases:

1.1. Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar.

Para dar cumplimiento al primer objetivo se realizó el estado de arte sobre las diferentes metodologías del desarrollo de aplicaciones Web de esta forma se logró realizar un análisis exhaustivo mediante la técnica de recolección bibliográfica. De esta forma se logró realizar un cuadro comparativo para obtener una visión más acertada acerca de las posibles metodologías a implementarse, para ello se ha creído conveniente el estudio de Scrum y XP (Extreme Programming) que se encuentran dentro las metodologías ágiles e Iconix y RUP (Rational Unified Procces) dentro de las tradicionales. Después de haber obtenido las características más relevantes de las metodologías se determinó que la más adecuada para el presente trabajo de titulación es la metodología Iconix, la cual ayudó a llevar de forma organizada el desarrollo de software y permitiendo que el usuario intervenga en cada una de las actividades que esta metodología propone.

1.2. Implementar la metodología seleccionada que permita guiar la gestión del proyecto.

En este objetivo se desarrolló todas las fases de la metodología seleccionada, en la primer fase que es análisis de requerimientos se aplicó el estándar de especificación de requerimientos IEEE 830, utilizando técnicas de recolección de información como es la observación activa, la cual permitió integrarse con cada uno de los usuarios y así poder obtener información detallada sobre el flujo de las actividades que desarrollan en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas y esta manera obtener una lista de requerimientos. También se utilizaron entrevistas (ver anexo

7, 8, 9, 16), con las cuales se logró conocer los objetos que intervienen en el mundo real y así plasmarlos en el primer diagrama a realizar como es el modelo del dominio. Luego de estas actividades se identificaron los actores y caso de uso que intervienen en el sistema, los cuales fueron agrupados de acuerdo al rol del actor en un diagrama de paquetes; para mayor entendimiento de los casos de uso se realizó el prototipo de pantallas, en el cual se indica el flujo de cada una de las actividades del sistema.

La segunda fase consistió en realizar el análisis y diseño preliminar, para su cumplimiento se realizó la descripción de los casos de usos empleando los flujos evento/respuesta en el cual se logra conocer el curso normal y alterno de eventos. Luego se realizó los diagramas de robustez en los cuales se ilustra gráficamente las iteraciones entre los objetos participantes por cada caso de uso.

La tercera fase es realizar el diseño detallado en el cual se cumplió con la construcción de los diagramas de secuencia en donde se indica cómo interactúan los objetos mediante el envío de mensajes unos a otros. También se actualizó el diagrama de clases y el modelo del dominio con el objetivo de verificar si realmente se cumple con los requerimientos identificados.

La cuarta fase se centra en la implementación del software, para ello se procede generar el código de la Aplicación Web tomando en cuenta los factores reusabilidad, extensibilidad y confiabilidad; de esta forma se obtuvo un software funcional que resuelva el problema de estudio del presente trabajo de titulación.

1.3. Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la Aplicación Web.

El desarrollo de este objetivo permitió garantizar la funcionalidad, eficiencia y adaptabilidad del software mediante la utilización de pruebas de rendimiento y usabilidad.

Las pruebas de rendimiento o de funcionalidad permitieron verificar que el sistema funcione correctamente para ello se utilizó el software de distribución gratuito JMeter, en el cual se generó un caso de prueba a través de la navegación de usuarios en las páginas en donde JMeter actúa como proxy para capturar los pasos que se realiza cuando se

navega, además con esta herramienta se obtiene diferentes resultados como gráficas que indican la media, mediana, dispersión y el rendimiento de la aplicación ante las solicitudes realizadas por los usuarios.

Las pruebas de usabilidad (ver anexo 11, 12), permitieron validar la navegabilidad y accesibilidad del software por parte de los usuarios que intervienen en el sistema como son el administrador y el secretario, con la finalidad de conocer la respuesta que tiene el sistema en lo referente a mensaje de información o de error.

2. Valoración técnica económica ambiental

2.1. Valoración técnica económica

El presente trabajo de titulación se ha desarrollado con la utilización de diferentes recursos como son: humanos, materiales y técnicos-tecnológicos. A continuación en la tabla XXXVII se indica detalladamente los recursos humanos que hace referencia al personal que intervino en el desarrollo del software además de contar con el asesoramiento técnico del tutor.

TABLA XXXVII.
RECURSOS HUMANOS

Recursos	Número de horas	Costo de horas	Costo total
Carla Romero	1056	\$ 5.00	\$ 5280.00
Yesica Saquicela	1056	\$ 5.00	\$ 5280.00
Luis Jácome	50	-	-
Subtotal			\$ 10560.00

En la tabla XXXVIII se indica detalladamente los recursos materiales que fueron necesarios para presentar los diferentes avances, entrevistas y el informe final con sus respectivos manuales del trabajo de titulación.

TABLA XXXVIII.
RECURSOS MATERIALES

Recursos	Cantidad	Valor unitario	Valor Total
Cartucho de tinta	5	\$ 25.00	\$ 125.00
Resma de papel	2	\$ 4.00	\$ 8.00
Transporte	2	\$ 0.25	\$ 200.00
Internet	300	\$ 0.60	\$ 18.00
Copias	200	\$ 0.02	\$ 4.00
Anillados	3	\$ 2.00	\$ 6.00
Empastados	3	\$ 4.00	\$ 12.00
Suministros de oficina (cuadernos, lápiz, borradores, perfiles, cd, entre otros).	-	\$ 15.00	\$ 15.00
Subtotal			\$ 388.00

En la tabla XXXIX se muestra cada uno de los recursos técnicos y tecnológicos que se emplearon en el desarrollo del presente trabajo de titulación, como se puede observar existieron algunos que fueron adquiridos de forma gratuita.

TABLA XXXIX.
RECURSOS TÉCNICOS Y TECNOLÓGICOS.

Recursos	Cantidad	Valor unitario	Valor Total
Hardware			
Computadora	2	\$ 1500.00	\$ 3000.00
Impresora	2	\$ 90.00	\$ 180.00
Disco USB	2	\$ 10.00	\$ 10.00
Software			
Netbeans		Gratuito	\$ 0.00

Enterprise Architect		Gratuito	\$ 0.00
JMeter		Gratuito	\$ 0.00
PostgreSQL		Gratuito	\$ 0.00
Servidor JBoss		Gratuito	\$ 0.00
Licencia creative commons		Gratuito	\$ 0.00
Subtotal			\$ 3190.00

En la tabla XL se indica el resumen del presupuesto del proyecto de trabajo de titulación, el cual se realizó en función de la estimación de los recursos humanos, materiales, técnicos y tecnológicos.

TABLA XL.
PRESUPUESTO DEL PROYECTO

Recursos	Subtotal
Recursos humanos	\$ 10560.00
Recursos materiales	\$ 388.00
Recursos técnicos y tecnológicos	\$ 3190.00
Total	\$ 14138.00

2.2 Valoración ambiental

Mediante el análisis exhaustivo se pudo identificar que las actividades que se realizan en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas, demanda gran cantidad de papel, para solventar este inconveniente la utilización del software se considera de gran ayuda para el registro y almacenamiento de solicitudes de reparación y mantenimiento, requisiciones, kardex, vehículos, entre otros, lo cual permite disminuir el impacto producido. De esta forma queda justificada la valoración ambiental del presente trabajo de titulación.

H. Conclusiones

Luego de haber realizado el presente trabajo de titulación se concluye con lo siguiente:

- Al aplicar la metodología Iconix en el desarrollo de software se pudo llevar un flujo adecuado de cada una de las actividades, además se comprobó que una de las fases más importantes de la misma y a la cual se le debe dar mayor énfasis es el análisis de requerimientos, ya que permite obtener lo que el usuario desea.
- La metodología Iconix es muy importante en el desarrollo de un software ya que permitió la comunicación entre los involucrados del proyecto y obtener documentación en todo el proceso del software.
- La utilización del estándar de especificación de requerimientos IEEE 830, ayudó a obtener un documento de requerimientos con el fin de tener un contrato con el cliente, y así obtener un sistema eficiente acorde a las necesidades del mismo.
- Gracias a las pruebas de funcionalidad y de usabilidad se pudo realizar una evaluación al sistema permitiendo obtener el tiempo de respuesta así como verificar su correcto funcionamiento en cuanto a la accesibilidad y eficiencia del mismo.
- De acuerdo al estudio realizado del modelo dinámico se logró implementar un sistema que ayude al usuario a crear nuevos requerimientos futuros, de una forma diferente a la parametrización que se realiza comúnmente en los softwares rígidos.
- Mediante las pruebas realizadas se concluye que la utilización del sistema ayuda que los procesos de registro de información sean de forma rápida y confiable permitiendo tener una mayor organización de las actividades que se realiza.
- Se puede concluir que el desarrollo del presente trabajo de titulación fue muy útil al momento de poner en práctica los conocimientos adquiridos durante la formación como ingenieros en sistemas.
- Obtener documentación suficiente durante el proceso de desarrollo de software es útil para que en un futuro la misma sirva de guía y se pueda ampliar fácilmente según nuevos requerimientos.

I. Recomendaciones

Luego de haber realizado el presente trabajo de titulación se concluye con las siguientes recomendaciones:

- Realizar un análisis correcto de herramientas y tecnología que ayuden en el desarrollo del software y que permitan tomar decisiones de mejoramiento continuo, con el fin de obtener un sistema de calidad.
- Se sugiere tener una buena relación entre el analista y el cliente para así realizar un correcto análisis de requerimientos del software.
- Se debe dedicar el tiempo necesario para realizar el análisis de requerimientos del sistema, debido a que es la base fundamental para la implementación correcta del sistema, considerando que en esta etapa se logra obtener lo que el usuario necesita.
- Es recomendable hacer uso del patrón de diseño MVC (Modelo -Vista -Controlador), con el fin de ir separando las capas de desarrollo que permita la reutilización de código.
- Se debe realizar una interfaz amigable para proporcionar al usuario confianza, con la finalidad de que el sistema cumpla con el objetivo para el que fue creado y produzca que el mismo se utilizado.
- La implementación de un módulo de inventarios en el futuro permitirá sincronizarse con el módulo de productos, el cual permitirá obtener con mayor facilidad los costos estimados de las reparaciones y mantenimientos de los vehículos.
- La generación de los reportes puede ser ampliado en un futuro de forma que los demás departamentos hagan uso de los mismos.
- Al ser este sistema desarrollado para el departamento vial del MTOP de la ciudad de Loja, se lo puedo utilizar como base para nuevos proyectos de gestión de información en diferentes departamentos del Ecuador.

- Con la aplicación del modelo dinámico se ha creído conveniente utilizar dos niveles de adaptabilidad, aunque el modelo podría servir para que en un futuro se pueda hacer uso de más niveles.

I. Bibliografía

- [1] UTPL, “¿Qué es IEEE?”, reporte técnico, publicado el 16 de abril del 2011.
- [2] Página oficial de Enterprise, “Enterprise Architect”, disponible en: <http://www.sparxsystems.com/products/ea/9/>
- [3] Página oficial de Netbeans, “Netbeans”, disponible en: <https://netbeans.org>
- [4] Página oficial de Oracle “Java SE Technologies - Base de Datos”, disponible: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>
- [5] D. Fernando I. López, “Benchmark para el uso de tecnologías relacionadas a servidores de aplicaciones glassfish y JBoss”, Escuela politécnica nacional, Quito del 2009.
- [6] Página oficial de JBoss, “Servidor JBoss”, disponible en: <http://jbossas.jboss.org>
- [7] Nicolás Tedeschi, “¿Qué es un Patrón de Diseño?”, reporte técnico, publicado el 25 de mayo del 2011.
- [8] M.A. Álvarez (2014, Enero 02), “¿Qué es MVC?”, reporte técnico, publicado el 02 de enero del 2014.
- [9] M. Acosta (2013, Junio 01), “Estudio de patrones de diseño en plataforma Java Enterprise Edition versión 6 para el desarrollo de aplicaciones web”, volumen 01, Nro 01
- [10] A. Otero (2010), “Tutorial básico de java EE”, volumen 1.0
- [11] N. Chisaguano, A. Naranjo, “Sistema de registro de animales y sus derivados del centro experimental Uyunbicho de la facultad de veterinaria de la Universidad Central del Ecuador”, Facultad de Ingeniería, ciencia físicas y matemáticas carrera de ingeniería informática, Quito 2012.
- [12] D. Paul T Barriga, “Análisis comparativo de los servidores Glassfish y JBoss para la plataforma javaee Aplicado al módulo de catálogos del sistema De recursos humanos de la Epoch”, escuela superior politécnica de Chimborazo, Riobamba 2014

- [13] M. Keith, M. Schincariol (New York 2009), “Persistence API”, reporte técnico, publicado en New York del 2009.
- [14] M. Degiovannini (2007 Febrero), “Comparativa de Frameworks WEB”, volumen 1.0, pp. 7–23
- [15] B. Gutierrez, “Introducción al Lenguaje de expresiones”, reporte técnico, publicado en el 2014.
- [16] Página oficial de Seam Security, “Framework Seam”, Disponible en: <http://www.seamframework.org/>
- [17] Picketlink, “Integración de Picketlink”, disponible en: <http://docs.jboss.org/picketlink/2/latest/reference/html/>
- [18] M. Grau, “Introducción a Maven”, reporte técnico, publicado el 26 de Julio del 2011
- [19] Página oficial de Maven, “Apache Maven Project”, disponible en: <http://maven.apache.org>
- [20] Página oficial de Bootstrap, “Bootstrap”, disponible en: <http://getbootstrap.com/>
- [21] J. Díaz, C. Tzancoff, A. Rodríguez, V. Soria, “Usando JMeter para pruebas de rendimiento”, Universidad Nacional de la Plata
- [22] Página oficial Apache JMeter, “JMeter”, Disponible en: <http://jmeter.apache.org/>
- [23] Roberth G. Figueroa, Camilo J. Solís, “Metodologías tradicionales vs. Metodologías ágiles”, Universidad Técnica Particular de Loja 2010.
- [24] J. Canós, P. Letelier, M. Penadés, “Metodologías Agiles en el desarrollo del software”, Universidad Politécnica de Valencia, publicado 2009
- [25] P. Caceras, E. Marcos, “Proceso Ágil para el desarrollo de aplicaciones”, Universidad Rey, Madrid-España 2010.
- [26] Luis M. Echeverry, Luz E. Delgado, “Caso práctico de la metodología ágil XP”, Universidad tecnológica de Pereira, 2009.

- [27] W. Murphy, "Gestión de proyectos Scrum Manager", Reporte técnico publicado en abril del 2014.
- [28] Isaías C. Pérez, Rodrigo P. González, Aureliano D. Rodríguez, "Metodología de desarrollo de software", Reporte técnico, publicado el 15 de octubre del 2008.
- [29] G. Álvarez, A. Garduño, "Metodología Iconix", Publicado el 12 de agosto de 201.
- [30] E. Menéndez, "Información General de Enterprise Architect", reporte técnico, publicado en el 2014.
- [31] Daniel L. Ortega, Jessica S. Villa, "Estudio comparativo de las herramientas case: staruml, Poseidon for UML y Enterprise architect, para el Modelamiento de diagramas uml", Universidad Tecnológica de Pereira 2012.
- [32] Netbeans, "Netbeans 7.4", disponible en: <http://wiki.netbeans.org/NewAndNoteworthyNB74>
- [33] D. Otero, "Iterar sobre listas en JSF, etiqueta ui: repeat y atributo varstatus", reporte técnico, publicado el 25 de octubre del 2015
- [34] D. González, "Responsive web design: diseño multidispositivo para mejorar la experiencia de usuario" Universidad de Barcelona, Junio 19 del 2013
- [35] J. Granda, "Software empresarial para gestión de proyectos", Disponible en: <https://github.com/jlgranda/glue-brms>
- [36] Instituto de Ciencias del Patrimonio (Incipit), Consejo Superior de Investigaciones Científicas (CSIC), "Clase de Cultural Heritage Abstract Reference Model CHARM", reporte técnico, publicado el 2013.
- [37] L. Alegsa, "¿Qué significa Entidad? - Información sobre Entidad", reporte técnico, publicado el 2014
- [38] Departamento de Ciencias de la Computación e Inteligencia Artificial (CCIA) de la Universidad de Granada, "Estructuras de datos", reporte técnico, publicado el 2013.

k. Anexos

Anexo 1: Metodologías de desarrollo de software.

1. Descripción de las metodologías del desarrollo de software.

1.1. Definición de metodología de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos y técnicas que ayudan a los desarrolladores a estructurar, planificar y controlar el proceso de desarrollo de software [24].

El objetivo de utilizar metodologías y técnicas para la construcción de un producto de software, es cumplir con los requerimientos del usuario, por lo general los requisitos para un software son tan variados y cambiantes, que ha dado lugar a que exista una gran variedad de metodologías para la creación del software, clasificándose en dos grandes grupos; la metodología ágil y metodología tradicional [24].

1. Clasificación.

2.1. Metodologías ágiles o ligeras.

Estas metodologías se basan en el desarrollo iterativo e incremental, teniendo presente los cambios y respondiendo a estos mediante la colaboración de un grupo de desarrolladores auto-organizados y multidisciplinares. Utiliza un enfoque basado en el valor para construir software, colaborando con el cliente e incorporando los cambios continuamente [25].

Dentro de las metodologías ágiles se puede mencionar algunas como: XP, Scrum, Crystal Methodologies, Dynamic Systems Development Method. A continuación se describirá sobre las metodologías XP y Scrum:

- **Programación extrema XP (Extreme Programming).**

Consiste en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. XP se basa: en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Esta metodología es adecuada para proyectos con requisitos

imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Dentro de las fases de XP tenemos [26]:

- 1. Exploración:** En esta fase, el cliente plantea a grandes rasgos las historias de usuario (Requisitos) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto ya que se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Esta fase toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología [26].
- 2. Planificación de la Entrega (Release):** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días [26].
- 3. Iteraciones:** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto, esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción [26].
- 4. Producción:** Esta fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase [26].
- 5. Mantenimiento:** En esta fase debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura [26].

6. Muerte del Proyecto: Esta fase se utiliza cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto indica que se satisfacen todas las necesidades del cliente en aspectos como rendimiento y fiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo [26].

- **Scrum**

Es una metodología de desarrollo muy simple, que requiere trabajo duro, porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. Está especialmente indicada para proyectos con un rápido cambio de requisitos y está orientada a las personas más que a los procesos [27]. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [27]. A continuación se mencionan las siguientes fases:

- 1. Revisión de planes de release:** Esta fase se realiza una vez establecido el Release Backlog (prioridades ordenadas según su necesidad y urgencia para la próxima versión) y es llevada a cabo por el equipo a fin de evaluar las diferentes factibilidades de los requerimientos y estimaciones [27].
- 2. Distribución, revisión y ajustes de estándares de producto:** En esta etapa los desarrolladores dejan todo listo para comenzar con la etapa fundamental de Scrum que son los Sprints [27].
- 3. Sprint:** La fase de Sprint es donde el desarrollo de software se lleva a cabo. Un Sprint consta de las siguientes actividades: elaborar, integrar, revisar, ajustar [27].
- 4. Revisión de sprint:** Cada Sprint es seguido por una revisión de Sprint. Durante esta revisión, el software desarrollado en el Sprint anterior se revisa y si es necesario se le añaden nuevos ítems del backlog. El grupo de revisores pueden ser: las partes interesadas del proyecto, gestores, desarrolladores y en ocasiones los clientes, ventas y marketing. Las actividades, y la revisión de Sprint se repiten hasta que el producto

se considera listo para su distribución por los participantes en el proyecto. Luego, el proyecto pasa a la fase de cierre en que el producto se prepara para el lanzamiento y la distribución [27].

- 5. Cierre:** En esta fase tienen lugar las actividades de debugging, marketing y promoción. Al acabar esta fase el proyecto quedará cerrado [27].

2.2. Metodologías tradicionales o pesadas.

Se centran en la definición detallada de los procesos, tareas a realizar y herramientas a utilizar, requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto más grande es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear para el desarrollo de un software en una organización [25]. Entre las metodologías tradicionales se puede mencionar algunas como: Iconix, MSF (Microsoft Solution Framework), Win-Win Spiral Model, RUP. A continuación se describirá las metodologías RUP e Iconix.

- **RUP (Rational Unified Procces)**

El Proceso Racional Unificado es un proceso de desarrollo de software y junto con el Lenguaje de modelado unificado (UML). Proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo de software. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos [28]. Se basa en un proceso dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación, prueba) que parten de los casos de uso; está centrado en la arquitectura y es iterativo e incremental. Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software. Dentro de las fases de esta metodología tenemos [28]:

- 1. Concepción:** Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una

visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores [28].

2. **Elaboración:** En la fase de elaboración seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar [28].
 3. **Construcción:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto [28].
 4. **Transición:** Esta fase se centra en asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto [28].
- **Iconix.**

Se encuentra entre la complejidad del RUP (Rational Unified Processes) y la simplicidad de XP (Extreme Programming). Presenta 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 [29]. Iconix es un proceso de modelado de objetos basado en casos de uso. Toma ideas de otros modelos como el Proceso Unificado de Rational (RUP) y Programación Extrema (XP) desarrollo ágil de software, aunque presenta algunas diferencias: es más liviano que RUP porque utiliza solo cuatro diagramas del UML y a diferencia del XP, provee de suficiente documentación de requerimientos y de diseño. Además esta metodología se caracteriza por ser iterativa e incremental, por su trazabilidad y uso dinámico de UML [29]. Dentro de las fases tenemos:

1. **Análisis de requerimientos:** Esta fase inicial tiene como objetivo principal realizar la revisión de requerimientos del problema, es decir identificar en el mundo real, los objetos y todas las relaciones de agregación y generalización entre ellos. Se deben analizar todos los requisitos que formarán parte del sistema y con estos construir el

diagrama de clases, que representa las agrupaciones funcionales que estructurarán el sistema en desarrollo. Para ello se utilizan 3 diagramas: modelo del dominio, modelo de casos de uso, prototipo de interfaz de usuario [29].

- 2. Análisis y diseño preliminar:** Se deben describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción. La principal sugerencia de Iconix, en esta actividad es que no se debe perder mucho tiempo con la descripción textual. Debería usarse un estilo consistente que sea adecuado al contexto del proyecto. Además se debe realizar el diagrama de robustez que permita capturar el que hacer y a partir de eso él como hacerlo, facilitando el reconocimiento e identificación de objetos que participan en cada caso de uso y una lectura sencilla del sistema [29].
- 3. Diseño detalle:** En esta fase se reconocen todos los elementos que forman parte de nuestro sistema. En los diagramas de secuencia se muestra: los métodos que llevarán las clases de nuestro sistema y todos los cursos alternos que pueden tomar los casos de uso. Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases y verificar si el diseño satisface todos los requisitos identificados [29].
- 4. Implementación:** En esta fase a partir del buen diseño logrado se creará el software; que posteriormente se entregará. Se debe utilizar el diagrama de componentes si fuera necesario para apoyar el desarrollo, es decir mostrar una distribución física de los elementos que componen la estructura interna del sistema. Así como escribir y generar el código, pero además se debe tener en cuenta factores como: reusabilidad, extensibilidad, confiabilidad, realizar pruebas [29].

Anexo 2: Tecnologías y herramientas.

1. Descripción de la tecnología a utilizar.

1.1. Enterprise Architect

Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, que cubre el desarrollo de software desde la captura de requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta de multi-usuarios, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Además, ofrece salida de documentación flexible y de alta calidad [30].

Como se puede observar en la tabla XLV, EA tiene soporte para los 13 diagramas de UML 2 y más [30].

TABLA XLV.
DIAGRAMA DE UML [31]

Diagramas Estructurales	Diagramas de Comportamiento	Extendidos
Clase	Casos de Uso	Análisis (actividad simple)
Objeto	Comunicación	
Compuesto	Secuencia	
Paquete	Interacción	
Componente	Actividades	Personalizado (para requisitos, cambios, UI)
Despliegue	Estado	
	Tiempo	

Además EA soporta la generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic, ActionScript y PHP [31].

1.2. Entorno de desarrollo.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, el cual consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Además ofrecen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como: C++, Python, Java, C#, Delphi, Visual Basic, etc.

Existen diversos IDEs para desarrollar programas entre ellos tenemos: Eclipse. Netbeans, Geany, CodeRun [32]. A continuación se centrara en el IDE Netbeans.

Es un entorno de desarrollo integrado (IDE), escrito en el lenguaje de programación Java, es un programa gratuito y de código abierto, se encuentra disponible en inglés, portugués, japonés, ruso y chino simplificado [32]. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web, o para dispositivos móviles [4]. Da soporte a diferentes tecnologías, entre ellas: Java, PHP, Groovy, C/C++, HTML5, entre otras. Además puede instalarse en varios sistemas operativos como: Windows, Linux, Mac OS, etc. Se considera un IDE ideal para trabajar con el lenguaje de desarrollo Java y (y todos sus derivados) [32].

Características principales [32]:

- Editor de código (multilenguaje).
- Simplifica la gestión de grandes proyectos.
- Herramientas para la depuración de errores.
- Permite la optimización de código.
- Permite conectarse a distintos sistemas gestores de bases de datos.
- Se integra con diversos servidores de aplicaciones.

La versión en la cual se va a trabajar para el presente trabajo de titulación es la 7.4, ya que permite mejorar el soporte de HTML 5 y su integración con Java EE, Maven, entre otras tecnologías.

1.3. Patrones de diseño

Para la construcción de aplicaciones informáticas se toma en cuenta dos aspectos fundamentales: el cómo construir aplicaciones en menos tiempo y el utilizar estándares que permitan mayor reutilización de código, más calidad, mejor mantenibilidad. Teniendo en cuenta la importancia que tiene hoy en día este tipo de aplicaciones se ha tomado en cuenta el uso de patrones internacionalmente aceptados [7]. A continuación se hará un análisis sobre el tipo de patrón a utilizar en el presente trabajo de titulación.

Definición.- Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Entre los cuales podemos nombrar [7]:

- Patrones Creacionales: inicialización y configuración de objetos.
- Patrones Estructurales: separan la interfaz de la implementación. Se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes.
- Patrones de Comportamiento: más que describir objetos o clases, describen la comunicación entre ellos.

Modelo Vista Controlador.-

El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista [8].

Este modelo de arquitectura presenta varias ventajas [8]:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones API (Application Programming Interface) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el modelo y sus vistas dinámicas; se produce en tiempo de ejecución, no en tiempo de compilación.

Elementos del patrón.

El modelo es el responsable de [8]:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define reglas de negocio (la funcionalidad del sistema).
- Notificará a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo activo.

El controlador es responsable de [8]:

- Recibir los eventos de entrada.
- Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta)".

Las vistas son responsables de [8]:

- Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes.

El diagrama de secuencia que se muestra en la figura 52 se ilustra la interrelación de los elementos del patrón.

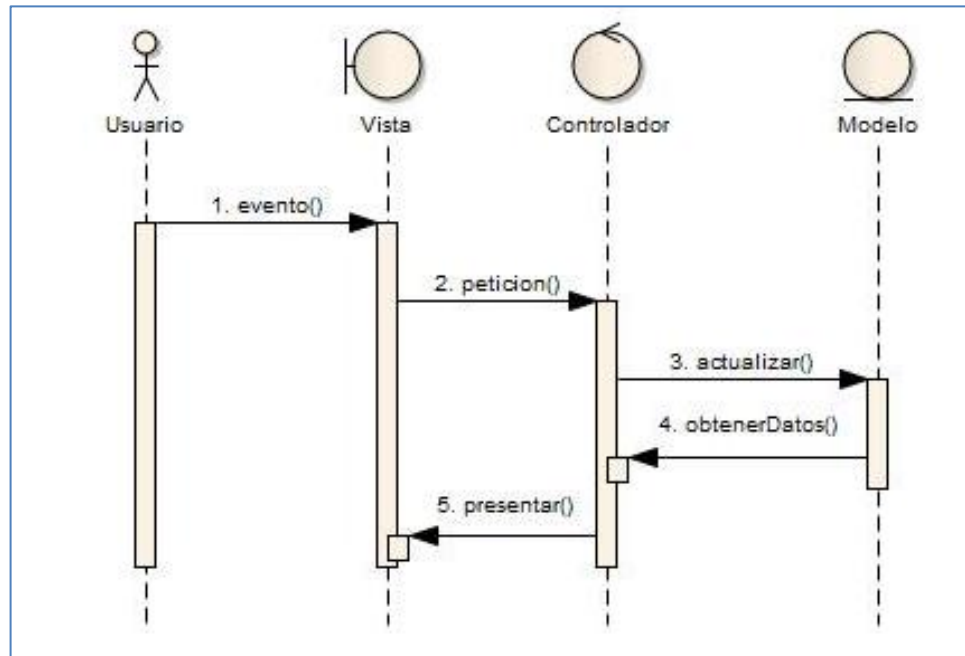


Figura 52. Ejemplo de construcción de diagrama de secuencia.

Pasos:

1. El usuario introduce el evento.
2. El Controlador recibe el evento y lo traduce en una petición al Modelo (aunque también puede llamar directamente a la vista).
3. El modelo (si es necesario) llama a la vista para su actualización.
4. Para cumplir con la actualización la Vista puede solicitar datos al Modelo.
5. El Controlador recibe el control.

Frameworks MVC.

Los patrones MVC se adaptan perfectamente a cualquier objetivo particular de cualquier framework ya que proveer una estructura bien definida que da soporte a un proyecto Web y sobre todo lo mantiene organizado.

Ventajas de los Frameworks MVC.

Los frameworks ofrecen una elevada organización en el trabajo de desarrollo de aplicaciones de software, ya que todo parece tener un sitio, aunque siempre existen cosas que son difíciles de acomodar, pero generalmente se obtiene mucha más organización

que cuando se hace el layout de carpetas y la organización de los archivos manualmente. Generalmente estos frameworks poseen generadores que crean los archivos base de los modelos o vistas, para no tener que crear cada archivo relacionado a mano.

Los frameworks MVC tienen como desventaja o limitación que el manejo de flujos de tareas tiene que hacerse a mano, o sea, codificando los flujos directamente.

Algunos de los frameworks MVC que se han desarrollado en base a este patrón de diseño tenemos [8]:

- **Java Server Faces (JSF)**

Es un framework de aplicaciones Java basadas en la Web, que simplifica el desarrollo de interfaces de usuarios en aplicaciones Java EE. JSF usa una tecnología como XUL (acrónimo de XML-based User_interface Language basado en xml para interfaces de usuario) [8].

Posee un controlador central (FrontController) que se encarga de manejar todas las peticiones del cliente y gestionar su ciclo de vida. Está basado en un modelo de componentes para la interfaz de usuario. Un componente JSF es un elemento reusable y configurable que se puede utilizar en la interfaz de usuario [8].

Los componentes se pueden anidar. Por ejemplo, una página contiene un componente mapa que a su vez contiene un componente botón. El diseño del framework permite navegar a través del árbol de componentes para acceder a sus propiedades. Además, los componentes pueden reaccionar a diferentes eventos y son capaces de almacenar su estado interno [8].

JSF dentro de su especificación incluye [8]:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejando eventos, validación de entradas, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad. Un conjunto por defecto de componentes de interfaz de usuario.
- Dos bibliotecas de etiqueta personalizadas para XUL que permite expresar una interfaz de usuario.

- Un modelo de eventos en el lado del servidor.
- Administración de estados.
- Beans administrados
- Beneficios de la tecnología Java Server Faces
- Componentes personalizados: Java Server Faces permite combinar fácilmente complejas GUIs en un componente.
- Mejor soporte para Ajax: Una gran parte de librerías de componentes dan soporte Ajax
- Ofrecer soporte para otras tecnologías display: Java Server Faces no está limitado a HTML y HTTP.
- Acceso a beans por nombre: Java Server Faces permite asignar nombres a beans, con lo que se puede referenciar en los formularios.
- Lenguaje de expresión: El lenguaje de expresión de Java Server Faces es más conciso y potente.
- Controladores y definiciones de beans más simples: Java Server Faces no exige que tu controlador y las clases beans sean extendidas a algunas clases padre en particular o usen algún método en particular.
- Herramientas más potentes: Java Server Faces ofrece controles GUI y manejadores que facilitan el uso de IDEs de arrastrar y soltar.

Lenguaje de Expresiones (EL)

En ocasiones en el desarrollo de presentación de una aplicación necesitamos hacer validaciones en la presentación, como por ejemplo dadas unas condiciones XY renderice un botón o muestre un link, etc. para este tipo de cosas debemos usar valores booleanos 'true' or 'false' con el tag render [15].

Algunas veces no es suficiente una variable booleana para decidir este tipo de cosa, es necesario hacer comparaciones lógicas que den como resultado un booleano. Para ello se hace uso del Lenguaje de Expresiones (EL), que permite facilitar la visualización y recogida de los datos de los formularios JSF introduce un lenguaje de expresiones similar al introducido en JSP. De hecho a partir de JSF 1.2, ambos lenguajes de expresiones se han unificado [15].

EL es un lenguaje para acceder a datos de varias fuentes en tiempo de ejecución. Su sintaxis es considerablemente más amigable que la de Java, que es el único lenguaje soportado directamente por la especificación JSP 1.2. Todas las acciones JSTL reconocen expresiones EL, en sus valores de atributos, y se podrían desarrollar acciones personalizadas para que hicieran lo mismo. Se espera que EL sea incorporado dentro de la próxima versión de la especificación JSP para mejorar su uso para acceder a datos sobre el lenguaje Java. Si es así, podremos usar expresiones EL en un valor de atributo de una acción, o incluso en una plantilla de texto [15].

Si has usado JavaScript deberías sentirte como en casa con EL. EL toma prestada de JavaScript la sintaxis para acceder a estructuras de datos tanto como propiedades de un objeto (con el operador.) como con elementos con nombres de un array (con el operador ["nombre"]). Las propiedades de los componentes JavaBeans y las entradas `java.util.Map`, que usan la clave como nombre de propiedad, pueden ser accedidas de esta forma.

Como se ve aquí, una expresión EL siempre debe estar encerrada entre los caracteres `${y}` `$`. Las dos primeras expresiones acceden a una propiedad llamada `myProperty` en un objeto representado por una variable llamada `myObj`. La tercera expresión acceder a una propiedad con un nombre contenido en un variable, esta sintaxis se puede utilizar con cualquier expresión que evalúe el nombre de la propiedad [33]. El operador de acceso a array también se usa para datos representados como una colección de elementos indexados, como un array Java o una `java.util.List` [33].

Además de los operadores de propiedad y elemento array y los operadores aritméticos, relacionales, y lógicos, hay un operador especial para comprobar si un objeto está "vacío" o no puede ser usado en una expresión EL [33].

- **Procesamiento Condicional y Bucles**

Veamos algunos ejemplos de cómo podemos usar el JSTL condicional y las acciones de iteración: `<c: if>`; `<c: choose>`, `<c: when>`, y un triple `<c: otherwise>`; y `<c: forEach>`. Por el camino, también usaremos acciones de salida básica y de selección de variables: `<c: out>` y `<c: set>` [15].

<c: if> permite incluir, o procesar, condicionalmente una parte de una página, dependiendo de la información durante la ejecución. El valor del atributo test chequea si la cookie existe. El operador empty combinado con el operador "not" (!) significa que evalúa a true si el cookie no existe, haciendo que el cuerpo del elemento sea procesado. Dentro del cuerpo, la acción <c: out>añade el valor de la cookie a la respuesta [15].

<c: forEach> procesa su cuerpo una vez por cada elemento de la colección especificado por el atributo var [15].

<c: set> selecciona una variable con el valor especificado por el atributo value; que puede ser un valor estático, como en este ejemplo, o una expresión EL. También podemos especificar el ámbito de la variable con el atributo scope (page, request, session o application) [15].

<c: choose> agrupa una o más acciones <c: when>, cada una especificando una condición booleana diferente. La acción <c: choose> chequea cada condición en orden y sólo permite la primera acción <c: choose> con una condición que se evalúe a true para procesar su cuerpo. El cuerpo <c: choose> también podría contener un <c: otherwise>. Su cuerpo sólo se procesa si ninguna de las condiciones de los <c: when> es true [15].

<Ui: repeat> permite hacer un "bucle for" para repetir 'n' veces lo que necesitemos en la vista de nuestro sitio Web [15].

1.4. Java/Javaee6

Java Enterprise Edición 6, es una especificación del lenguaje Java para el desarrollo de aplicaciones empresariales mediante el uso del IDE Netbeans que a partir de la versión 7.0 soporta todas las funcionalidades de Java EE para poder realizarlo de forma rápida y sencilla. Algunas de las características nuevas de la plataforma son [9]:

- **Perfiles:** Java EE 6 introduce el concepto de perfiles: distintas configuraciones específicas de la plataforma para distintos fines. El primero en ser introducido en esta versión es el Web Profile, un perfil más liviano que usa únicamente lo necesario para desarrollos Web [9].
- **Productividad del desarrollo:** apuntando las configuraciones a más anotaciones y menos configuración XML [9].
- **Nuevas especificaciones:** EJB 3.1, Java API for RESTful Web Services (JAX-RS), Contextos e inyección de dependencias (CDI), JPA 2, etc [9].

Cuenta con una capa de aplicación (o negocio) implementada con componentes EJB (Enterprise Java Bean) que hace uso de JPA (Java Persistence API) como mecanismo de mapeo Objeto/Relacional para proveer de una capa de acceso a la base de datos [10].

Contenedores.- Un contenedor es un proceso donde se ejecutan los componentes [10]:

- Gestiona los componentes de la aplicación.- Ciclo de vida
- Proporciona acceso a servicios de la plataforma.- Seguridad, transacciones, persistencia, conectividad, etc.

Los componentes se despliegan y ejecutan en contenedores especializados como [10]:

- Contenedor de applets en un navegador Web
- Contenedor Web Tomcat
- Contenedor de EJBs

Componentes web en Java EE

- **Java Servlets.**- Clases escritas en Java que procesan peticiones HTTP y construyen respuestas [10].
- **Java Server Pages (JSP).**- Documentos basados en texto que contienen dos tipos de texto
 - Una plantilla de datos estática que puede expresarse en un formato como (X)HTML o XML

- Elementos JSP que determinan cómo la página construye el contenido dinámico [10].
- **Java Server Faces (JSF).**- Componentes de interfaz de usuario para aplicaciones Web [10].
- **Los clientes en Java EE son.-**
 - Clientes Web: navegadores Web, páginas Web, applets
 - Aplicaciones de cliente [10].

Componentes de negocio en Java EE [4]

Lógica que resuelve las necesidades de un determinado dominio de aplicación [10]:

- **Enterprise beans (EJBs)**
 - Pueden procesar datos recibidos del lado cliente y enviarlos al nivel de sistema de información para su almacenamiento [10].
 - Pueden recuperar datos del sistema de información, procesarlos y enviarlos al cliente [10].
- **Tipos de EJBs**
 - Bean de sesión.- Una conversación con un cliente [10].
 - Bean dirigido por mensajes.-Permite que un componente de negocio pueda recibir mensajes asíncronamente, normalmente con el Java Message Service (JMS) [10].

Además se hace uso del API Criterios para crear consultas, que se utiliza para definir las consultas de las entidades y su estado persistente mediante la creación de objetos de consulta que define. Criteria query se escriben utilizando los API del lenguaje de programación Java. Tales consultas funcionan independientemente del almacén de datos subyacente [10].

1.5. Framework Maven

Maven es un “Project Management Framework”, es decir, un Framework de gestión de proyectos de software, que ayuda en el proceso de creación de un proyecto en Java. Mediante el uso de los arquetipos, que son estructuras de proyecto predefinidas y

reconocidas en un archivo xml (Pom.xml), se generan mediante los plugins de Maven (Cada plugin de arquetipo relacionado a un jar necesario) [18].

Los principales proyectos OpenSource en Java, están utilizando cada día más el uso de Maven para su desarrollo, por lo que cada vez es más frecuente encontrar plugins de arquetipos para el desarrollo de aplicaciones [18].

Maven da soluciones a tareas que abarcan desde la compilación hasta la distribución, despliegue y documentación de los proyectos. Se podría describir como “un sistema de estándares, un repositorio, y un software usado para manejar y describir proyectos [18].

1.6. Servidor de aplicaciones

Un servidor de aplicaciones, es un servidor en una red de computadores que ejecuta ciertas aplicaciones, permite gestionar la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación [5]. Los principales beneficios de los servidores de aplicaciones son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones.

Un servidor Web es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol), se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones y responde a estas peticiones adecuadamente, mediante una página Web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error [5].

Características de los servidores de aplicaciones:

- **La alta disponibilidad:** Hace referencia a que un sistema debe estar funcionando las 24 horas del día los 365 días del año. Para poder alcanzar esta característica es necesario el uso de técnicas de balanceo de carga y de recuperación ante fallos [5].
- **La escalabilidad:** Es la capacidad de hacer crecer un sistema cuando se incrementa la carga de trabajo. Cada máquina tiene una capacidad finita de recursos y por lo tanto solo puede servir un número limitado de peticiones. Si, por ejemplo, se tiene una tienda que incrementa la demanda de servicio, se debe ser capaces de incorporar nuevas máquinas para dar servicio [5].

- **El mantenimiento:** Tiene que ver con la versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares [5].

Entre los servidores de aplicaciones encontramos: Apache, Bea WebLogic, Glassfish, JBoss. A continuación se detalla sobre el servidor que se utilizará en la aplicación a implementar [5].

JBoss es un servidor de aplicaciones Java EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java [5].

1.7. Base de datos relacional.

Una base de datos es una colección integrada de registros o archivos lógicamente relacionados consolidados en una unidad que provee datos para usos múltiples, se pueden clasificar de acuerdo a su modelo de administración de datos, entre ellos están el modelo plano, modelo jerárquico, modelo de red, modelo transaccionales, modelo relacional, entre otras [4].

Un modelo de datos es básicamente una descripción de algo conocido como contenedor de datos, es decir algo en donde se guarda la información. El modelo de datos más utilizado en la actualidad para implementar base de datos es el modelo relacional, que permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas [4].

Para manipular la información se utiliza un lenguaje relacional, actualmente es más común realizar las consultas a bases de datos relacionales es SQL (Structured Query Language), un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales integradas [4].

Existe software exclusivamente dedicado a tratar con bases de datos relacionales. Este software se conoce como SGBD (Sistema de Gestión de Base de Datos relacional) o RDBMS (del inglés Relational Database Management System). Entre los gestores o

manejadores actuales encontramos: MySQL, PostgreSQL, Oracle, DB2, Microsoft SQL Server, entre otros [4].

A continuación se centrará en PostgreSQL, el cual se utilizara la versión 9.1 como base de datos para el desarrollo del presenta trabajo de titulación.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Esta base de datos utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando [4].

Para conectarse a una base de datos, consultarla o actualizarla usando SQL, se utiliza el puente JDBC el cual es un componente de software que permite a las aplicaciones escritas en Java interactuar con una base de datos utilizando diversos controladores (conocidos también como drivers) y un conjunto de objetos y métodos de la API (Interfaz de Programación de Aplicaciones) de Java. Para conectar con bases de datos individuales, JDBC requiere controladores para cada base de datos. El controlador JDBC ofrece la conexión a la base de datos e implementa el protocolo para la transferencia de las consultas y resultados entre el cliente (aplicación) y la base de datos [4].

Arquitectura

En la figura 53, se observa una aplicación front-end que utiliza la API de Java para interactuar con el Driver Manager JDBC. El Driver Manager JDBC utiliza la API de Driver JDBC para cargar el driver apropiado de JDBC. Los drivers JDBC, los cuales están disponibles por los proveedores de las diferentes bases de datos del mercado, se comunican con la capa subyacente del DBMS (Database Manager System, Sistema Manejador de Bases de Datos [4].

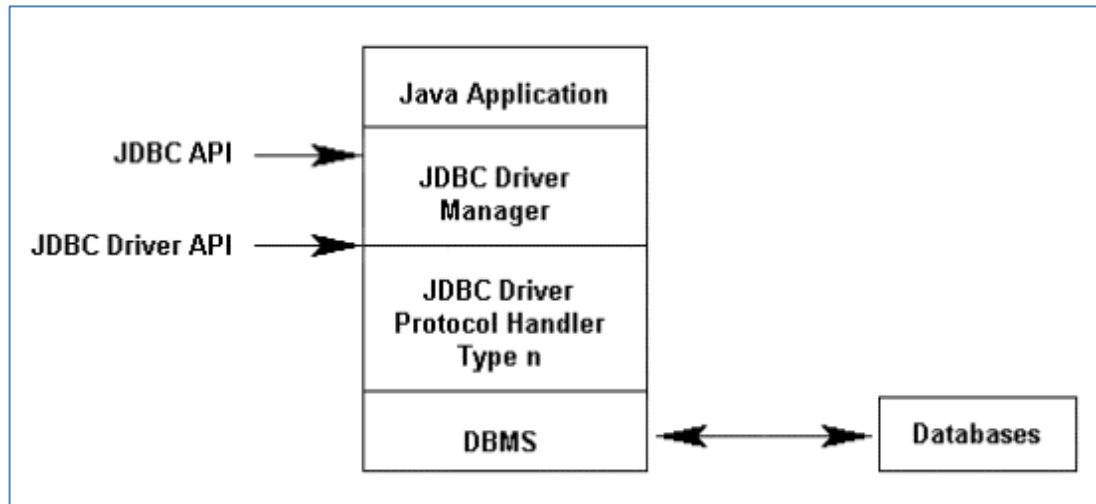


Figura 53. Arquitectura de JDBC [4].

Funciones [4]

- Básicamente el API JDBC hace posible la realización de las siguientes tareas:
- Establecer una conexión con una base de datos.
- Enviar sentencias SQL.
- Manipular los datos.
- Procesar los resultados de la ejecución de las sentencias
- Usando JDBC es fácil enviar sentencias SQL a virtualmente cualquier base de datos relacional.

1.8. Interfaz de persistencia Java (JPA).

JPA proporciona un modelo de persistencia basado en POJOs para mapear bases de datos relacionales en Java. La persistencia de Java fue desarrollada por expertos de EJB 3.0 como parte de JSR 220, aunque su uso no se limita a los componentes software EJB. Se puede utilizar en aplicaciones Web y aplicaciones clientes [12].

Para ello, combina ideas y conceptos de los principales frameworks de persistencia, como Hibernate, Toplink y JDO. El mapeo objeto-relacional (es decir, la relación entre entidades Java y tablas de la base de datos, query con nombre, etc) se realiza mediante anotaciones en las propias clases de entidad [12].

Antes de seguir con JPA se debe tener claro el concepto de "persistencia". La persistencia o el almacenamiento permanente, es una de las necesidades básicas de cualquier sistema de información de cualquier tipo [12].

Persistir objetos Java en una base de datos relacional implica serializar un árbol de objetos Java en una base de datos de estructura tabular y viceversa. Es esencial mapear objetos Java para optimizar velocidad y eficiencia de la base de datos [12].

La unidad de persistencia define un conjunto de todas las entidades (clases) que son gestionadas por la instancia del EntityManager en una aplicación. Este conjunto de clases de entidades representa los datos contenidos en una única Base de Datos. Las unidades de persistencia se definen en el fichero de configuración [12].

1.9. Enterprise Java Beans (EJB)

Es un modelo de programación que permite construir aplicaciones Java mediante objetos simples (POJO). Cuando se construye una aplicación, son muchas las responsabilidades que se deben tener en cuenta, como la seguridad, transaccionabilidad, concurrencia, etc. El estándar EJB permite centrarse en el código de la lógica de negocio del problema que se desea solucionar y dejar el resto de responsabilidades al contenedor de aplicaciones donde se ejecutará la aplicación [11].

Es parte de la plataforma Java EE 6 y este provee diversas APIs para la construcción de aplicaciones empresariales, entre ellas EJB, JPA, JMS y JAX-WS. Cada una de ellas se centra en un área específica, resolviendo problemas concretos. Además, cada especificación con su respectiva API está preparada para funcionar en compañía de las demás de forma nativa y por tanto en su conjunto son una solución perfectamente válida para desarrollar una aplicación end-to-end [11].

El uso de POJOs para encapsular la lógica de negocio proporciona un modelo simple que es altamente reutilizable. Se debe tener en cuenta que un POJO no actuará como un componente EJB hasta que haya sido empaquetado y desplegado en un contenedor EJB y accedido a dicho contenedor. Una vez que un POJO sea definido como EJB y este haya sido desplegado en el contenedor, se convertirá en uno de los tres siguientes [11]:

Session Beans o Beans de Sesión: Son los componentes que contienen la lógica de negocio que requiere los clientes de la aplicación. Son accedidas a través de un proxy tras una solicitud al contenedor. Tras dicha solicitud, el cliente tiene una vista del Session Bean, pero no el Session Bean real. Esto permite al contenedor realizar ciertas operaciones sobre el Session Bean real de forma transparente para el cliente. Los componentes Session Bean pueden ser de tres tipos: Stateless Session Beans (SLSB), Stateful Session Beans (SFSB), Singleton [11].

Message-Drive Bean: Son componentes de tipo listener que actúan de forma asincrónica. Su misión es la de consumir mensajes, los cuales pueden gestionar directamente o enviar a otro componente. Los Message-Drive Bean actúan sobre un proveedor de mensajería por ejemplo Java Messaging System. Al igual que los Stateless Session Beans, los Message-Drive Beans no mantienen estado entre invocaciones [11].

Entity Bean: Son representaciones de datos almacenados en una base de datos, siempre en forma de POJOs. El encargado de gestionar los Entity Beans es EntityManager, un servicio que es suministrado por el contenedor y que está incluido en la especificación Java Persistence API. JPA es parte de EJB desde la versión 3.0 [11].

1.10. Bootstrap

Permite crear interfaces Web con CSS y JavaScript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design [20].

Un caso de éxito de la utilización de Bootstrap es Twitter, pero existen otros sitios que implementan diseños Responsive basados en este framework como por ejemplo [20]:

- Breezi: <http://breezi.com/>
- Stackideas: <http://stackideas.com/>
- iAcquire: <http://www.iacquire.com/>
- AudioMack: <http://www.audiomack.com/>
- LearnDot: <http://www.learndot.com/>

Funcionalidades [13]:

- Interfaces que funcionen de manera brillante en los navegadores actuales, y correcta en los no tan actuales.
- Un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones.
- Una mejor integración con las librerías que sueles usar habitualmente, como por ejemplo jQuery.
- Un diseño sólido basado en herramientas actuales y potentes como LESS o estándares como CSS3/HTML5.

1.11. Responsive web design.

Responsive web design o diseño Web adaptable o adaptativo en español, es una técnica de diseño Web en la que se utiliza una estructura fluida. Esto hace posible que el contenido de la página se adapte en función de la resolución de la pantalla del visitante, permitiéndole navegar de una forma más cómoda por la Web independientemente de si se accede desde un ordenador, un tablet o un móvil, sin tener que hacer zoom continuamente [34].

Para aplicar el Responsive web design en una Web se utilizan las siguientes técnicas [34]:

- Una **estructura fluida cuyas medidas se expresan en unidades relativas (%)** en vez de en unidades absolutas (px o pt), esto permite al contenido adaptarse automáticamente [34].
- **Dimensiones de las imágenes expresadas en unidades relativas (%)** para que no rompan el diseño de la página y se redimensionen correctamente [34].
- **Declaración de la meta:** viewport <meta name= "viewport" content="width=device-width, initial-scale=1, maximum-scale=1">, para que los móviles y tablets escalen la Web correctamente a la pantalla del dispositivo> [34].
- **CSS media query** permiten a los diseñadores establecer unas propiedades (color, ancho, si es o no visible, etc) u otras sobre un determinado elemento (imagen, texto, menú, etc) de la página dependiendo de la resolución de la pantalla del usuario [34].
- No es estrictamente necesario, pero en determinados diseños es necesario utilizar JavaScript para cambiar el comportamiento de los elementos de la página, para ello

se utiliza el evento `window.onresize` que permite ejecutar código cuando la ventana del navegador cambia de tamaño [34].

1.12. Framework Picketlink.

Picketlink es un proyecto global para la seguridad y la gestión de identidades para aplicaciones Java. Es un proyecto importante en virtud de las ofertas de seguridad de JBoss [17].

Picketlink es un marco de seguridad de aplicaciones para las aplicaciones Java EE. Proporciona funciones para la autenticación de los usuarios, garantizando el acceso a los métodos de negocio de la aplicación, gestión de usuarios, grupos, roles y permisos de la aplicación, y mucho más. El siguiente diagrama presenta una visión general de alto nivel de los módulos Picketlink y las principales características proporcionadas por los módulos [17]:

Módulos:

- **Módulo de Base**

El módulo base proporciona el marco de integración necesario para utilizar Picketlink dentro de una aplicación Java EE. Se define una API de autenticación flexible que permite a los mecanismos de autenticación conectables a configurarse fácil, con una política de autenticación por defecto razonable que delega en el subsistema de gestión de identidades. Se proporciona un seguimiento de autenticación de sesión de ámbito de las aplicaciones Web y otros clientes de los períodos de sesiones capaces, además de un SPI permisos personalizables que es compatible con una gama flexible de los mecanismos de autorización para la seguridad a nivel de objeto [17].

Las bibliotecas de módulos de base son los siguientes [17]:

- `picketlink-api` - API para el módulo de base de Picketlink.
- `picketlink-impl` - clases de implementación internos para la API de base.

- **Gestión de Identidad**

El módulo de gestión de identidad define el modelo de identidad de base; una colección de interfaces y clases que representan las construcciones de identidad (por ejemplo, usuarios, grupos y roles) utiliza a lo largo Picketlink (consulte el capítulo Administración de identidades para más detalles). Como tal, es un módulo requerido y siempre debe ser incluido en cualquier implementación de aplicaciones que utilizan Picketlink para la seguridad. También proporciona una API uniforme para la gestión de los objetos de identidad dentro de la aplicación [17].

Las bibliotecas son como sigue [17]:

- picketlink-IDM-api - Gestión de Identidad de Picketlink (IDM) de la API. Esta biblioteca se define el modelo de identidad fundamental para todos Picketlink, y todas las interfaces relacionadas con la gestión de identidad [17].
- picketlink-IDM-impl - clases de implementación internos para la API de Corral [17].

Anexo 3: Especificación de requerimientos del software.

1. Fase especificación.

1.1. Introducción

La presente especificación de requisitos de Software del trabajo de titulación denominado “Software para la gestión de información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas”, se realiza con el objetivo de analizar y entender los requerimientos de usuario que servirá para el desarrollo del mismo. Se describe en forma detallada siguiendo el estándar IEEE 830 de Ingeniería de Software.

1.2. Propósito.

- Determinar un acuerdo con el cliente sobre lo que va a permitir realizar el Software.
- Establecer las bases del desarrollo de Software, que describirá cuáles son los requisitos que debe cumplir el producto que se desarrollará.

1.3. Ámbito del sistema.

El sistema a desarrollar permitirá la gestión de información del estado en que se encuentran los vehículos del Ministerio de Transporte y Obras Públicas (MTO). Constará de métodos para el ingreso y almacenamiento de información con la que trabaja el departamento de mantenimiento vial del MTO para el control del estado de los vehículos y se obtendrá como resultado reportes mensuales sobre el estado general de los vehículos.

1.4. Personal involucrado

Para el desarrollo del presente trabajo de titulación, intervendrá el siguiente personal:

TABLA XLVI.
DATOS PERSONALES PERSONA 1.

Nombres:	Yesica Norelly
Apellidos:	Saquicela Celi
Dirección:	Loja: Celi Román
Teléfono:	2570942/ 0939141680
E-mail:	ynsaquicelac@unl.edu.ec

Instrucción:	Superior.
Rol:	Analista/ Desarrollador
Cargo:	Analista, Desarrollador
Responsabilidades:	Obtener requerimientos, analizarlos e interpretarlos para su implementación.

TABLA XLVII.

DATOS PERSONALES PERSONA 2.

Nombres:	Carla Lisbeth
Apellidos:	Romero CCórdova
Dirección:	Loja: Ciudad victoria
Teléfono:	2326003/ 0985537193
E-mail:	clromeroc@unl.edu.ec
Instrucción:	Superior.
Rol:	Analista/ Desarrollador
Cargo:	Analista, Desarrollador
Responsabilidades:	Obtener requerimientos, analizarlos e interpretarlos para su implementación.

1.5. Definición de acrónimos y abreviaturas.

TABLA XLVIII.

DEFINICIONES.

Gestión:	Insertar, Modificar, Buscar, Dar de baja y Listar la información almacenada en el sistema.
Base de Datos:	Una base de datos es un conjunto de información estructurada organizada en registros y almacenada en un soporte electrónico legible desde un ordenador.[1]
IEEE:	Instituto de ingenieros eléctricos y electrónicos.
IEEE 830-1998:	Estándar que sirve para realizar el documento de especificación de requerimientos de software, la última versión

	es la del año 1998. [2]
PostgreSQL:	Es un sistema de gestión de bases de datos relacional de licenciamiento libre.[3]
Requisición:	Documento elaborado con el fin de notificar la adquisición de un producto de reparación de un vehículo [4].
Solicitud de Reparación y Mantenimiento:	Documento elaborado con el fin de informar que parte del vehículo necesita reparación o mantenimiento [5].
Partida de contabilidad:	Serie numérica que sirve para identificar los productos, reportes e información y se utilizan para poder realizar una requisición [6].

TABLA XLIX.
ACRÓNIMOS.

ERS:	Especificación de Requerimientos de Software.
JPA:	API de persistencia de Java.

TABLA L.
ABREVIATURAS

SW:	Software
Ing:	Ingeniero(a)

1.6. Referencias

TABLA LI.
REFERENCIAS.

Referencia	Título	Ruta	Fecha	Autor
1	Bases de Datos	http://docencia.lbd.udc.es	2011	Luis

	documentales: Estructura y Principios de Uso	/bdd-grao/teoria/tema1/1.3-IntroduccionALasBDsDocumentales.pdf		Rodríguez Yunta.
2	Especificación de Requisitos según el Estándar de IEEE 830-1998	https://www.fdi.ucm.es/pr ofesor/gmendez/docs/is0809/ieee830.pdf	22/10/2008	G. Méndez .
3	Postgresql vs. MySQL	http://danielpecos.com/documents/postgresql-vs-mysql/#AEN59	2013	Daniel Pecos Martínez
4	Investigación de Operaciones: orden de compra y requisición de compra.	http://elymon.blogspot.com/2008/05/orden-de-compra-y-requisicion-de-compra.html .	22/05/2008	M. Gasga, Elizabeth Martínez
5	Reparación y mantenimiento de vehículos	http://www.secogem.gob.mx/gestiondecalidad/documentos/contenido_sistema/PROCED_REP_MTT_O_VEHICULOS_VER02.pdf	16/02/2012	Secretaría de la Contraloría Coordinación de Administración de México
6	Partida	http://azure.afi.es/Content/Web/EmpresasUnicaja/paises/herramientas/glosar	06/05/2014	Unicaja Banco, S. A de

		<u>io/listado sidN 1052251</u> <u>sid2N 1052252 cidIL 55</u> <u>7531 utN 1.aspx?txtS=P</u>		España
--	--	--	--	--------

1.7. Visión general del documento.

En el presente documento de especificación de requisitos consta de las siguientes partes:

- **Introducción.**- en donde se menciona los objetivos que tiene el sistema que se va a desarrollar.
- **Descripción general.**- se describe a perspectiva general del sistema, los usuarios que utilizaran el sistema con sus características, así como las limitaciones de estos.
- **Requerimientos específicos.**- se detalla los requerimientos de del producto final.

2. Descripción.

2.1. Perspectiva del producto.

El presente software es para la gestión de la información de los vehículos en el Departamento de Mantenimiento Vial de la Subsecretaria regional zona 7 dirección provincial de Loja del Ministerio de Transporte y Obras Públicas, el mismo que será parametrizable lo cual permita adaptar nuevos requerimientos solicitados por el usuario y de esta forma tengan la facilidad de gestionar (crear, modificar, activar/desactivar) funcionalidades de acuerdo a sus necesidades futuras.

2.2. Funciones del producto.

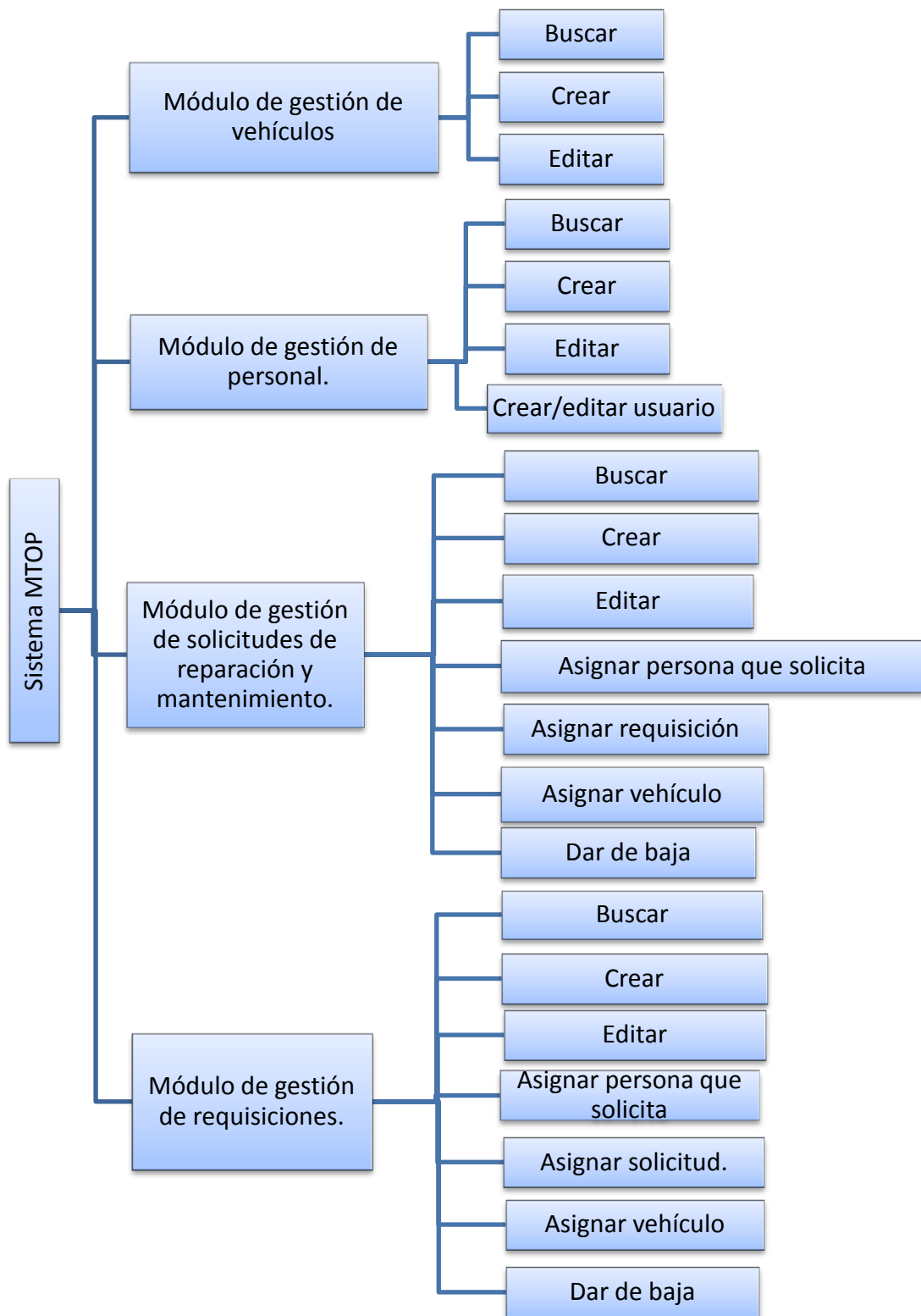


Figura 54. Módulos del software (a)

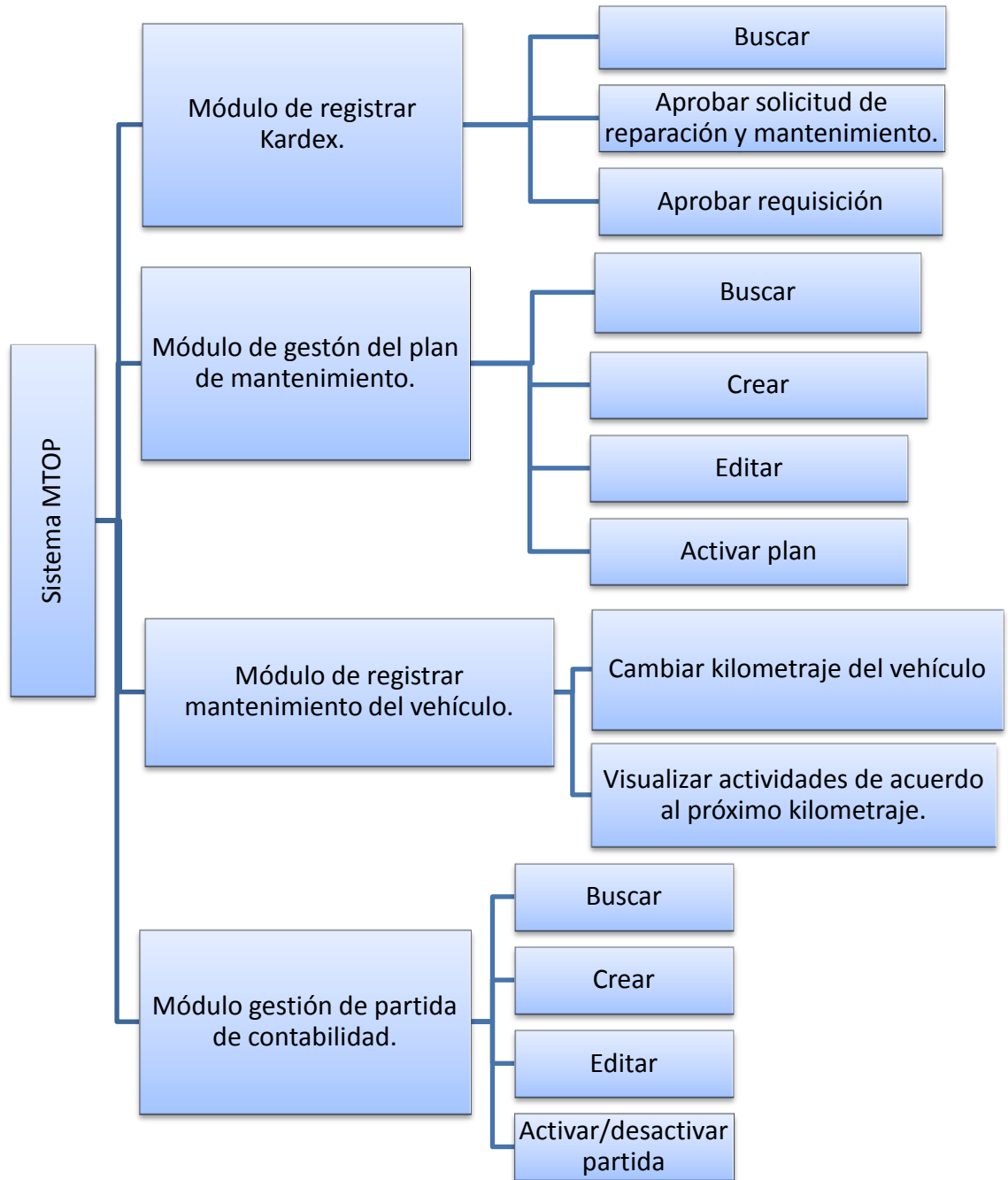


Figura 55. Módulos del software (b)

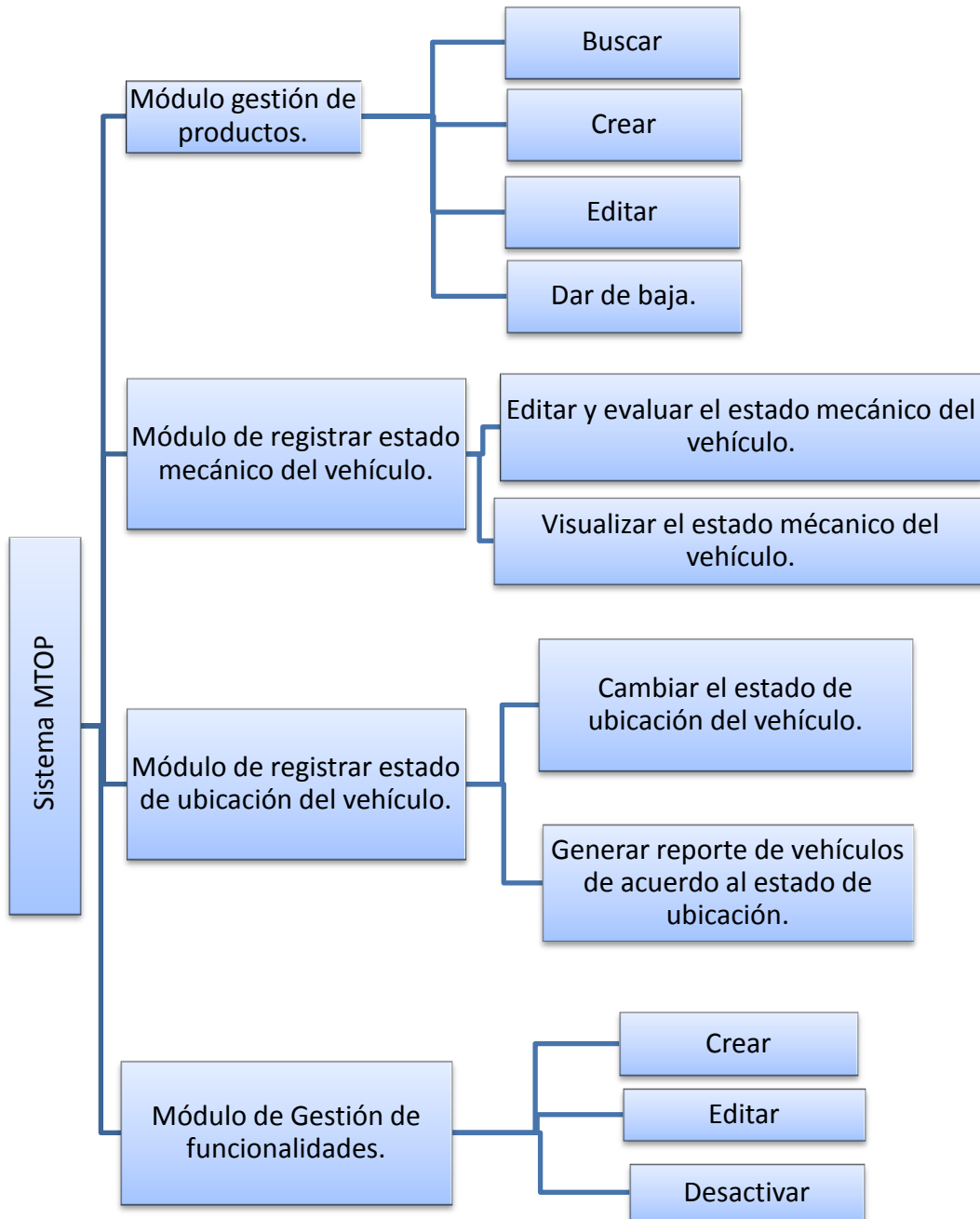


Figura 56. Módulos del software (c)

- Características de los usuarios.

TABLA LII.

CARACTERÍSTICAS DEL ADMINISTRADOR DEL SISTEMA.

Tipo de usuario:	Administrador del Sistema
Formación:	Universitario con conocimientos básicas de mecánica automotriz y afines.
Habilidades:	Conocimiento de estados y funcionamiento de vehículos.
Actividades:	<ul style="list-style-type: none"> • Crear, editar, buscar vehículos. • Crear, editar, buscar personal. • Gestionar usuarios. • Registrar kardex, asignar solicitud de reparación y mantenimiento a kardex, asignar requisición a kardex. • Generar: informe mensual de estado de los vehículos, resumen de estado general por vehículo. • Registrar estado de ubicación actual y estado mecánico del vehículo. • Generar reporte de usuarios y personal.

TABLA LIII.

CARACTERÍSTICAS DEL SECRETARIO DEL SISTEMA.

Tipo de usuario:	Secretario del Sistema
Formación:	Universitario con conocimientos básicas de secretaria.
Habilidades:	Conocimiento de solicitudes de reparación y requerimientos de reparación de vehículos.
Actividades:	<ul style="list-style-type: none"> • Crear solicitud de reparación y mantenimiento, asignar vehículo a solicitud, asignar persona que solicita la reparación, crear ítems a solicitud de reparación y mantenimiento. • Crear requisición, asignar vehículo a requisición, asignar persona que solicita la requisición, crear ítems a

	requisición. <ul style="list-style-type: none"> • Asignar solicitud a una requisición. • Asigna requisición a una solicitud.
--	--

2.3. Restricciones.

Para desarrollar la propuesta se usara lenguaje java específicamente JEE6, que consta JSF + Primefaces + Bootstrap para la interfaz, también se usará el repositorio de Maven y para la persistencia JPA. Los datos serán almacenados en la base de datos relacional PostgreSQL.

- El sistema operativo que se usará es Linux v12.04 para el desarrollo.
- Plataforma de desarrollo: Java Enterprise Edition 6
- Lenguajes de programación: Java, HTML 5, CSS3.
- Framework MVC: Java Server Faces (JSF)
- Implementación JSF para vista: Primefaces y Bootstrap v3.1.
- Servidor de aplicaciones: JBoss v2.
- Base de datos: PostgreSQL v9.1.
- El sistema no será integrado a ningún otro.
- Repositorio de librerías: Maven v2.3.1
- Framework para gestión de usuarios: Picketlink (Seam)

2.4. Suposición y dependencias.

- El sistema se desarrollará con arquitectura de tres capas.
- El equipo en el cual funcionará la aplicación deberá contar con los paquetes de Java y la base de datos PostgreSQL.

3. Requisitos específicos.

TABLA LIV.
REQUERIMIENTO FUNCIONAL 1

Número de requisito	RF001
Nombre del requisito	El sistema permitirá al administrador buscar crear, editar vehículos.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LV.
REQUERIMIENTO FUNCIONAL 2.

Número de requisito	RF002
Nombre del requisito	El sistema permitirá al administrador cambiar el estado de las partes mecánicas del vehículo.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LVI.
REQUERIMIENTO FUNCIONAL 3.

Número de requisito	RF003
Nombre del requisito	El sistema permitirá al administrador editar el estado de ubicación del vehículo (Equipo bueno inactivo, Equipo trabajando normalmente, Equipo trabajando con fallas, equipo en reparación, equipo para baja o remate), el lugar en donde se encuentra y la fecha por cada estado.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LVII.
REQUERIMIENTO FUNCIONAL 4.

Número de requisito	RF004
Nombre del requisito	El sistema permitirá al administrador asignar un conductor a un vehículo.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LVIII.
REQUERIMIENTO FUNCIONAL 5.

Número de requisito	RF005
Nombre del requisito	El sistema permitirá a la secretaria buscar, crear una solicitud de reparación y mantenimiento de un vehículo, con sus respectivos, ítems y el vehículo a reparar; las puede editar y dar de baja siempre y cuando la solicitud no esté aprobada por el administrador caso contrario solamente podrá visualizarla.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LIX.
REQUERIMIENTO FUNCIONAL 6.

Número de requisito	RF006
Nombre del requisito	El sistema permitirá al secretario asignar requisición de reparación en una solicitud de reparación y mantenimiento.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LX.
REQUERIMIENTO FUNCIONAL 7.

Número de requisito	RF007
Nombre del requisito	El sistema permitirá al secretario buscar y crear una requisición para las reparaciones de un vehículo, con sus respectivos ítems, vehículo y el número de partida que le corresponde; las podrá editar y dar de baja siempre y cuando no estén aprobadas por el

	administrador caso contrario solo podrá visualizarlas.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXI.
REQUERIMIENTO FUNCIONAL 8.

Número de requisito	RF008
Nombre del requisito	El sistema permitirá al secretario asignar solicitud de reparación y mantenimiento en una requisición de reparación.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXII.
REQUERIMIENTO FUNCIONAL 9.

Número de requisito	RF009
Nombre del requisito	El sistema permitirá al secretario asignar un producto a un ítem de requisición de reparación.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXIII.
REQUERIMIENTO FUNCIONAL 10.

Número de requisito	RF010
Nombre del requisito	El sistema permitirá al secretario buscar, crear, editar, y desactivar/activar una partida de contabilidad.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXIV.
REQUERIMIENTO FUNCIONAL 11.

Número de requisito	RF011
Nombre del requisito	El sistema permitirá al secretario buscar, crear, editar, y dar de baja un producto de reparación del vehículo.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXV.
REQUERIMIENTO FUNCIONAL 12.

Número de requisito	RF012
Nombre del requisito	El sistema permitirá al administrador buscar y editar kardex.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXVI.
REQUERIMIENTO FUNCIONAL 13.

Número de requisito	RF013
Nombre del requisito	El sistema permitirá al administrador aprobar solicitud de reparación y mantenimiento a una kardex.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXVII.
REQUERIMIENTO FUNCIONAL 14.

Número de requisito	RF014
Nombre del requisito	El sistema permitirá al administrador aprobar requisición de reparación a una kardex.
Tipo	Requisito

Prioridad del requisito	Alta/Esencial
-------------------------	---------------

TABLA LXVIII.
REQUERIMIENTO FUNCIONAL 15.

Número de requisito	RF015
Nombre del requisito	El sistema permitirá al administrador buscar, crear, editar y activar planes de mantenimiento.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXIX.
REQUERIMIENTO FUNCIONAL 16.

Número de requisito	RF016
Nombre del requisito	El sistema permitirá al administrador conocer por medio de alertas las revisiones necesarias que debe hacer al vehículo según el plan de mantenimiento asignado.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXX.
REQUERIMIENTO FUNCIONAL 17.

Número de requisito	RF017
Nombre del requisito	El sistema permitirá al administrador imprimir reportes del informe mensual del estado de los vehículos, resumen del estado general de los vehículos.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXXI.
REQUERIMIENTO FUNCIONAL 18.

Número de requisito	RF018
Nombre del requisito	El sistema permitirá al administrador buscar, crear, editar el personal con sus diferentes tipos (administrador, secretario, conductor, mecánico, etc).
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXXII.
REQUERIMIENTO FUNCIONAL 19.

Número de requisito	RF019
Nombre del requisito	El sistema permitirá al administrador crear, editar y desactivar las funcionalidades de acuerdo a requerimientos futuros.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA LXXIII.
REQUERIMIENTO FUNCIONAL 20.

Número de requisito	RF020
Nombre del requisito	Iniciar sesión en el sistema mediante el ingreso del usuario y contraseña
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.1. Requisitos comunes de la interfaz.

La interfaz del usuario deberá contener los campos necesarios para que el usuario o administrador del sistema agregue la información que requiera.

3.1.1. Interfaz de usuario.

- La interfaz del usuario deberá ser amigable y fácil de manejar, de esta manera el usuario podrá interactuar con el sistema sin que exista problemas.
- En la pantalla principal existirá un mensaje de bienvenida con el logotipo de la herramienta, e incluirá un link que enlazara a la pantalla del inicio de sesión.

3.1.2. Interfaz de hardware.

El sistema se ejecutará en un equipo con las siguientes características:

- Sistema operativo: Windows 7 de 32 bits.
- Procesador mínimo: DualCore 1.6 GHz o similar
- Procesador recomendado: Core i3 2.1 GHz o similar
- Memoria mínima: 1 GB
- Memoria recomendada: 2 GB
- Espacio en disco mínimo: 500 MB de espacio libre
- Espacio en disco recomendado: 1 GB de espacio libre.

Características del servidor:

- Procesador x64: AMD
- Disco duro 250 Gb, SAS o SATA, 7,2K rpm.
- Memoria al menos 4 GB
- Pantalla/tarjeta de video Pantalla mínimo de 17"

3.1.3. Interfaz del software.

La tecnología a utilizar es la siguiente:

- Plataforma de desarrollo: Java Enterprise Edition 6
- Lenguajes de programación: Java, HTML 5, CSS3.
- Framework MVC: JSF.
- Implementación JSF para vista: Primefaces y Bootstrap.
- Servidor de aplicaciones: JBoss.
- Base de datos: PostgreSQL.
- El sistema no será integrado a ningún otro.
- Repositorio de librerías: Maven
- Framework para gestión de usuarios: Picketlink

3.1.4. Interfaz de comunicación.

Se usara el protocolo de comunicación entre el programa y la base de datos el driver de Java JDBC.

3.2. Requerimientos funcionales.

Actores del sistema.

TABLA LXXIV.
ACTOR ADMINISTRADOR.

Actor:	Administrador.
Descripción:	Representa la persona que podrá, gestionar vehículos, kardex, personal, usuarios, realizar las configuraciones funcionales del software, obtener reportes.

TABLA LXXV.
ACTOR SECRETARIO

Actor:	Secretario
Descripción:	Representa la persona que podrá realizar “solicitudes de reparación y mantenimiento” y “requisiciones de reparación” o “requisiciones de bienes y servicios”, así como también asignar vehículos, conductor y personal que solicita.

TABLA LXXVI.
ACTOR INTERFAZ DEL SISTEMA

Actor:	Interfaz del sistema.
Descripción:	Representa el sistema que se encargara de realizar todas las validaciones de los datos que el Usuario Final del Sistema ingrese.

Los casos de uso gestionar funcionalidades, registrar estado de ubicación, registrar estado mecánico, registrar mantenimiento del vehículo y registrar kardex se encuentran en la sección de resultados.

TABLA LXXVII.
DESCRIPCIÓN DEL CASO DE USO GESTIONAR VEHÍCULO

Caso de Uso:	Gestionar Vehículo			Actor:	Administrador
Código:	CU001	Tipo:	Primario-Sistema	Pantallas:	P001, P001.01, P001.02, P001.03, P001.04, P001.05, P001.06, P001.07, P001.08, P001.09, P001.10.
Ref. Req:	RF001, RF004, RF0017.				
Resumen:	El Administrador, visualiza los vehículos, para poder editarlo o crearlo en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar vehículo. • Crear, editar un vehículo. 				
PreCondiciones:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. 				
PosCondiciones:	<ul style="list-style-type: none"> • Vehículo creado. 				
Curso normal de eventos					

1. El Administrador del sistema ingresa los criterios de búsqueda (Placa, número de registro, serie del chasis, serie del motor) para un vehículo y presiona el botón [Buscar] en la pantalla **Buscar Vehículo**.
2. El sistema actualiza la tabla [Resultados de Búsqueda] con los vehículos que coincidan con el criterio de búsqueda.
3. El Administrador verifica que exista el vehículo en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear] de la pantalla **Buscar vehículo**.
4. El sistema invoca la pantalla **Crear/Editar Vehículo**.
5. El Administrador selecciona el conductor que se va a asignar a un vehículo e ingresa las especificaciones técnicas del vehículo(registro, clase, tipo, marca, color, número de pasajeros, cilindraje, peso, placa, año de fabricación, modelo, tipo combustible, cabina, kilometraje, avalúo, tracción, número de cilindro, fecha de inicio y fin de soat), el historial(tiempo garantía, chatarrización, fecha de adquisición, año de servicio) y también cada una de las partes mecánicas del Vehículo presionando cada uno de los botones [Motor], [Chasis], [Transmisión], [Suspensión], [Sistema Eléctrico], [Frenos], [Dirección], [Carrocería], [Lubricantes], [Neumáticos]
6. El sistema invocará las pantallas **Motor, Chasis, Transmisión, Suspensión, Sistema Eléctrico, Frenos, Dirección, Carrocería, Lubricantes, Neumáticos, respectivamente**.
7. El administrador ingresa cada uno de los datos en las pantallas **Motor** (serie, marca, modelo, tipo, tanque), **Chasis** (serie, tipo, capacidad de tolva), **Transmisión** (marca, tipo, tracción, embrague), **Suspensión** (eje delantero, eje posterior), **Sistema Eléctrico** (voltaje, alternador, motor arranque, luces), **Frenos** (tipo, delantero, posterior), **Dirección** (tipo, radio), **Carrocería** (forma), **Lubricantes** (motor, caja, corona, hidráulico), **Neumáticos** (llantas, aros, presión) y presiona el botón [guardar] de la pantalla **Crear/Editar Vehículo**.
8. El sistema guarda el vehículo y presenta la pantalla **Buscar Vehículo**.
9. El caso de uso finaliza.

A. Limpiar Búsqueda.

A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Vehículo**.

A.2. El sistema limpia el criterio de búsqueda y actualiza en la tabla [Resultados de búsqueda] de la pantalla **Buscar Vehículo**.

B. Editar Vehículo.

B.3. El administrador elige el vehículo a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar vehículo**.

B.4. El sistema recupera la información del vehículo a editar, y presenta la pantalla **Crear/Editar vehículo**.

C.5. El Administrador modifica la información del vehículo como especificaciones técnicas del vehículo(registro, clase, tipo, marca, color, número de pasajeros, cilindraje, peso, placa, año de fabricación, modelo, tipo combustible, cabina, kilometraje, avalúo, tracción, número de cilindro), el historial tiempo garantía, chatarrización, fecha de adquisición, año de servicio) y también cada una de las partes mecánicas del Vehículo presionando cada uno de los botones [Motor], [Chasis], [Transmisión], [Suspensión], [Sistema Eléctrico], [Frenos], [Dirección], [Carrocería], [Lubricantes], [Neumáticos]

C.6. El sistema invoca las pantallas **Motor, Chasis, Transmisión, Suspensión, Sistema Eléctrico, Frenos, Dirección, Carrocería, Lubricantes, Neumáticos, respectivamente**.

C.7. El administrador modifica cada uno de los datos en las pantallas **Motor** (serie, marca, modelo, tipo, tanque), **Chasis** (serie, tipo, capacidad de tolva), **Transmisión** (marca, tipo, tracción, embrague), **Suspensión** (eje delantero, eje posterior), **Sistema Eléctrico** (voltaje, alternador, motor arranque, luces), **Frenos** (tipo, delantero, posterior), **Dirección** (tipo, radio), **Carrocería** (forma), **Lubricantes** (motor, caja, corona, hidráulico), **Neumáticos** (llantas, aros presión) y presiona el botón [Guardar] de la pantalla **Crear/Editar Vehículo**.

C.8. El sistema guarda el vehículo y presenta la pantalla **Buscar Vehículo**.

D. Campos Obligatorios Vacíos.

D.8 - (C.B.8.) El sistema verifica que estén llenos los campos obligatorios (conductor, registro, clase, tipo, marca, color, número de pasajeros, cilindraje, peso, placa, año de fabricación, modelo, tipo de combustible, cabina, kilometraje, avalúo, tracción, número de cilindro) presenta un mensaje “Campos obligatorios vacíos” en la pantalla **Crear/Editar Vehículo** y continúa (7, B.7).

E. Placa Duplicada

E.8 - (E.B.8) El sistema verifica si la placa esta duplicada y presenta un mensaje “La placa no está disponible. ¡Pertenece a otro vehículo!” en la pantalla **Crear/Editar Vehículo** y continúa (7, B.7).

F. Registro del vehículo Duplicado

F.8 EL sistema verifica si el número de registro esta duplicado y presenta un mensaje “El número de registro ingresado no está disponible. ¡Pertenece a otro vehículo!” en la pantalla **Crear/Editar Vehículo** y continúa (7, B.7).

G. Editar estado mecánico

G.8 El administrador elige el vehículo a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar estado mecánico] de la pantalla **Buscar vehículo**, e invoca al caso de uso CU002.

TABLA LXXVIII.
DESCRIPCIÓN DEL CASO DE USO GESTIONAR SOLICITUDES DE
REPARACIÓN Y MANTENIMIENTO.

Caso de Uso:	Gestionar Solicitudes de Reparación y Mantenimiento.	Actor:	Secretario.
---------------------	--	---------------	-------------

Código:	CU004	Tipo:	Primario-Sistema	Pantallas:	P003, P003.01, P003.02, P003.03, P003.04, P003.05, P003.06, P003.07, P003.08, P003.09, P004.01.
Ref. Req:	RF005.				
Resumen:	El Secretario, visualiza las solicitudes de reparación y mantenimiento que no están aprobadas, para poder editarlas o crearlas en caso que no existan.				
Objetivos:	<ul style="list-style-type: none"> • Buscar solicitudes de reparación y mantenimiento. • Crear, editar y dar de baja una solicitud de reparación y mantenimiento. 				
PreCondicion es:	<ul style="list-style-type: none"> • El Secretario se haya autenticado. 				
PosCondicion es:	<ul style="list-style-type: none"> • Solicitud de reparación y mantenimiento creada 				
Curso normal de eventos					
<ol style="list-style-type: none"> 1. El Secretario ingresa los criterios de búsqueda (registro del vehículo, placa, serie de motor, serie de chasis, fecha solicitud, fecha de la solicitud) para una solicitud de reparación y mantenimiento, y presiona el botón [Buscar] en la pantalla Buscar solicitudes de reparación y mantenimiento. 2. El sistema actualiza la tabla [Resultados de búsqueda], con las solicitudes de 					

reparación y mantenimiento que coincidan con el criterio de búsqueda.

3. El Secretario verifica que exista la solicitud de reparación y mantenimiento en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear].
4. El sistema invoca la pantalla **Crear/Editar solicitudes de reparación y mantenimiento.**
5. El Secretario registra los datos generales (número de solicitud, fecha solicitud, observación, persona que solicita (personal de la empresa), persona o entidad que recibe, persona que apruebe, requisición (CU008), vehículo (CU007) y los ítems (CU005)) de la solicitud de reparación y mantenimiento y presiona el botón [Guardar] de la pantalla **Crear/Editar solicitudes de reparación y mantenimiento.**
6. El sistema guarda la solicitud de reparación y mantenimiento con su respectiva información y presenta la pantalla **Buscar solicitudes de reparación y mantenimiento.**
7. El caso de uso finaliza

Curso alterno de eventos

A. Limpiar Búsqueda.

- A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar solicitudes de reparación y mantenimiento.**
- A.2. El sistema limpia el criterio de búsqueda de la solicitud de reparación y mantenimiento a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Solicitudes de reparación y mantenimiento.**

B. Editar solicitud de reparación y mantenimiento.

- B.3. El Secretario selecciona la solicitud de reparación y mantenimiento a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar solicitudes de reparación y mantenimiento.**
- B.4. El sistema recupera la información de la solicitud de reparación y mantenimiento, y los muestra en la pantalla **Crear/Editar solicitudes de reparación y mantenimiento.**
- B.5. El Secretario edita los datos correspondientes de la solicitud de reparación y

<p>mantenimiento (número de solicitud, fecha solicitud, observación, persona que solicita (personal de la empresa), persona o entidad que recibe, persona que apruebe, requisición (CU008), vehículo (CU007) y los ítems (CU005)), y presiona el botón [Guardar] de la pantalla Crear/Editar solicitudes de reparación y mantenimiento.</p> <p>B.6. El sistema guarda la nueva solicitud de reparación y mantenimiento, con su respectiva información y presenta la pantalla Buscar solicitudes de reparación y mantenimiento.</p>
<p>C. Dar de baja solicitudes de reparación y mantenimiento.</p> <p>C.3. El secretario selecciona la solicitud de reparación y mantenimiento a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [Dar de baja] de la pantalla Buscar solicitudes de reparación y mantenimiento.</p> <p>C4. El sistema invoca la pantalla Dar de baja solicitud de reparación y mantenimiento.</p> <p>C5. El secretario presiona el botón [Sí] de la pantalla Dar de baja solicitud de reparación y mantenimiento.</p> <p>C6. El sistema presenta un mensaje “Solicitud de reparación y mantenimiento seleccionada ha sido dada de baja”.</p>
<p>D. Visualizar solicitudes de reparación y mantenimiento aprobadas.</p> <p>D.1. El secretario presiona el botón [Solicitudes de Reparación Aprobadas] de la pantalla Buscar solicitudes de reparación y mantenimiento e invoca al caso de uso CU006.</p>
<p>E. Validar campos obligatorios.</p> <p>E.6 - (E.B.6). El sistema verifica que estén llenos los campos obligatorios (número de solicitud, fecha de solicitud, persona que solicita, persona o entidad que recibe, vehículo y al menos un ítem) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla Crear/Editar solicitudes de reparación y mantenimiento y continúa (5, B5).</p>

TABLA LXXIX.
DESCRIPCIÓN DEL CASO DE USO AGREGAR ÍTEM DE SOLICITUD DE
REPARACIÓN Y MANTENIMIENTO.

Caso de Uso:	Agregar ítem de solicitud de reparación y mantenimiento.		Actor:	Secretario
Código:	CU005	Tipo:	Primario-Sistema	Pantallas: P003, P003.01, P003.04, P003.05, P003.06.
Ref. Req:	RF005.			
Resumen:	El Secretario, agrega ítem a la solicitud de reparación y mantenimiento.			
Objetivos:	<ul style="list-style-type: none"> • Crear, editar, dar de baja ítem de solicitud de reparación y mantenimiento. • Agregar ítem a solicitud de reparación y mantenimiento. 			
PreCondicion es:	<ul style="list-style-type: none"> • Haya ingresado a la pantalla Buscar Solicitudes de Reparación y Mantenimiento y luego a la pantalla Crear/Editar Solicitudes de Reparación y Mantenimiento. 			
PosCondicion es:	<ul style="list-style-type: none"> • Ítem agregado a solicitud de reparación y mantenimiento. 			
Curso Normal de Eventos				
<ol style="list-style-type: none"> 1. El Secretario ingresa la información (elemento a revisar y descripción falla) del ítem y presiona el botón [Guardar] en la pantalla Crear/Editar ítem de solicitud de reparación y mantenimiento. 2. El sistema guarda el ítem de solicitud de reparación y actualiza la tabla de [Ítems] de solicitud de reparación de la pantalla Crear/Editar solicitudes de reparación y mantenimiento. 3. El caso de uso finaliza. 				
Curso alterno de eventos				

A. Editar ítem de solicitud de reparación y mantenimiento

A.1. El sistema recupera la información del ítem y los muestra en la pantalla **Crear/Editar ítem de solicitud de reparación y mantenimiento**

A.2. El Secretario edita los datos correspondientes del [Ítem] para la solicitud de reparación y mantenimiento (elemento a revisar y descripción falla) y presiona el botón [Guardar] de la pantalla **Crear/Editar ítem de solicitud de reparación y mantenimiento**.

A.3. El sistema guarda el nuevo ítem de la solicitud de reparación y actualiza la tabla de [Ítems] de solicitud de reparación de la pantalla **Crear/Editar Solicitudes de Reparación y Mantenimiento**.

B. Dar de baja ítem de solicitudes de reparación y mantenimiento

B.3. El secretario selecciona el ítem de solicitud de reparación y mantenimiento a dar de baja de la tabla [Ítem] de solicitud de reparación de mantenimiento y presiona el botón [Dar de baja] de la pantalla **Crear/Editar solicitudes de reparación y mantenimiento**.

B4. El sistema invoca la pantalla **Dar de baja ítem de solicitud de reparación y mantenimiento**.

B5. El secretario presiona el botón [Sí], en la pantalla **Dar de baja ítem de solicitud de reparación y mantenimiento**.

B6. El sistema presenta un mensaje "ítem de solicitud de reparación y mantenimiento seleccionado ha sido dado de baja".

C. Validar campos obligatorios.

C.3 - (C.A.3). El sistema verifica que estén llenos los campos obligatorios (elemento a revisar y descripción falla) en la pantalla **Crear/Editar ítem de solicitud de reparación y mantenimiento**, y continúa (1, A2).

TABLA LXXX.
DESCRIPCIÓN DEL CASO DE USO VISUALIZAR SOLICITUD DE REPARACIÓN
Y MANTENIMIENTO APROBADAS.

Caso de Uso:	Visualizar solicitudes de reparación y mantenimiento aprobadas.			Actor:	Secretario
Código:	CU006	Tipo:	Primario-Sistema	Pantallas:	P003, P003.09
Ref. Req:	RF005.				
Resumen:	El Secretario, busca las solicitudes que están aprobadas, para poder visualizarlas.				
Objetivos:	<ul style="list-style-type: none"> • Buscar solicitudes de reparación y mantenimiento aprobadas. • Visualiza las solicitudes de reparación y mantenimiento aprobadas. 				
PreCondiciones:	<ul style="list-style-type: none"> • El Secretario se haya autenticado. • Haya ingresado a la pantalla Buscar Solicitudes de Reparación y Mantenimiento. 				
PosCondiciones:	<ul style="list-style-type: none"> • Solicitud de reparación y mantenimiento visualizada. 				
Curso normal de eventos					
<ol style="list-style-type: none"> 1. El Secretario presiona el botón [Solicitudes Aprobadas] en la pantalla Buscar solicitudes de reparación y mantenimiento. 2. El sistema invoca la pantalla Buscar solicitudes de reparación y mantenimiento aprobadas. 3. El Secretario ingresa los criterios de búsqueda para una solicitud de reparación 					

<p>y mantenimiento aprobada (registro del vehículo, placa, serie de motor, serie de chasis, fecha de solicitud, registro de la solicitud) y presiona el botón [Buscar] en la pantalla Buscar solicitudes de reparación y mantenimiento aprobadas.</p> <p>4. El sistema actualiza la tabla [Resultados de búsqueda], con las solicitudes de reparación aprobadas que coincidan con el criterio de búsqueda.</p> <p>5. El secretario selecciona la solicitud de reparación a visualizar de la tabla [Resultados de Búsqueda] y presiona el botón [Visualizar] de la pantalla Buscar solicitudes de reparación y mantenimiento aprobadas.</p> <p>6. El sistema presenta la solicitud de reparación y mantenimiento seleccionada en la pantalla solicitud de reparación y mantenimiento.</p> <p>7. El caso de uso finaliza.</p>
Curso alternativo de eventos
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Buscar solicitudes de reparación y mantenimiento aprobadas.</p> <p>A.2. El sistema limpia el nombre de la solicitud de reparación y mantenimiento a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar solicitudes de reparación y mantenimiento aprobadas.</p>

TABLA LXXXI.

DESCRIPCIÓN DEL CASO DE USO ASIGNAR VEHÍCULO A UNA SOLICITUD DE REPARACIÓN Y MANTENIMIENTO.

Caso de Uso:	Asignar vehículo a una solicitud de reparación y mantenimiento.		Actor:	Secretario
Código:	CU007	Tipo:	Primario-Sistema	Pantallas: P003, P003.01,P0 03.03
Ref. Req:	RF005.			
Resumen:	El secretario, busca y selecciona el vehículo que va asignar a la solicitud de reparación y mantenimiento.			

Objetivos:	<ul style="list-style-type: none"> • Buscar vehículo. • Agregar vehículo a solicitud de reparación y mantenimiento.
PreCondiciones:	<ul style="list-style-type: none"> • Haya ingresado a la pantalla Crear/Editar Solicitudes de Reparación y Mantenimiento.
PosCondiciones:	<ul style="list-style-type: none"> • Vehículo agregado a solicitud de reparación y mantenimiento.
Curso normal de eventos	
<ol style="list-style-type: none"> 1. El Secretario ingresa los criterios de búsqueda para un vehículo (registro, placa, serie de motor, serie de chasis) y presiona el botón [Buscar] en la pantalla Asignar Vehículo. 2. El sistema actualiza la tabla [Resultados de búsqueda], con el vehículo que coincida con el criterio de búsqueda. 3. El Secretario verifica que exista el vehículo en la tabla [Resultados de búsqueda] y presiona el botón [Agregar] en la pantalla Asignar Vehículo. 4. El sistema invoca la pantalla Crear/Editar solicitudes de reparación y mantenimiento. 5. El caso de uso finaliza. 	
Curso alternativo de eventos	
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Asignar Vehículo.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Asignar Vehículo.</p>	

TABLA LXXXII.
DESCRIPCIÓN DEL CASO DE USO ASIGNAR REQUISICIÓN.

Caso de Uso:	Asignar Requisición			Actor:	Secretario
Código:	CU008	Tipo:	Primario-Sistema	Pantallas:	P003, P003.01,P003.02

Ref. Req:	RF006.
Resumen:	El Secretario, busca y selecciona la requisición que va agregar a la solicitud de reparación y mantenimiento.
Objetivos:	<ul style="list-style-type: none"> • Buscar requisición • Agregar requisición a solicitud de reparación y mantenimiento.
PreCondiciones:	<ul style="list-style-type: none"> • Haya ingresado a la pantalla Buscar solicitudes de reparación y mantenimiento y haya ingresado a la pantalla Crear/Editar solicitudes de reparación y mantenimiento.
PosCondiciones:	<ul style="list-style-type: none"> • Requisición agregada a solicitud de reparación y mantenimiento.
Curso normal de eventos	
<ol style="list-style-type: none"> 1. El Secretario ingresa los criterios de búsqueda para una requisición (fecha requisición, registro de requisición). y presiona el botón [Buscar] en la pantalla Asignar requisición. 2. El sistema actualiza la tabla [Resultados de búsqueda], con la requisición que coincida con el criterio de búsqueda en la pantalla Asignar requisición. . 3. El Secretario verifica que exista la requisición en la tabla [Resultados de búsqueda] y presiona el botón [Agregar] en la pantalla Asignar requisición. 4. El sistema invoca la pantalla Crear/Editar solicitudes reparación y mantenimiento. 5. El caso de uso finaliza. 	
Curso alterno de eventos	
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Asignar requisición.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Asignar requisición.</p>	

B. Crear requisición.

B.1 El secretario presiona el botón [Crear requisición], de la pantalla Asignar solicitud requisición e invoca al CU009.

TABLA LXXXIII.

DESCRIPCIÓN DEL CASO DE USO GESTIONAR REQUISICIÓN.

Caso de Uso:	Gestionar requisición.		Actor:	Secretario
Código:	CU009	Tipo:	Primario - Sistema	Pantallas: P004, P004.01, P004.02, P004.03, P004.04, P004.05, P004.06, P004.07, P004.08, P004.09.
Ref. Req:	RF007.			
Resumen:	El Secretario, visualiza las requisiciones que no están aprobadas, para poder editar o crear en caso que no exista.			
Objetivos:	<ul style="list-style-type: none"> • Buscar requisiciones. • Crear, editar y dar de baja una requisición. 			
PreCondiciones :	<ul style="list-style-type: none"> • El Secretario se haya autenticado. 			
PosCondiciones :	<ul style="list-style-type: none"> • Requisición creada. 			
Curso normal de eventos				
1. El Secretario ingresa los criterios de búsqueda (registro del vehículo, placa, serie de motor, serie de chasis, registro de requisición, fecha de requisición,				

número de partida de contabilidad) para una requisición y presiona el botón [Buscar] en la pantalla **Buscar requisición**.

2. El sistema actualiza la tabla [Resultados de búsqueda], con las requisiciones que coincidan con el criterio de búsqueda.
3. El secretario verifica que exista la requisición en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear].
4. El sistema invoca la pantalla **Crear/Editar requisición**.
5. El secretario registra los datos generales (número de requisición, fecha de requisición, tipo de requisición, partida, tipo adquisición, observación, persona que solicita, persona que apruebe, ítems (CU012) de la requisición en la pantalla **Crear/Editar requisición** y según el tipo de requisición que seleccione se activaran cierto botones, en el caso de seleccionar requisición de reparación se activara el botón [Asignar Vehículo] en la pantalla **Crear/Editar requisición** e invoca al caso de uso CU011, caso contrario no se activará, también si escoge el tipo de adquisición bodega se activará el botón [Asignar Producto] en la pantalla **Crear/Editar ítem de requisición** invoca al caso de uso CU014. El Secretario también podrá asignar una requisición presionando el botón [Asignar solicitud] en la pantalla **Crear/Editar requisición**, e invoca al caso de uso CU013. Finalmente presiona el botón [Guardar] de la pantalla **Crear/Editar requisición**.
6. El sistema guarda la requisición y presenta la pantalla **Buscar requisición**.
7. El caso de uso finaliza.

Curso alternativo de eventos

A. Limpiar Búsqueda.

- A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar Requisición**.
- A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Requisición**.

B. Editar Requisición.

- B.3. El Secretario selecciona la requisición a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar requisición**.

B.4. El sistema recupera la información de la requisición y los muestra en la pantalla **Crear/Editar requisición**.

B.5. El Secretario edita los datos correspondientes de la requisición generales (número de requisición, fecha de requisición, tipo de requisición, partida, tipo adquisición, observación, persona que solicita, persona que apruebe, ítems (CU012)) de la requisición en la pantalla **Crear/Editar requisición**, si elige el tipo de requisición de reparación se activara el botón [Asignar Vehículo] en la pantalla **Crear/Editar Requisición** e invoca al caso de uso CU011, también si escoge el tipo de adquisición bodega se activará el botón [Asignar Producto] en la pantalla **Crear/Editar ítem de requisición** invoca al caso de uso CU014, caso contrario no se activara. El Secretario también podrá asignar una requisición presionando el botón [Asignar solicitud] en la pantalla **Crear/Editar requisición**, e invoca al caso de uso CU013) y presiona el botón [Guardar] de la pantalla **Crear/Editar Requisición**.

B.6. El sistema guarda la nueva requisición con su respectiva información y presenta la pantalla **Buscar requisiciones** con el mensaje “Requisición editada”.

C. Dar de baja Requisición.

C.3. El secretario selecciona la requisición a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [dar de baja] de la pantalla **Buscar Requisición**.

C.4. El sistema invoca la pantalla **Dar de baja Requisición**.

C.5. El secretario presiona el botón [Sí], de la pantalla **Dar de baja Requisición**.

C.6. El sistema presenta un mensaje “Requisición seleccionada ha sido dado de baja”.

D. Visualizar requisiciones aprobadas.

D.1. El secretario presiona el botón [Requisiciones Aprobadas] de la pantalla **Buscar Requisición** e invoca al caso de uso CU010.

E. Validar Campos Obligatorios.

E.6 (E.B.6). El sistema verifica que estén llenos los campos obligatorios (número de requisición, fecha de requisición, tipo de requisición, partida, tipo adquisición, personal que solicita, el vehículo y al menos un ítem) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla **Crear/Editar requisición**, y continúa

(5, B5).

TABLA LXXXIV.

DESCRIPCIÓN DEL CASO DE USO VISUALIZAR REQUISICIONES APROBADAS.

Caso de Uso:	Visualizar requisiciones aprobadas			Actor:	Secretario
Código:	CU010	Tipo:	Primario-Sistema	Pantallas:	P004, P004.09
Ref. Req:	RF007.				
Resumen:	El Secretario, busca las requisiciones que están aprobadas, para poder visualizarlas.				
Objetivos:	<ul style="list-style-type: none">• Buscar requisiciones aprobadas.• Visualiza las requisiciones aprobadas.				
PreCondiciones:	<ul style="list-style-type: none">• El Secretario se haya autenticado.• Haya ingresado a la pantalla Buscar requisición				
PosCondiciones:	<ul style="list-style-type: none">• Requisición visualizada.				
Curso normal de eventos					
<ol style="list-style-type: none">1. El Secretario presiona el botón [Requisiciones Aprobadas] en la pantalla Buscar Requisición.2. El sistema presenta la pantalla Buscar requisiciones aprobadas.3. El Secretario ingresa los criterios de búsqueda (placa, serie chasis, serie motor, registro del vehículo, registro de requisición, fecha de requisición) para una					

<p>requisición aprobada y presiona el botón [Buscar] en la pantalla Buscar Requisiciones Aprobadas.</p> <p>4. El sistema actualiza la tabla [Resultados de búsqueda], con las requisiciones aprobadas que coincidan con el criterio de búsqueda.</p> <p>5. El secretario selecciona la requisición a visualizar de la tabla [Resultados de Búsqueda] y presiona el botón [Visualizar] de la pantalla Buscar requisiciones aprobadas.</p> <p>6. El sistema presenta la Requisición seleccionada en la pantalla Requisición.</p> <p>7. El caso de uso finaliza.</p>
Curso alterno de eventos
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Buscar requisiciones aprobadas.</p> <p>A.2. El sistema limpia el nombre de la requisición a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar requisiciones aprobadas.</p>

TABLA LXXXV.

DESCRIPCIÓN DEL CASO DE USO ASIGNAR VEHÍCULO A REQUISICIÓN.

Caso de Uso:	Asignar vehículo a requisición.		Actor:	Secretario
Código:	CU011	Tipo:	Primario-Sistema	Pantallas: P004,P004.01, P003, P003.01, P003.03
Ref. Req:	RF007.			
Resumen:	El Secretario, busca y selecciona el vehículo que va agregar a la Requisición.			
Objetivos:	<ul style="list-style-type: none"> • Buscar vehículo. • Agregar vehículo a Requisición. 			
PreCondicion es:	<ul style="list-style-type: none"> • El Secretario se haya autenticado. • Haya ingresado a la pantalla Crear/Editar requisición. 			

PosCondicion es:	<ul style="list-style-type: none"> • Vehículo agregado a solicitud de reparación y mantenimiento.
Curso normal de eventos	
<ol style="list-style-type: none"> 1. El Secretario ingresa los criterios de búsqueda para un vehículo (registro del vehículo, placa, serie de motor, serie de chasis) y presiona el botón [Buscar] en la pantalla Asignar vehículo. 2. El sistema actualiza la tabla [Resultados de búsqueda], con el vehículo que coincida con el criterio de búsqueda. 3. El Secretario verifica que exista el vehículo en la tabla [Resultados de búsqueda] y presiona el botón [Agregar] en la pantalla Asignar vehículo y presenta la pantalla Crear/Editar requisición. 4. El caso de uso finaliza. 	
Curso alternativo de eventos	
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Asignar Vehículo.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Asignar Vehículo.</p>	

TABLA LXXXVI.
DESCRIPCIÓN DEL CASO DE USO AGREGAR ÍTEM DE REQUISICIÓN

Caso de Uso:	Agregar ítem de requisición.		Actor:	Secretario
Código:	CU012	Tipo:	Primario-Sistema	Pantallas: P004,P004.01,P004.04, P004.05, P004.06
Ref. Req:	RF007.			
Resumen:	El Secretario, agrega ítem a requisición.			
Objetivos:	<ul style="list-style-type: none"> • Crear, editar, dar de baja ítem de requisición. • Agregar ítem a requisición. 			

PreCondicion es:	<ul style="list-style-type: none"> • Haya ingresado a la pantalla Buscar Requisiciones y haya ingresado a la pantalla Crear/Editar Requisición.
PosCondicion es:	<ul style="list-style-type: none"> • Ítem agregado a requisición
Curso Normal de Eventos	
<ol style="list-style-type: none"> 1. El Secretario ingresa la información del ítem (cantidad, descripción y unidad de medida en caso de que haya seleccionado el tipo de adquisición bodega se activa el botón [Asignar Producto], selecciona un producto y presiona el botón [Asignar Producto] en la pantalla Crear/Editar ítem de requisición e invoca al caso de uso CU014) y presiona el botón [Guardar] en la pantalla Crear/Editar ítem de requisición. 2. El sistema guarda el ítem de solicitud de requisición y actualiza la tabla de ítems de requisición de la pantalla Crear/Editar Requisición. 3. El caso de uso finaliza. 	
Curso alternativo de eventos	
A. Editar ítem de requisición.	
A.1. El sistema recupera la información del ítem y los muestra en la pantalla Crear/Editar ítem de requisición.	
A.2. El Secretario edita los datos correspondientes del ítem de la requisición y presiona el botón [Guardar] de la pantalla Crear/Editar ítem de requisición.	
A.3. El sistema guarda el nuevo ítem de la requisición y actualiza la tabla de [ítems] de requisición de la pantalla Crear/Editar requisición.	
B. Dar de baja ítem de requisición	
B.1. El secretario selecciona el ítem de requisición a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [dar de baja] de la pantalla Crear/Editar ítem de requisición.	
B.2. El sistema invoca la pantalla Dar de baja ítem de requisición.	
B.3. El secretario presiona el botón [Sí], en la pantalla Dar de baja ítem de requisición.	
B. B.4. El sistema presenta un mensaje "ítem de requisición seleccionada ha sido	

dado de baja”.
<p>C. Validar Campos Obligatorios.</p> <p>C.2 - (C.A.3). El sistema verifica que estén llenos los campos obligatorios (cantidad, descripción y unidad medida) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla Crear/Editar Ítem de Requisición y continúa (1, A2).</p>

TABLA LXXXVII.
DESCRIPCIÓN DEL CASO DE USO ASIGNAR SOLICITUD DE REPARACIÓN Y
MANTENIMIENTO

Caso de Uso:	Asignar solicitud reparación y mantenimiento.		Actor:	Secretario
Código:	CU013.	Tipo:	Primario-Sistema.	Pantallas: P004, P004.01, P004.02.
Ref. Req:	RF008.			
Resumen:	El Secretario, busca y selecciona la solicitud reparación y mantenimiento que va asignar a la requisición.			
Objetivos:	<ul style="list-style-type: none"> • Buscar solicitud reparación y mantenimiento. • Agregar solicitud reparación y mantenimiento a requisición. 			
Precondicion es:	<ul style="list-style-type: none"> • El Secretario se haya autenticado. • Haya ingresado a la pantalla Buscar Requisiciones y Haya ingresado a la pantalla Crear/Editar requisiciones 			
PosCondicion es:	<ul style="list-style-type: none"> • Solicitud de reparación y mantenimiento agregada a requisición. 			
Curso normal de eventos.				
1. El Secretario ingresa los criterios de búsqueda para una solicitud de reparación y mantenimiento (registro de solicitud, fecha de registro de solicitud), y presiona el botón [Buscar] en la pantalla Asignar Solicitud de Reparación y				

<p>Mantenimiento.</p> <p>2. El sistema actualiza la tabla [Resultados de búsqueda], con la solicitud de reparación y mantenimiento que coincida con el criterio de búsqueda en la pantalla Asignar Solicitud de Reparación y Mantenimiento.</p> <p>3. El Secretario verifica que exista la solicitud de reparación y mantenimiento en la tabla [Resultados de búsqueda] y presiona el botón [Agregar] en la pantalla Asignar Solicitud de Reparación y Mantenimiento</p> <p>4. El sistema invoca la pantalla Crear/Editar Requisición.</p> <p>5. El caso de uso finaliza</p>
Cursos alterno de eventos.
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Asignar Solicitud de Reparación y Mantenimiento.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Asignar Solicitud de Reparación y Mantenimiento.</p>
<p>B. Crear Solicitud</p> <p>B.1. El secretario presiona el botón [Crear Solicitud], de la pantalla Asignar Solicitud de Reparación y Mantenimiento e invoca el CU004</p>

TABLA LXXXVIII.
DESCRIPCIÓN DEL CASO DE USO ASIGNAR PRODUCTO.

Caso de Uso:	Asignar producto.			Actor:	Secretario
Código:	CU014	Tipo:	Primario-Sistema	Pantallas:	P004,P004.01, P004.03, P004.04
Ref. Req:	RF009.				
Resumen:	El Secretario, busca y selecciona el producto que va asignar a ítem de requisición.				
Objetivos:	<ul style="list-style-type: none"> • Buscar producto • Agregar producto a ítem de requisición. 				

Precondicion es:	<ul style="list-style-type: none"> • El Secretario se haya autenticado. • Haya ingresado a la pantalla Buscar Requisiciones, a la pantalla Crear/Editar requisiciones y a la pantalla Crear/Editar ítem de requisición.
PosCondicion es:	<ul style="list-style-type: none"> • Producto agregado al ítem de requisición.
Curso normal de eventos	
<ol style="list-style-type: none"> 1. El Secretario presiona el botón [Asignar Producto] en la pantalla Crear/Editar ítem de requisición 2. El sistema invoca la pantalla Asignar Producto. 3. El Secretario ingresa los criterios de búsqueda del producto (código, descripción) y presiona el botón [Buscar] en la pantalla Asignar Producto. 4. El sistema actualiza la tabla [Resultados de búsqueda], con los productos que coincidan con el criterio de búsqueda de la pantalla Asignar Producto. 5. El secretario selecciona el producto y presiona el botón [Agregar] en la pantalla Asignar Producto. 6. El sistema invoca la pantalla Crear/Editar ítem de requisición. 7. El caso de uso finaliza 	
Curso alterno de eventos	
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Secretario presiona el botón [Limpiar] de la pantalla Asignar Producto.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla Asignar Producto.</p>	

TABLA LXXXIX.

GESTIONAR PARTIDA DE CONTABILIDAD.

Caso de Uso:	Gestionar Partida de Contabilidad.		Actor:	Secretario
Código:	CU015.	Tipo:	Primario-Sistema.	Pantallas: P006, P006.01, P006.02, P006.03, P006.04, P006.05.

Ref. Req:	RF010.
Resumen:	El Secretario, visualiza las partidas de contabilidad, para poder editar o crear en caso que no exista.
Objetivos:	<ul style="list-style-type: none"> • Buscar partida contabilidad. • Crear, editar y dar de baja una partida de contabilidad.
Precondiciones:	<ul style="list-style-type: none"> • El Secretario se haya autenticado.
PosCondiciones:	<ul style="list-style-type: none"> • Partida de contabilidad creada.
Curso normal de eventos.	
<ol style="list-style-type: none"> 1. El Secretario ingresa los criterios de búsqueda (Descripción de la partida, número de partida, tipo de partida) para una partida de contabilidad y presiona el botón [Buscar] en la pantalla Buscar Partida de Contabilidad. 2. El sistema actualiza la tabla [Resultados de búsqueda], con la partida que coincidan con el criterio de búsqueda. 3. El Secretario verifica que exista la partida de contabilidad en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear] de la pantalla Buscar Partida de Contabilidad. 4. EL sistema invoca la pantalla Ingresar Partida de Contabilidad. 5. El Secretario registra los datos generales (tipo, número de provincia, número de programa, número de proyecto, número de ítem, número de la fuente financiera, descripción) de la partida Contabilidad y presiona el botón [Guardar] de la pantalla Ingresar Partida de Contabilidad. 6. El sistema guarda la partida de contabilidad con su respectiva información (número de provincia, número de programa, número de proyecto, número de ítem, número de la fuente financiera, descripción) y presenta la pantalla Buscar Partida de Contabilidad. 7. El caso de uso finaliza. 	
Curso alterno de eventos.	
A. Limpiar Búsqueda.	
A.1. El Secretario presiona el botón [Limpiar] de la pantalla Buscar Partida de Contabilidad .	
A.2. El sistema limpia el criterio de búsqueda de la partida de contabilidad a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla Buscar Partida de	

<p>Contabilidad.</p>
<p>B. Editar Partida de contabilidad.</p> <p>B.3. El Secretario selecciona la partida de contabilidad a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla Buscar Partida de Contabilidad.</p> <p>B.4. El sistema recupera la información de la partida de contabilidad a editar, y presenta la pantalla Ingresar Partida de Contabilidad.</p> <p>B.5. El Secretario modifica la información (número de provincia, número de programa, número de proyecto, número de ítem, número de la fuente financiera, descripción) de la partida de contabilidad y presiona el botón [Guardar] de la pantalla Ingresar Partida de Contabilidad.</p> <p>B.6. El sistema guarda la nueva partida de contabilidad y presenta la pantalla Buscar Partida de Contabilidad con el mensaje “Partida de contabilidad editada”.</p>
<p>C. Desactivar.</p> <p>C.3. El secretario selecciona la partida a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [desactivar] de la pantalla Buscar Partida de Contabilidad.</p> <p>C.4. El sistema invoca la pantalla Desactivar partida de contabilidad.</p> <p>C.5. El secretario presiona el botón [Sí], de la pantalla Desactivar partida de contabilidad.</p> <p>C.6. El sistema presenta un mensaje “La Partida de contabilidad seleccionada ha sido dada de baja”.</p> <p>C.6. El sistema presenta la pantalla Buscar Partida de Contabilidad, con el mensaje “La partida de contabilidad seleccionada ha sido desactivada”.</p>
<p>D. Activar</p> <p>D.3. El secretario selecciona la partida a activar de la tabla [Resultados de Búsqueda] y presiona el botón [activar] de la pantalla Buscar Partida de Contabilidad.</p> <p>D.4. El sistema invoca la pantalla Activar partida de contabilidad.</p> <p>D.5. El secretario presiona el botón [Sí], de la pantalla Activar partida de contabilidad.</p> <p>D.6. El sistema presenta la pantalla Buscar Partida de Contabilidad, con el mensaje “La partida de contabilidad seleccionada ha sido activada”.</p>

E. Validar Campos Obligatorios.

E.6-(E.B.6). El sistema verifica que estén llenos los campos obligatorios (número de provincia, número de programa, número de proyecto, número de ítem, número de la fuente financiera, descripción) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla **Ingresar Partida de Contabilidad** y continúa (5, B.5).

TABLA XC.

DESCRIPCIÓN DEL CASO DE USO GESTIONAR PRODUCTO.

Caso de Uso:	Gestionar Productos.			Actor:	Secretario
Código:	CU016.	Tipo:	Primario-Sistema.	Pantallas:	P007, P007.01, P007.02, P007.03.
Ref. Req:	RF011.				
Resumen:	El Secretario, visualiza el producto registrado para poder editarlo o crearlo en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar Producto. • Crear, Editar o dar de baja un producto. 				
Precondicion es:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. 				
PosCondicion es:	<ul style="list-style-type: none"> • Producto creado. 				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El Secretario del sistema ingresa los criterios de búsqueda (código, descripción) para un producto y presiona el botón [Buscar] en la pantalla Buscar Productos de Reparación. 2. El sistema actualiza la tabla [Resultados de Búsqueda] con los productos que coincidan con el criterio de búsqueda. 3. El Secretario verifica que exista el producto en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear] de la pantalla 					

Buscar Productos de Reparación.

4. El sistema invoca la pantalla **Crear/Editar producto de reparación.**
5. El Secretario ingresa la información del producto (código, cantidad, costo, descripción) y presiona el botón [Guardar] de la pantalla **Crear/Editar Productos de Reparación.**
6. El sistema guarda el producto y presenta la pantalla **Buscar Productos de Reparación.**
7. El caso de uso finaliza.

Curso alternativo de eventos.

A. Limpiar Búsqueda.

- A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar Producto de Reparación.**
- A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Producto de Reparación.**

B. Editar.

- B.3. El Secretario elige el producto a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Producto de Reparación.**
- B.4. El sistema recupera la información del personal a editar, y la presenta la pantalla **Crear/Editar Producto de Reparación.**
- B.5. El Secretario modifica la información (código, cantidad, costo, descripción) del producto y presiona el botón [Guardar] de la pantalla **Crear/Editar Producto de Reparación.**
- B.6. El sistema guarda al nuevo producto y presenta la pantalla **Buscar Producto de Reparación.**

C. Campos Obligatorios Vacíos.

- C.6. - (C.B.6). El sistema verifica que estén llenos los campos obligatorios (código, cantidad, costo, descripción) presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Producto de Reparación** y continúa (5, B.5).

D. Dar de Baja

- D.3. El Secretario elige el producto a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [Dar de Baja] de la pantalla **Buscar Producto de**

<p>Reparación.</p> <p>D.4. El sistema invoca la pantalla Dar de baja Producto de Reparación</p> <p>D.5. El Secretario presiona el botón [Sí], de la pantalla Dar de Producto de Reparación.</p> <p>D.6. El sistema presenta un mensaje “El Producto seleccionado ha sido dado de baja”.</p>
--

TABLA XCI.

DESCRIPCIÓN DEL CASO DE USO APROBAR SOLICITUD DE REPARACION Y MANTENIMIENTO.

Caso de Uso:	Aprobar solicitud reparación y mantenimiento.			Actor:	Administrador
Código:	CU018	Tipo:	Primario-Sistema.	Pantallas:	P007, P007.01, P007.02
Ref. Req:	RF013.				
Resumen:	El Administrador, busca y selecciona la solicitud reparación que va aprobar.				
Objetivos:	<ul style="list-style-type: none"> • Buscar solicitud reparación. • Aprobar solicitud reparación. • Asignar solicitud reparación a Kardex. 				
Precondicione s:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. • Haya ingresado a las pantallas Buscar Kardex y Crear/Editar kardex. 				
PosCondicion es:	<ul style="list-style-type: none"> • Solicitud de reparación aprobada y asignar en kardex. 				
Curso normal de eventos.					
1. El Administrador ingresa los criterios de búsqueda (número de solicitud, fecha de solicitud) para una solicitud y presiona el botón [Buscar] en la pantalla					

<p>Aprobar solicitud de reparación y mantenimiento.</p> <p>2. El sistema actualiza la tabla [Solicitudes aprobadas], con la solicitud de reparación que coincida con el criterio de búsqueda en la pantalla Aprobar solicitud de reparación y mantenimiento.</p> <p>3. El Administrador verifica que exista la solicitud de reparación en la tabla [Solicitudes aprobadas], selecciona la solicitud de reparación y mantenimiento y presiona el botón [Aprobar] en la pantalla Aprobar solicitud de reparación y mantenimiento.</p> <p>4. El sistema invoca la pantalla Aprobar solicitud de reparación y mantenimiento.</p> <p>5. El administrador presiona el botón [Guardar], de la pantalla Aprobar solicitud de reparación y mantenimiento.</p> <p>6. El sistema invoca la pantalla Crear/Editar kardex.</p> <p>7. El caso de uso finaliza.</p>
Curso alterno de eventos.
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Administrador presiona el botón [Limpiar] de la pantalla Aprobar solicitud de reparación y mantenimiento.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Solicitudes aprobadas] de la pantalla Aprobar solicitud de reparación y mantenimiento.</p>

TABLA XCII.
DESCRIPCIÓN DEL CASO DE USO APROBAR REQUISICIÓN.

Caso de Uso:	Aprobar requisición.			Actor:	Administrador.
Código:	CU019	Tipo:	Primario-Sistema.	Pantallas:	P007, P007.01, P007.03
Ref. Req:	RF014.				
Resumen:	El Administrador, busca y selecciona la requisición que va aprobar.				

Objetivos:	<ul style="list-style-type: none"> • Buscar requisición • Aprobar requisición. • Agregar requisición a Kardex.
Precondicion es:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. • Haya ingresado a las pantallas Asignar Requisición.
PosCondicion es:	<ul style="list-style-type: none"> • Requisición aprobada y agregada en kardex.
Curso normal de eventos.	
<ol style="list-style-type: none"> 1. El Administrador ingresa los criterios de búsqueda (fecha de requisición, número de requisición) para una requisición y presiona el botón [Buscar] en la pantalla Aprobar Requisición. 2. El sistema actualiza la tabla [Requisiciones no aprobadas], con la requisición que coincida con el criterio de búsqueda en la pantalla Aprobar Requisición. 3. El Administrador verifica que exista la requisición en la tabla [Requisiciones no aprobadas] y presiona el botón [Aprobar] en la pantalla Aprobar Requisición. 4. El sistema invoca la pantalla Aprobar Requisición. 5. El administrador presiona el botón [Guardar], de la pantalla Aprobar Requisición. 6. El sistema invoca la pantalla Crear/Editar kardex. 7. El caso de uso finaliza. 	
Curso alternos de eventos.	
<p>A. Limpiar Búsqueda.</p> <p>A.1. El Administrador presiona el botón [Limpiar] de la pantalla Aprobar Requisición.</p> <p>A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Requisiciones no aprobadas] de la pantalla Aprobar Requisición.</p>	

TABLA XCIII.

DESCRIPCIÓN DEL CASO DE USO GESTIONAR PLAN DE MANTENIMIENTO.

Caso de uso:	Gestionar plan de	Actor:	Administrador
---------------------	-------------------	---------------	---------------

	Mantenimiento.				
Código:	CU020	Tipo:	Primario-Sistema	Pantallas:	P008, P008.01, P008.02, P008.03, P009.04, P009.05.
Ref. Req:	RF015.				
Resumen:	El Administrador, visualiza los planes de mantenimiento, para poder editarlo o crearlo en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar Plan de Mantenimiento. • Crear, Editar, activar plan, visualizar un plan de mantenimiento. 				
Precondiciones:	<ul style="list-style-type: none"> • El Administrador se haya autenticado. 				
PosCondiciones:	<ul style="list-style-type: none"> • Plan de mantenimiento creado. 				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El Administrador del sistema ingresa los criterios de búsqueda (registro, fecha de creación) para un plan de mantenimiento y presiona el botón [Buscar] en la pantalla Buscar Plan de Mantenimiento. 2. El sistema actualiza la tabla [Resultados de Búsqueda] con los planes que coincidan con el criterio de búsqueda. 3. El Administrador verifica que exista el plan de mantenimiento en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear] de la pantalla Buscar Plan de Mantenimiento. 4. El sistema invoca la pantalla Crear/Editar Plan de Mantenimiento. 5. El Administrador ingresa la información general (registro, fecha de creación) y las actividades por kilometraje (kilometraje, actividades), presiona el botón [Crear actividades] del plan del mantenimiento y presiona el botón [Guardar] de la pantalla Crear/Editar Plan de Mantenimiento. 6. El sistema guarda al nuevo plan de mantenimiento y presenta la pantalla 					

Buscar Plan de Mantenimiento.

7. El caso de uso finaliza.

Curso alterno de eventos.

A. Limpiar Búsqueda.

A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Plan de Mantenimiento.**

A.2. El sistema limpia el criterio de búsqueda y actualiza y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Plan de Mantenimiento.**

B. Editar.

B.3. El administrador elige el plan de mantenimiento a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Plan de Mantenimiento.**

B.4. El sistema recupera la información del plan de mantenimiento a editar, y la presenta en la pantalla **Crear/Editar Plan de Mantenimiento.**

B.5. El Administrador modifica la información general (registro, fecha de creación) y las actividades por kilometraje (kilometraje, actividades), presiona el botón [Crear actividades] del plan del mantenimiento y presiona el botón [Guardar] de la pantalla **Crear/Editar Plan de Mantenimiento.**

B.6. El sistema guarda al nuevo plan de mantenimiento presenta la pantalla **Buscar Plan de Mantenimiento.**

C. Campos Obligatorios Vacíos Plan de Mantenimiento.

C.6 - (C.6, C.B.6). El sistema verifica que estén llenos los campos obligatorios (registro, fecha de creación, kilometraje, actividades) y presenta un mensaje "Campos obligatorios vacíos" continúa (5, B.5).

D. Activar Plan.

D.3. El administrador elige el plan de mantenimiento a activar de la tabla [Resultados de Búsqueda] y presiona el botón [Activar Plan] de la pantalla **Buscar Plan de Mantenimiento.**

D.4. El sistema invoca la pantalla **Activar Plan de Mantenimiento**

D.5. El administrador presiona el botón [Sí], de la pantalla **Activar Plan de Mantenimiento.**

D.6. El sistema presenta un mensaje “El Plan de mantenimiento seleccionado ha sido activado”.
E. Visualizar plan activo.
E.3. El administrador presiona el botón [Visualizar plan activo] de la pantalla Buscar Plan de Mantenimiento .
E.4 El sistema muestra las próximas actividades del mantenimiento a realizar del vehículo seleccionado.

TABLA XCIV.

DESCRIPCIÓN DEL CASO DE USO GESTIONAR PERSONAL

Caso de Uso:	Gestionar personal.			Actor:	Administrador
Código:	CU022.	Tipo:	Primario-Sistema.	Pantalla:	P010, P010.01, P010.02.
Ref. Req:	RF018.				
Resumen:	El Administrador, visualiza el personal registrado para poder editarlo o crearlo en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar Personal. • Crear, Editar un personal o crear una cuenta de usuario para un personal. 				
Precondiciones:	<ul style="list-style-type: none"> • El administrador se haya autenticado. 				
PosCondiciones:	<ul style="list-style-type: none"> • Personal y/o usuario creado. 				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El Administrador del sistema ingresa los criterios (nombres, apellidos, número de cédula, tipo) de búsqueda para un personal y presiona el botón [Buscar] en la pantalla Buscar Personal. 2. El sistema actualiza la tabla [Resultados de Búsqueda] con las personas que coincidan con el criterio de búsqueda. 					

3. El Administrador verifica que exista el personal en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear] de la pantalla **Buscar Personal**.
4. El sistema invoca la pantalla **Crear/Editar Personal**.
5. El Administrador ingresa la información (nombres, apellidos, número de cédula, tipo, fecha de nacimiento, email, cargo) del personal y presiona el botón [Guardar] de la pantalla **Crear/Editar Personal**.
6. El sistema guarda al nuevo personal y presenta la pantalla **Buscar Personal**.
7. El caso de uso finaliza

Curso alterno de eventos.

A. Limpiar Búsqueda.

- A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Personal**.
- A.2. El sistema limpia el criterio de búsqueda y actualiza en la tabla [Resultados de búsqueda] de la pantalla **Buscar Personal**.

B. Editar personal.

- B.3. El administrador elige el personal a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Personal**.
- B.4. El sistema recupera la información del personal a editar, y la presenta en la pantalla **Crear/Editar Personal**.
- B.5. El Administrador modifica la información (nombres, apellidos, número de cédula, tipo, fecha de nacimiento, email, cargo) del personal y presiona el botón [Guardar] de la pantalla **Crear/Editar Personal**.
- B.6. El sistema guarda al nuevo personal y presenta la pantalla **Buscar Personal**.

C. Campos obligatorios vacíos del personal.

- C.6. - (C.B.6) El sistema verifica que estén llenos los campos obligatorios (nombres, apellidos, número de cédula, tipo, fecha de nacimiento, email) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Personal** y continúa (5, B.5).

D. Número de cédula duplicado del personal.

- D.5 – (D.B.5) El sistema verifica si el número de cédula del personal esta duplicado

y presenta un mensaje “Número de cedula invalido o El DNI ingresado pertenece a otra persona” en la pantalla **Crear/Editar Personal** y continúa (5, B.5).

E. Crear Cuenta de Usuario.

E.3. El administrador elige el personal de la tabla [Resultados de Búsqueda] y presiona el botón [Crear Cuenta de Usuario] de la pantalla **Buscar Personal**.

E.4. El sistema invoca la pantalla **Crear/Editar Usuario**.

E.5. El Administrador ingresa la información (usuario, contraseña, confirmar contraseña) para el usuario presiona el botón [Guardar] de la pantalla **Crear/Editar Usuario**.

E.6. El sistema guarda al nuevo usuario y presenta la pantalla **Buscar Personal**.

F. Editar Usuario.

F.3. El administrador elige el personal a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Crear Cuenta de Usuario] de la pantalla **Buscar Personal**.

F.4. El sistema recupera la información del usuario a editar, y presenta la pantalla **Crear/Editar Cuenta de usuario**

F.5. El Administrador modifica la información (usuario, contraseña, confirmar contraseña) del personal y presiona el botón [Guardar] de la pantalla **Crear/Editar Cuenta de Usuario**.

F.6. El sistema guarda al nuevo usuario y presenta la pantalla **Buscar Personal**.

G. Campos obligatorios vacíos de usuario.

G.6.- (G.F.6) El sistema verifica que estén llenos los campos obligatorios (usuario, contraseña, confirmar contraseña) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla **Crear/Editar Personal** y continúa (5, F.5).

H. Usuario duplicado.

H.6 – (H.F.6) El sistema verifica si el usuario esta duplicado y presenta un mensaje “El nombre de usuario no está disponible. Pertenece a otro usuario” en la pantalla **Crear/Editar Personal** y continúa (5, F.5).

TABLA XCV.

DESCRIPCIÓN DEL CASO DE USO GESTIONAR FUNCIONALIDAD

Caso de Uso:	Gestionar funcionalidades.			Actor:	Administrador
Código:	CU023	Tipo:	Primari o- Sistema .	Pantallas :	P011, P011.01, P011.02, P011.03, P011.04, P011.05, P011.06, P011.07, P011.08, P011.09.
Ref. Req:	RF019.				
Resumen:	El Administrador, visualiza las entidades para crear y visualiza las propiedades para poder desactivar, editar o crear en caso que no exista.				
Objetivos:	<ul style="list-style-type: none"> • Buscar entidad y propiedad. • Crear una entidad. • Crear y editar una propiedad. 				
Precondiciones:	<ul style="list-style-type: none"> • El administrador se haya autenticado. 				
PosCondiciones:	<ul style="list-style-type: none"> • Entidad creada y propiedad creada o editada. 				
Curso normal de eventos.					
<p>12. El Administrador ingresa los criterios de búsqueda (nombre de la entidad, etiqueta de la entidad) para una entidad y presiona el botón [Buscar] en la pantalla Buscar Entidad.</p> <p>13. El sistema actualiza la tabla [Resultados de búsqueda], con la entidad que coincidan con el criterio de búsqueda.</p> <p>14. El Administrador selecciona la entidad de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla Buscar Entidad.</p>					

15. El sistema invoca a la pantalla **Buscar Propiedad**.
16. El Administrador ingresa los criterios de búsqueda (nombre de la propiedad, etiqueta de vista) para una propiedad y presiona el botón [Buscar] en la pantalla **Buscar Propiedad**.
17. El sistema actualiza la tabla [Resultados de búsqueda], con la propiedad que coincidan con el criterio de búsqueda.
18. El Administrador verifica que exista la propiedad en la tabla [Resultados de búsqueda] en caso de que no exista presiona el botón [Crear Propiedad] de la pantalla **Buscar Propiedad**.
19. El sistema invoca a la pantalla **Crear/Editar Propiedad**.
20. El Administrador registra los datos generales (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) de la propiedad y presiona el botón [Guardar] de la pantalla **Crear/Editar Propiedad**.
21. Si el Administrador crear una propiedad de tipo de datos **entidad**, el sistema presenta en la pantalla **Buscar Propiedad** el botón [Crear Entidad].
22. El caso de uso finaliza.

Curso alternativo de eventos.

J. Limpiar búsqueda entidad.

- A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Entidad**.
- A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Entidad**.

K. Limpiar búsqueda propiedad.

- B.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Propiedad**.
- B.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Propiedad**.

L. Editar propiedad.

- C.7. El Administrador selecciona la propiedad a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Propiedad**.
- C.8. El sistema recupera la información de la propiedad y la muestra en la pantalla **Crear/Editar Propiedad**.

C.9. El Administrador edita los datos correspondientes (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de vista, texto de ayuda, posición en vista, requerido) de la propiedad y presiona el botón [Guardar] de la pantalla **Crear/Editar Propiedad**.

C.10. El sistema guarda la nueva propiedad con su respectiva información y presenta la pantalla **Buscar Propiedad**.

M. Crear entidad.

D.7. El administrador elige la propiedad a crear como entidad de la tabla [Resultados de Búsqueda] y presiona el botón [Crear entidad], en la pantalla **Buscar propiedad**

D.8. El sistema invoca la pantalla **Crear/Editar Entidad**.

D.9. El Administrador registra los datos correspondientes (descripción) de la Entidad y presiona el botón [Guardar] de la pantalla **Crear/Editar Entidad**.

D.10. El sistema guarda la entidad con su respectiva información y presenta la pantalla **Buscar Propiedad** para agregar propiedades.

N. Desactivar propiedad.

E.7. El administrador elige la propiedad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Propiedad**.

E.4. El sistema invoca pantalla **Desactivar propiedad**.

E.5. El administrador presiona el botón [Sí], de la pantalla **Desactivar propiedad**.

E.6. El sistema presenta un mensaje “La propiedad seleccionada ha sido desactivada exitosamente”.

O. Desactivar entidad.

F.3. El administrador elige la entidad a desactivar de la tabla [Resultados de Búsqueda] y presiona el botón [Desactivar] de la pantalla **Buscar Entidad**.

F.4. El sistema invoca la pantalla **Desactivar entidad**.

F.5. El administrador presiona el botón [Sí], de la pantalla **Desactivar entidad**.

F.6. El sistema presenta un mensaje “La entidad seleccionada ha sido desactivada”.

P. Campos obligatorios y vacíos de propiedad.

G.10.- (G.C.10) El sistema verifica que estén llenos los campos obligatorios (nombre del grupo, nombre de la propiedad, tipo de dato, valor defecto, etiqueta de

vista, texto de ayuda, posición en vista, requerido) y presenta un mensaje “Campos obligatorios vacíos” en la pantalla **Crear/Editar Propiedad** y continúa (9, C.9).

Q. No puede crear entidad

H.D.8. El sistema verifica que la propiedad escogida sea de tipo estructura y no haya sido creada como entidad, si no es así presenta el mensaje “Solamente se puede crear como entidad a las propiedad tipos estructura o esta propiedad ya ha sido creada como entidad”.

TABLA XCVI.

DESCRIPCIÓN DEL CASO DE USO INGRESAR AL SISTEMA.

Caso de uso:	Ingresar al sistema			Actor:	Administrador/ Secretario
Código:	CU024	Tipo:	Primario- Sistema	Pantallas:	P012.
Ref. Req:	RF020.				
Resumen:	El administrador o secretario, ingresaran al sistema con su usuario y contraseña.				
Objetivos:	<ul style="list-style-type: none"> • Ingresar al sistema 				
Precondiciones:	<ul style="list-style-type: none"> • El administrador o secretario haya ingresado a la pantalla de inicio. 				
PosCondiciones :	<ul style="list-style-type: none"> • El administrador o secretario logeado. 				
Curso normal de eventos.					
<ol style="list-style-type: none"> 1. El administrador del sistema o secretario ingresan los parámetros (usuario, contraseña) y presiona el botón [Ingresar] en la pantalla Ingresar al sistema. 2. El sistema guarda el usuario registrado y presenta la pantalla de Inicio. 3. El caso de uso finaliza. 					

Curso alternativo de eventos.
<p>A. Campos Obligatorios Vacíos.</p> <p>A.1 El sistema verifica que estén llenos los campos obligatorios (usuario y contraseña) y presenta un mensaje “Campos obligatorios vacíos” y continúa (1).</p>
<p>B. Usuario inactivo</p> <p>B.1 El sistema verifica que el usuario ingresado se encuentre activado caso contrario presenta un mensaje “El usuario está inactivo o no existe” y continúa (1).</p>
<p>C. Usuario y contraseña incorrecta</p> <p>C.1 El sistema verifica que el usuario y contraseña ingresado sean correctos caso contrario presenta un mensaje “Nombre de usuario y contraseña incorrectos” y continúa (1).</p>

3.3. Requerimientos no funcionales.

3.3.1. Requerimientos de rendimiento.

TABLA XCVII.

REQUERIMIENTO NO FUNCIONAL 1

Número de requisito	RNF001
Nombre del requisito	El sistema permitirá dos tipos de usuarios, permitiendo un manejo más eficiente del sistema.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA XCVIII.

REQUERIMIENTO NO FUNCIONAL 2

Número de requisito	RNF002
Nombre del requisito	El tiempo máximo de respuesta del sistema en cada función que el usuario solicite será de máximo 8 segundos.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.3.2. Requerimientos de seguridad.

TABLA XCIX.
REQUERIMIENTO NO FUNCIONAL 3

Número de requisito	RNF003
Nombre del requisito	La información de los reportes deberá ser clara y precisa de acuerdo al formato que el usuario solicite.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.3.3. Requerimientos de fiabilidad.

TABLA C.
REQUERIMIENTO NO FUNCIONAL 4

Número de requisito	RNF004
Nombre del requisito	El sistema deberá evitar el ingreso de información incorrecta por medio de mensajes que genere el sistema.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.3.4. Requerimientos de disponibilidad.

TABLA CI.
REQUERIMIENTO NO FUNCIONAL 5

Número de requisito	RNF005
Nombre del requisito	En caso de que el usuario olvide su usuario/contraseña podrá contactar al administrador quien le proveerá una nueva.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA CII.
REQUERIMIENTO NO FUNCIONAL 6

Número de requisito	RNF006
Nombre del requisito	El sistema podrá ser usado en toda la jornada laboral del usuario.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

TABLA CIII.
REQUERIMIENTO NO F2UNCIONAL 7

Número de requisito	RNF007
Nombre del requisito	La información ingresada en el sistema podrá ser visualizada, cambiada y usada para obtener reportes por el usuario registrado
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.3.5. Requerimiento funcional de mantenibilidad.

TABLA CIV.
REQUERIMIENTO NO FUNCIONAL 8

Número de requisito	RNF008
Nombre del requisito	El sistema permitirá que el administrador pueda gestionar funcionalidades de acuerdo a los requerimientos futuros.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

3.3.6. Requerimiento funcional de portabilidad.

TABLA CV.
REQUERIMIENTO NO FUNCIONAL 9

Número de requisito	RNF009
Nombre del requisito	El sistema será desarrollado en su totalidad en Java Enterprise Edition 6 y con PostgreSQL como base de datos debido a su licencia GPL para evitar costos y problemas de licenciamiento.
Tipo	Requisito
Prioridad del requisito	Alta/Esencial

4. Verificación de requerimientos.

Se entregó el documento de especificación de requerimientos al Ing. Juan Carlos Vinuesa quien en este caso cumple el rol de cliente en la propuesta que presentamos. Luego de la revisión realizada (ver anexo 10), informa que el documento entregado cumple con las necesidades de un usuario final, es así como proseguiremos con el avance de las demás etapas en el desarrollo de software.

Anexo 4: Interfaz gráfica

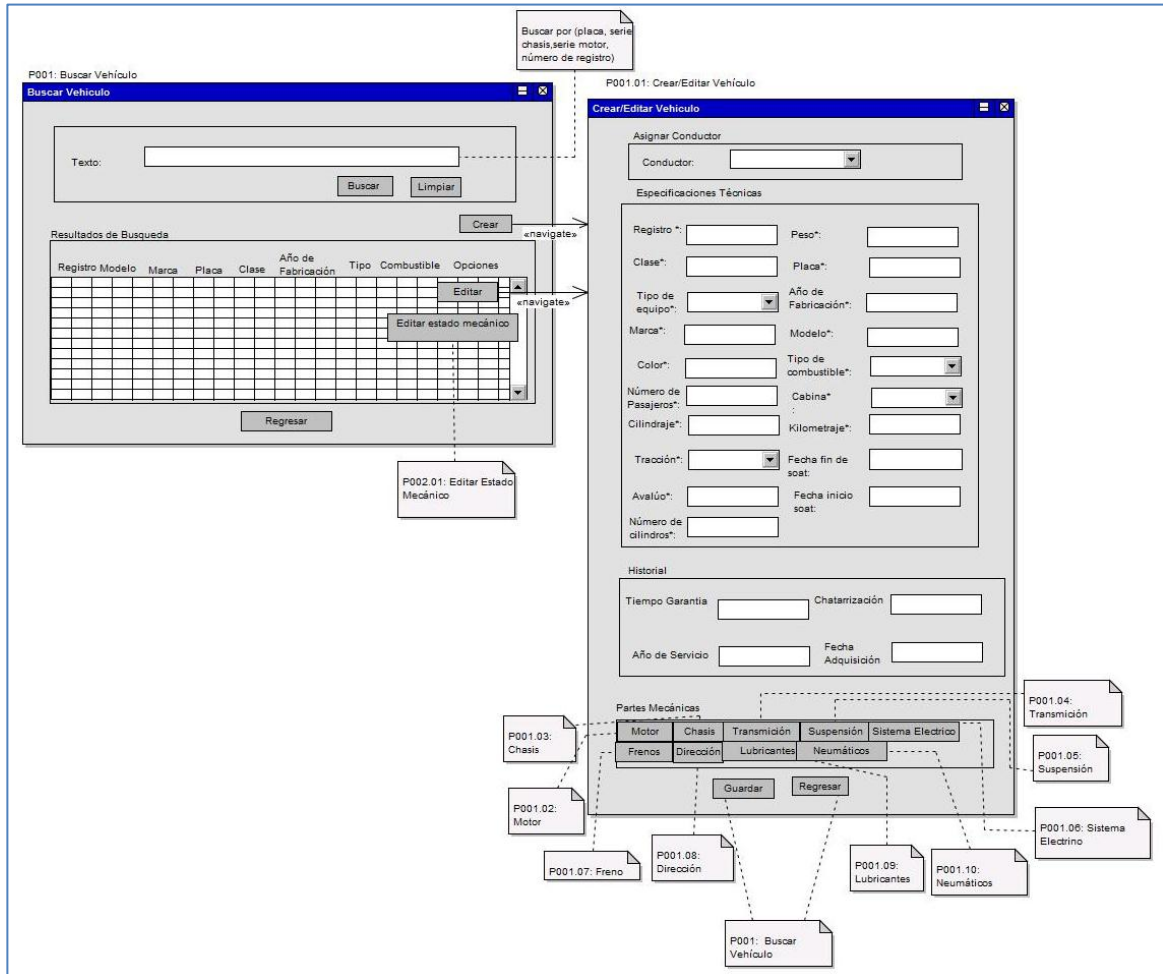


Figura 57. Prototipo del CU001

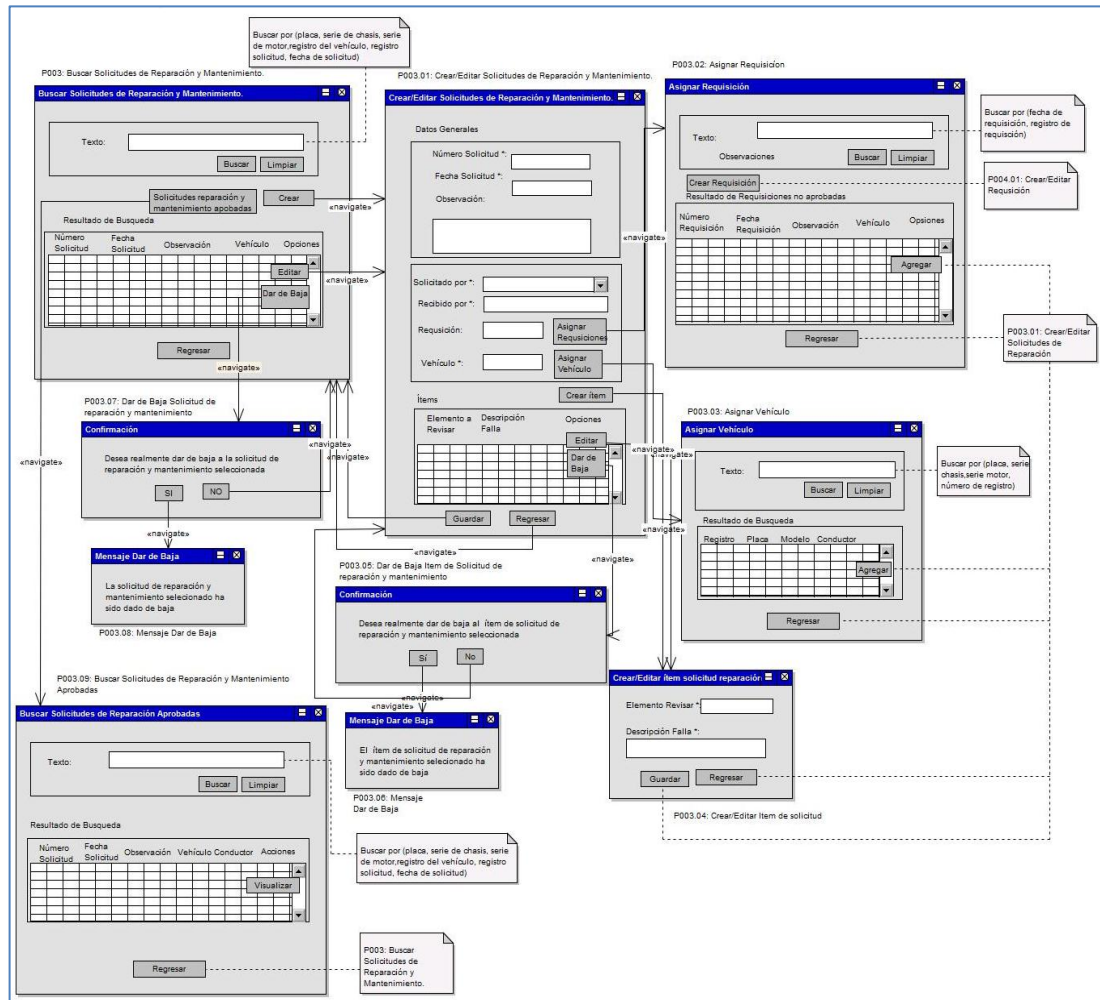


Figura 58. Prototipo del CU004, CU005, CU006, CU007, CU008

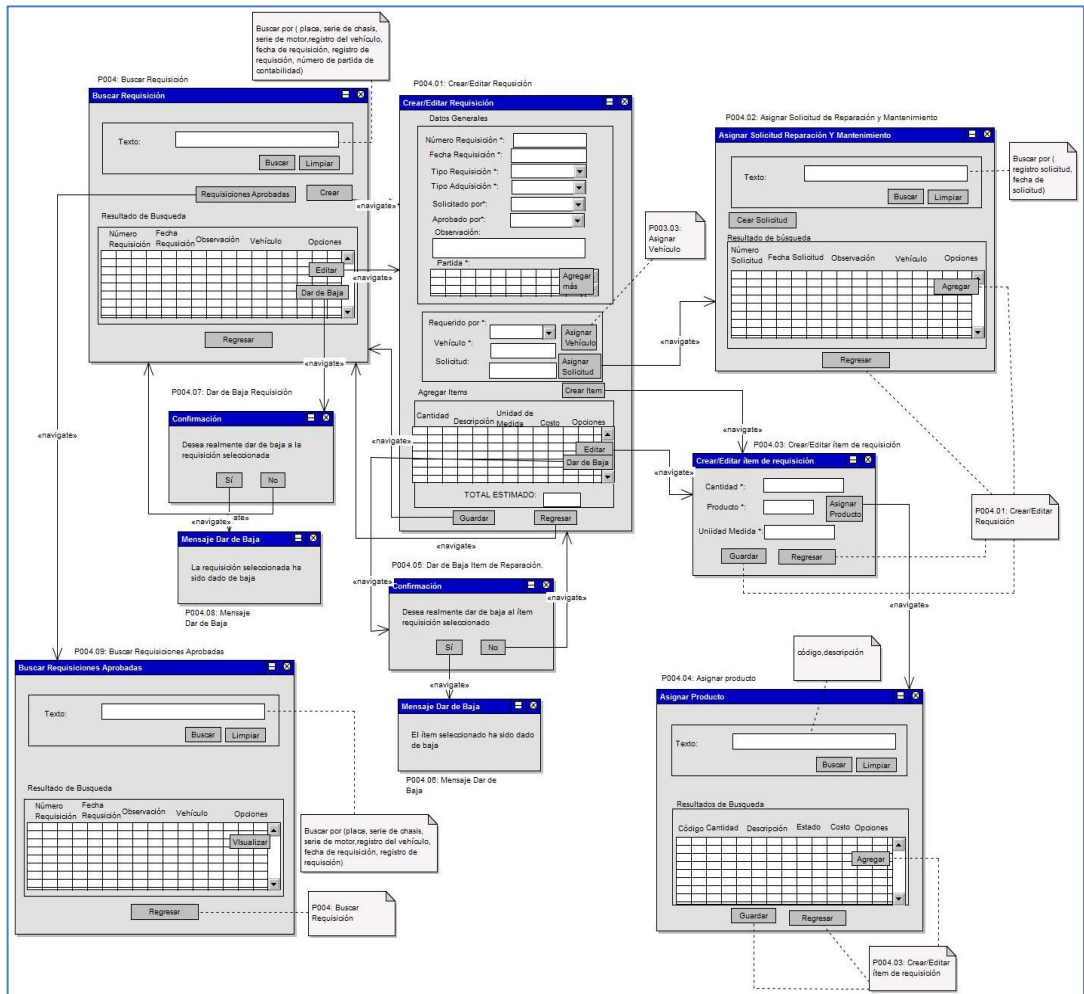


Figura 59. Prototipo del CU009, CU010, CU011, CU012, CU013, CU014

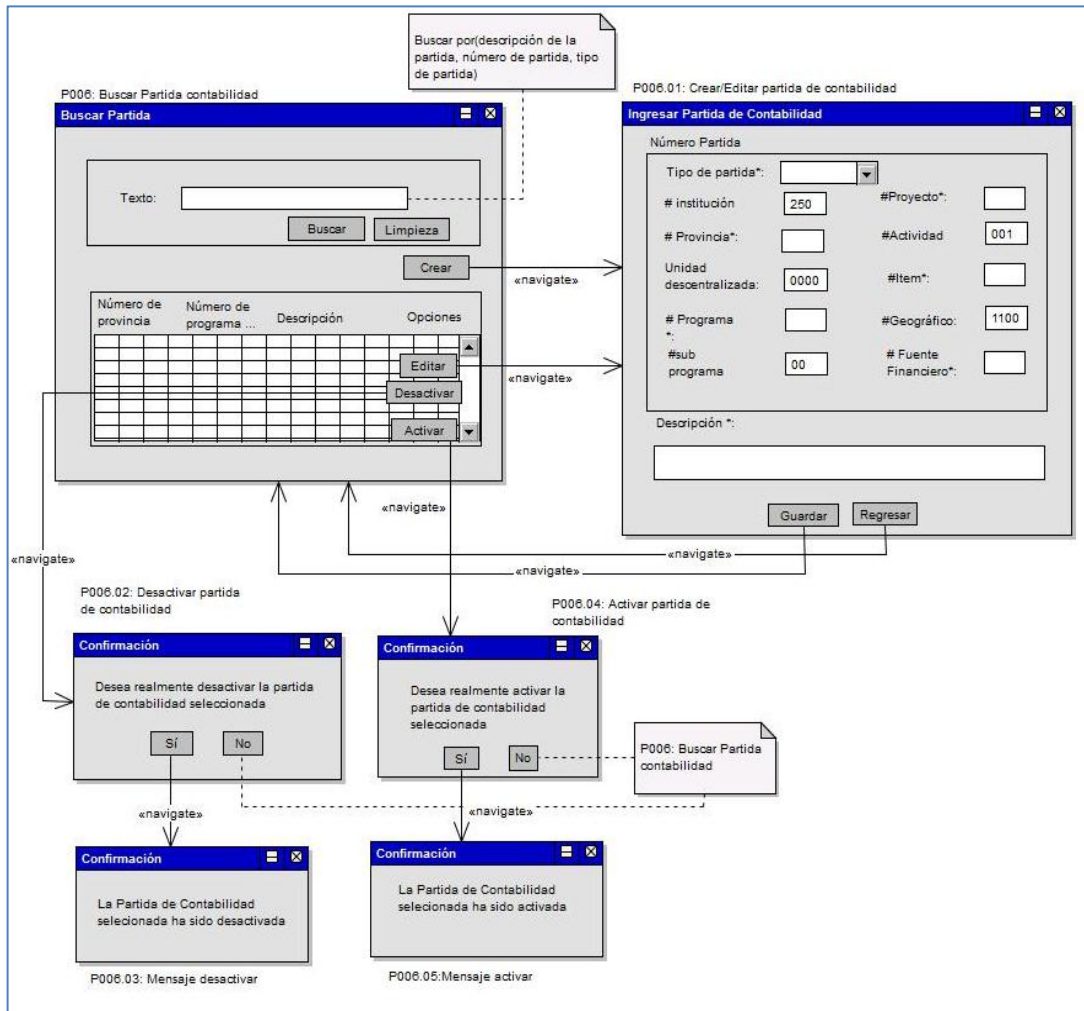


Figura 60. Prototipo del CU015

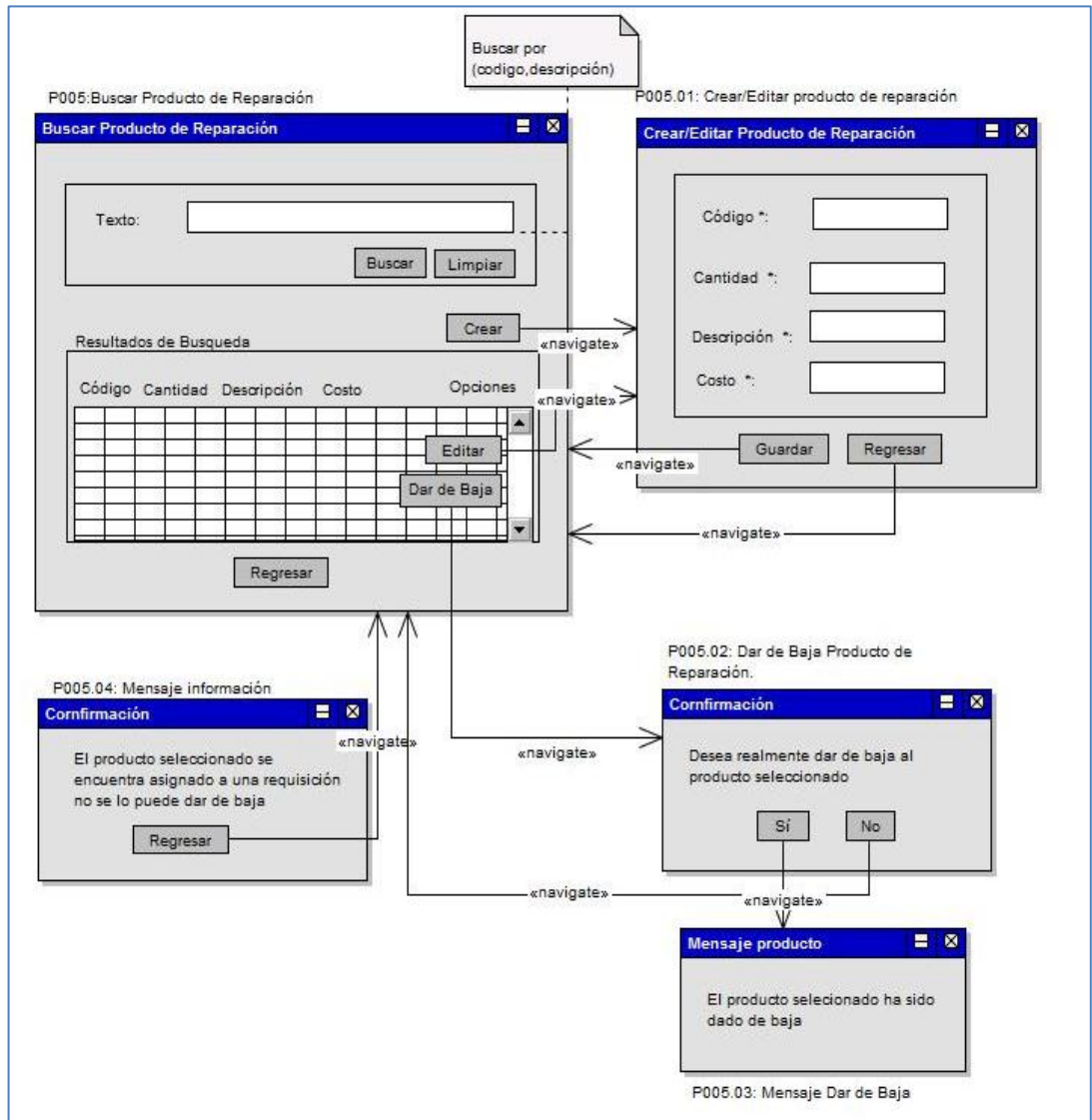


Figura 61. Prototipo del CU016

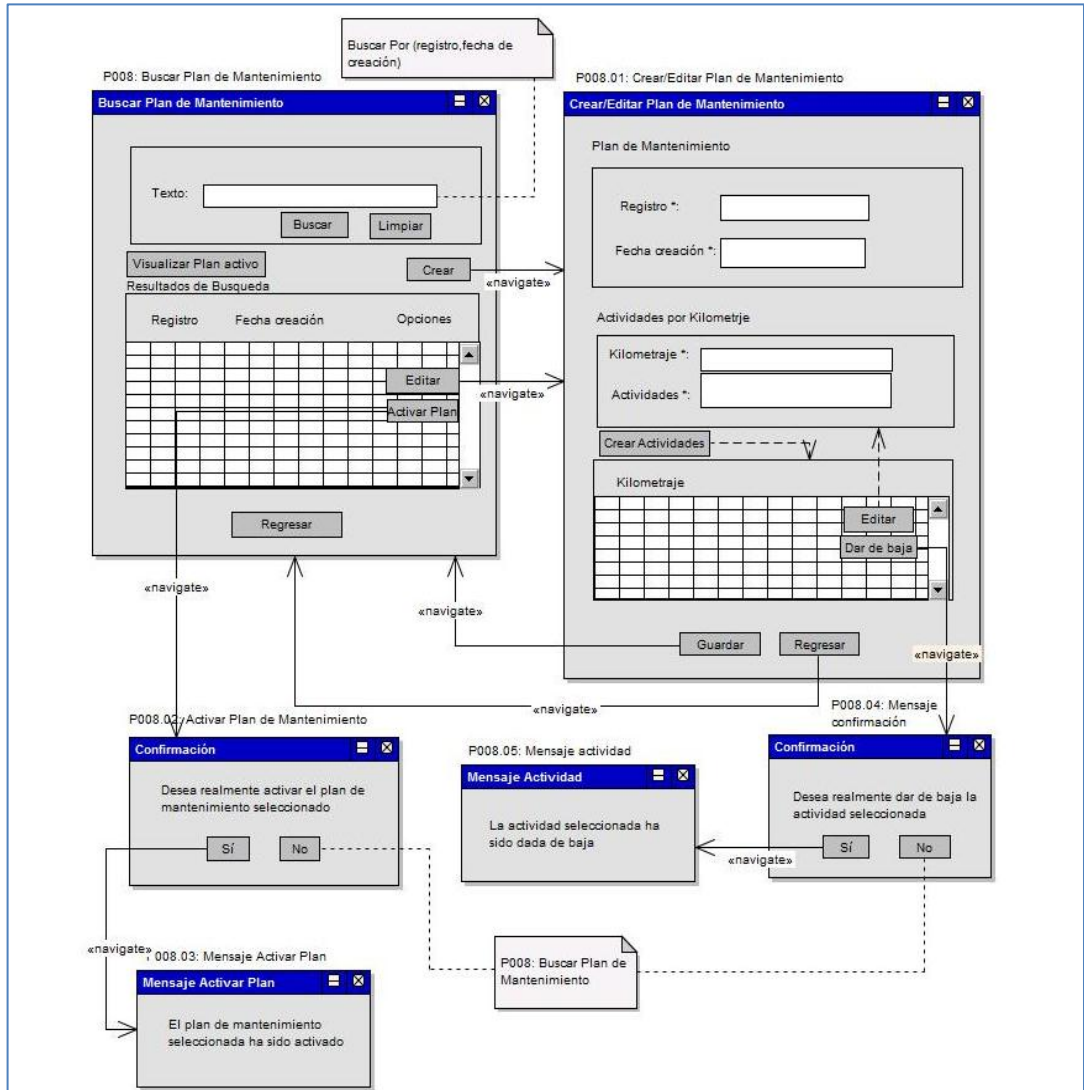


Figura 62. Prototipo del CU020

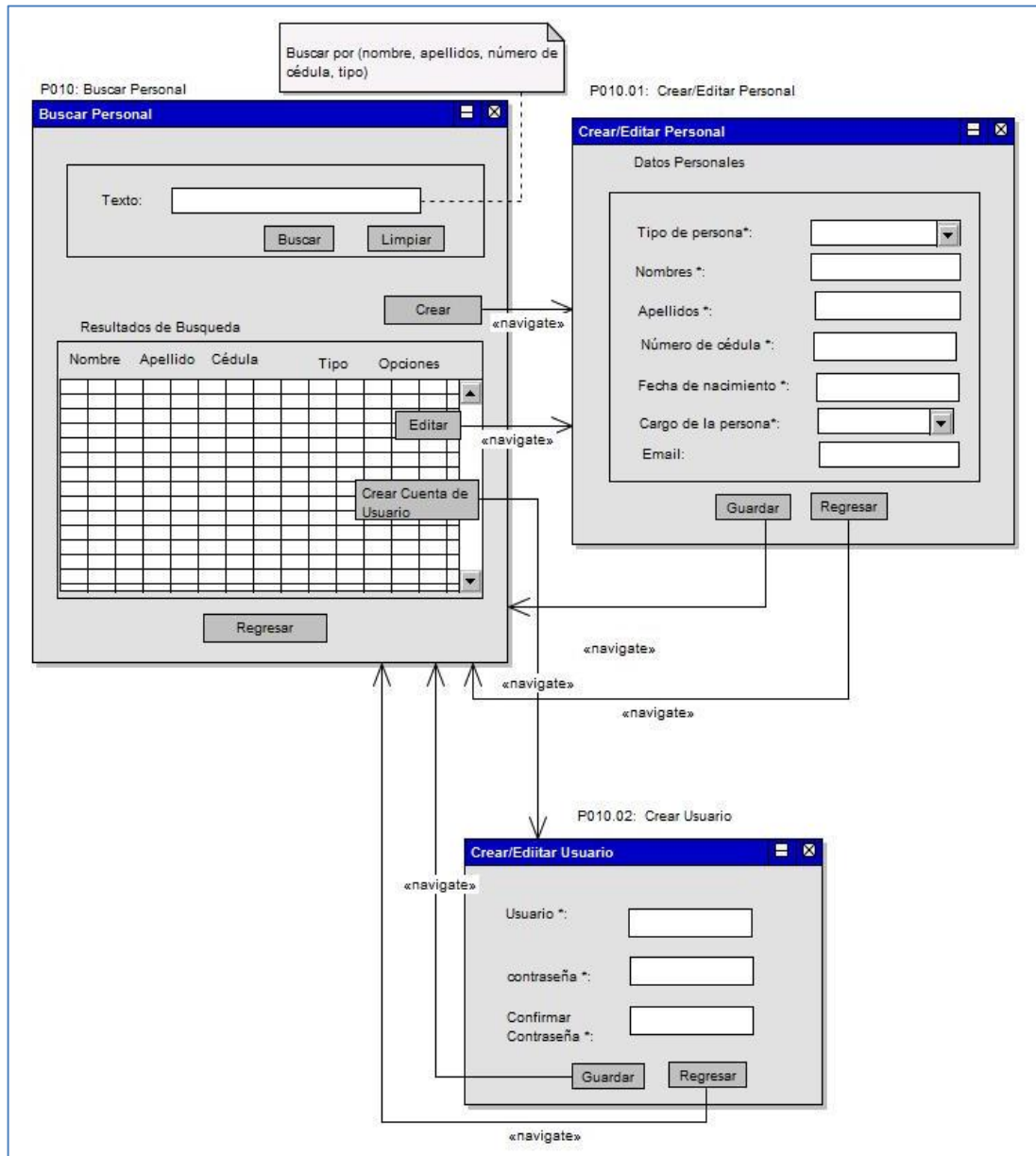


Figura 63. Prototipo del CU022

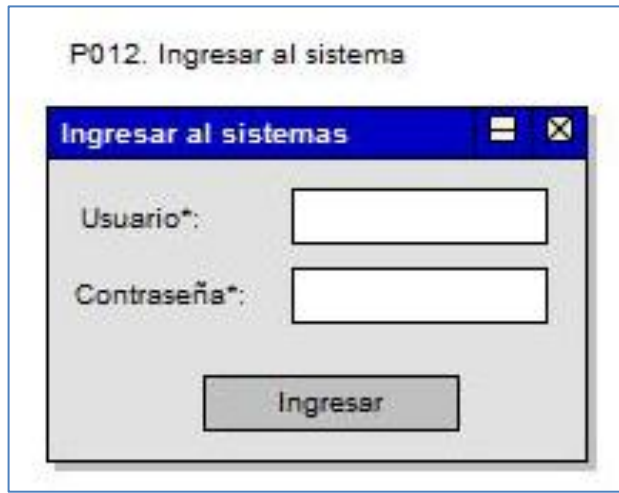


Figura 64. Prototipo del CU024

Anexo 5: Diagramas de robustez

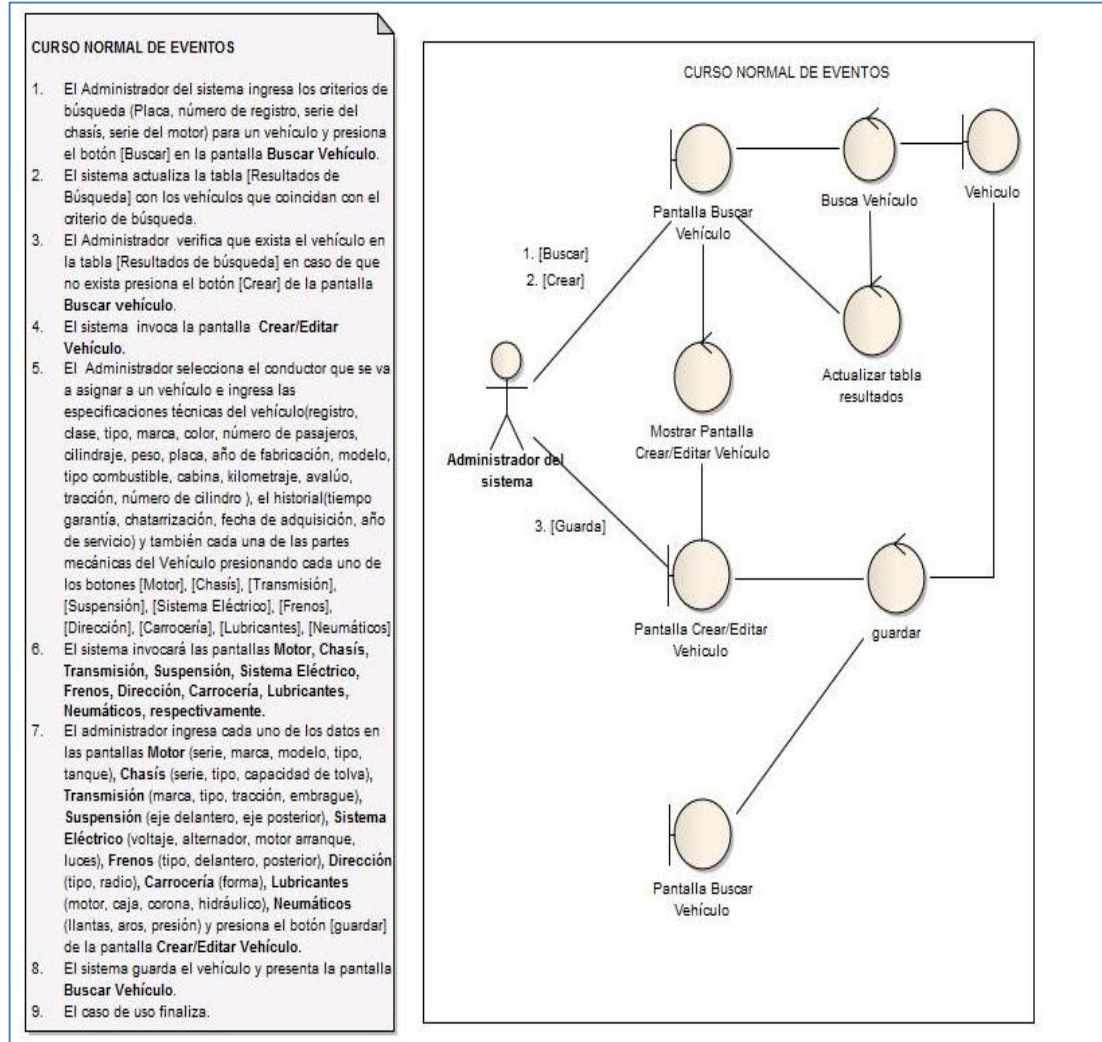


Figura 65. Diagrama de robustez CU001 curso normal eventos

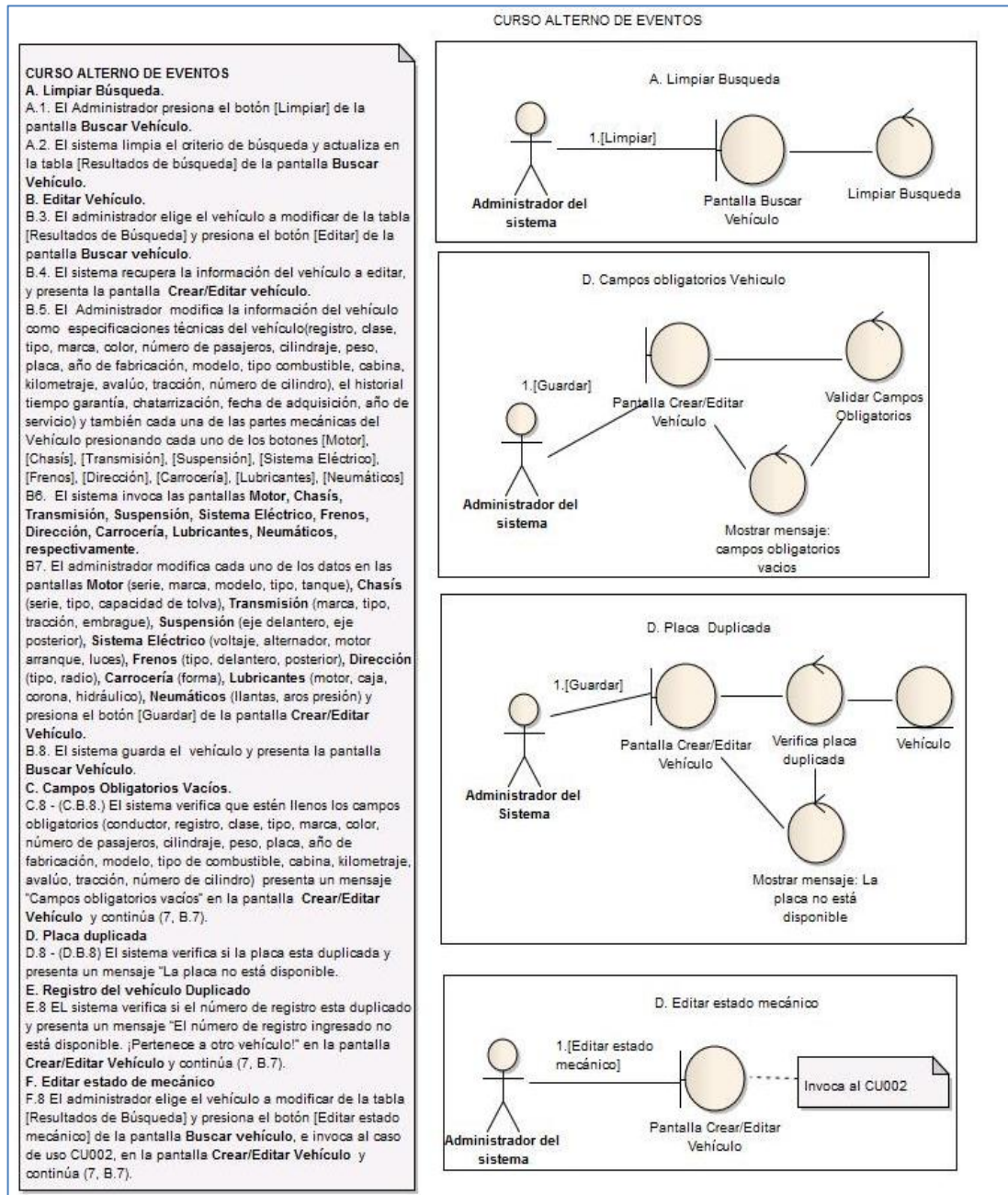


Figura 66. Diagrama de robustez CU001 curso alternativo de eventos (a).

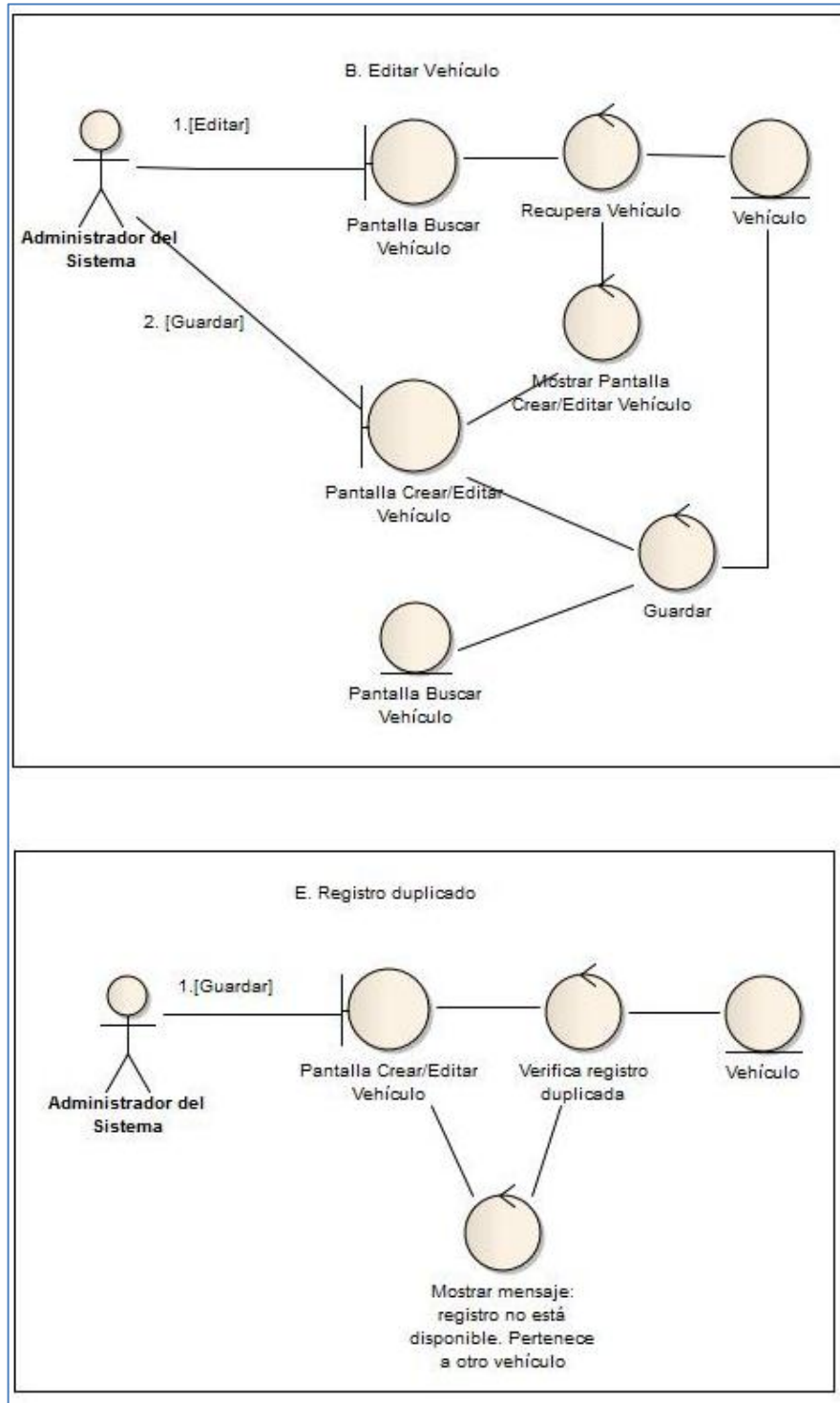


Figura 67. Diagrama de robustez CU001 curso alterno eventos (b)

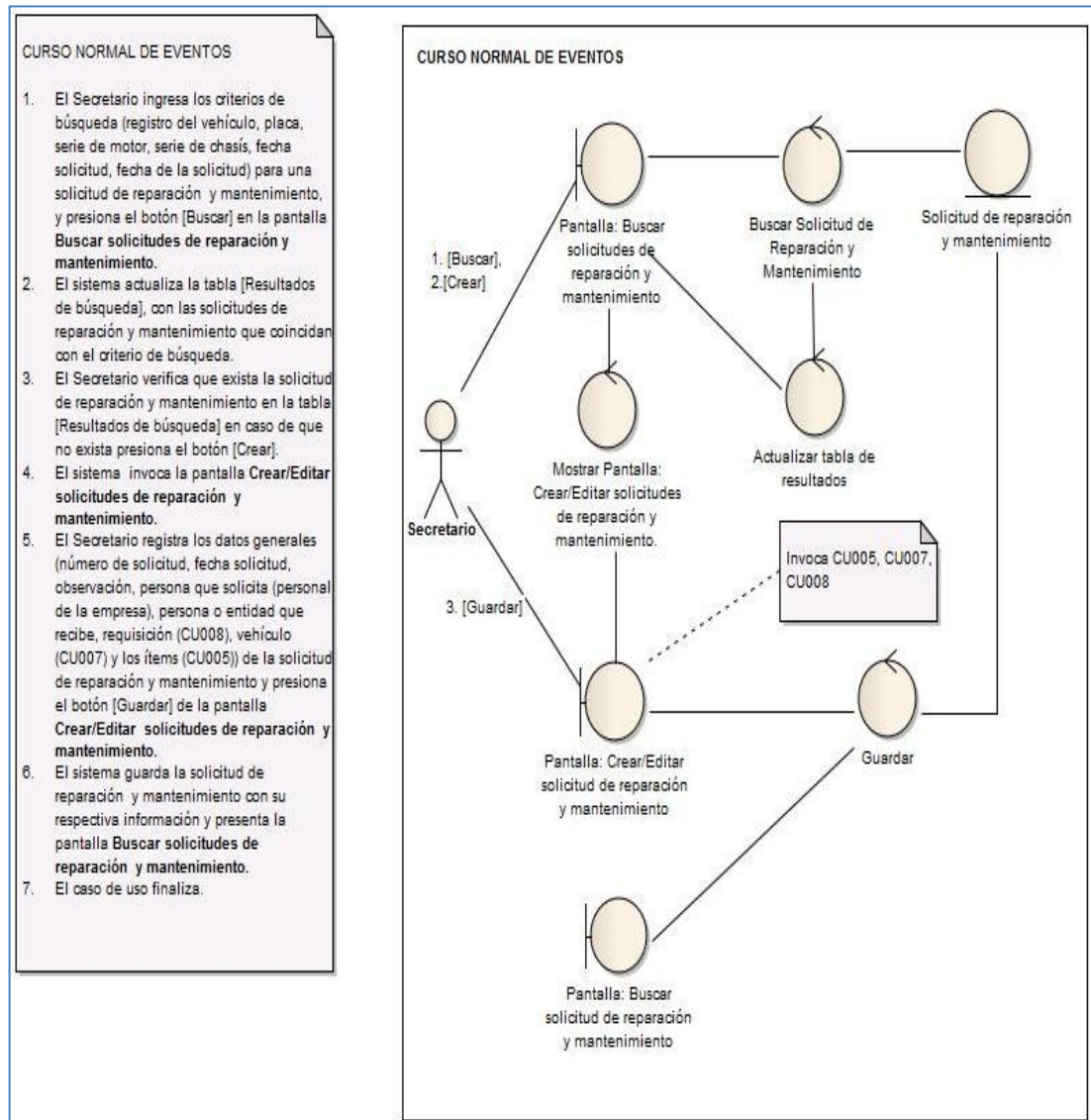


Figura 68. Diagrama de robustez CU004 curso normal eventos

CURSO ALTERNO DE EVENTOS

A. Limpiar Búsqueda.

A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar Solicitudes de Reparación y Mantenimiento**.

A.2. El sistema limpia el criterio de búsqueda de la solicitud de reparación y mantenimiento a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Solicitudes de Reparación y Mantenimiento**.

B. Editar Solicitud de Reparación y Mantenimiento.

B.3. El Secretario selecciona la solicitud de reparación y mantenimiento a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Solicitudes de Reparación y Mantenimiento**.

B.4. El sistema recupera la información de la solicitud de reparación y mantenimiento, y los muestra en la pantalla **Crear/Editar Solicitudes de Reparación y Mantenimiento**.

B.5. El Secretario edita los datos correspondientes de la solicitud de reparación y mantenimiento (número de solicitud, fecha solicitud, observación, persona que solicita (personal de la empresa), persona o entidad que recibe, persona que aprueba, requisición (CU008), vehículo (CU007) y los ítems (CU005)), y presiona el botón [Guardar] de la pantalla **Crear/Editar Solicitudes de Reparación y Mantenimiento**.

B.6. El sistema guarda la nueva solicitud de reparación y mantenimiento, con su respectiva información y presenta la pantalla **Buscar Solicitudes de Reparación y Mantenimiento**.

C. Dar de baja solicitudes de Reparación y Mantenimiento.

C.3. El secretario selecciona la solicitud de reparación y mantenimiento a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [Dar de baja] de la pantalla **Buscar Solicitudes de Reparación y Mantenimiento**.

C.4. El sistema invoca la pantalla **Dar de baja solicitud de reparación y mantenimiento**.

C.5. El secretario presiona el botón [Sí] de la pantalla **Dar de baja solicitud de reparación y mantenimiento**.

C.6. El sistema presenta un mensaje "Solicitud de reparación y mantenimiento seleccionada ha sido dada de baja".

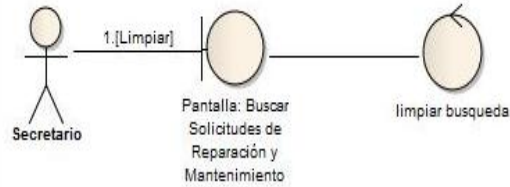
D. Visualizar Solicitudes de Reparación y Mantenimiento Aprobadas.

D.1. El secretario presiona el botón [Solicitudes de Reparación Aprobadas] de la pantalla **Buscar Solicitudes de Reparación y Mantenimiento** e invoca al caso de uso CU008.

E. Validar campos obligatorios.

E.6 - (E.B.6). El sistema verifica que estén llenos los campos obligatorios (número de solicitud, fecha de solicitud, persona que solicita, persona o entidad que recibe, vehículo y al menos un ítem) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Solicitudes de Reparación y Mantenimiento** y continúa (E. B5).

A. Limpiar Búsqueda



B. Editar Solicitudes de reparación y mantenimiento

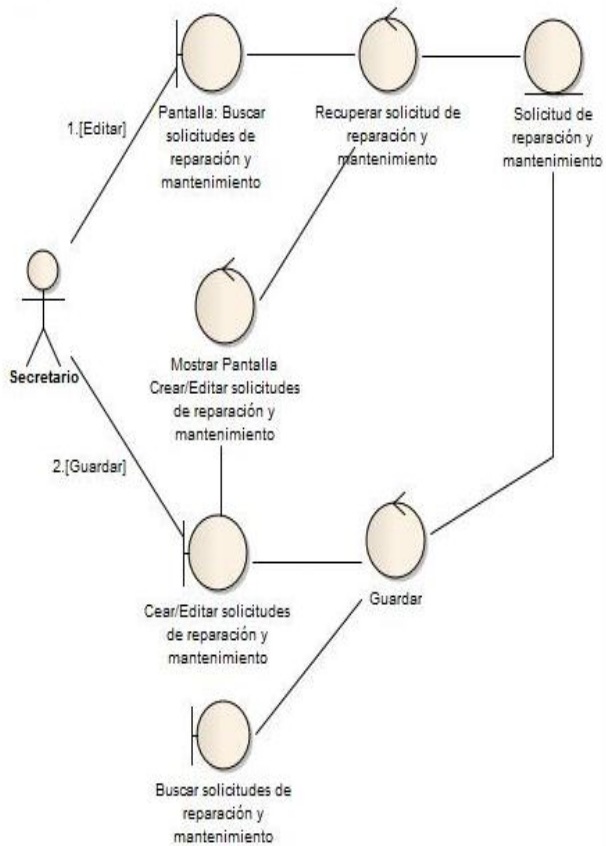


Figura 69. Diagrama de robustez CU004 curso alterno eventos(a)

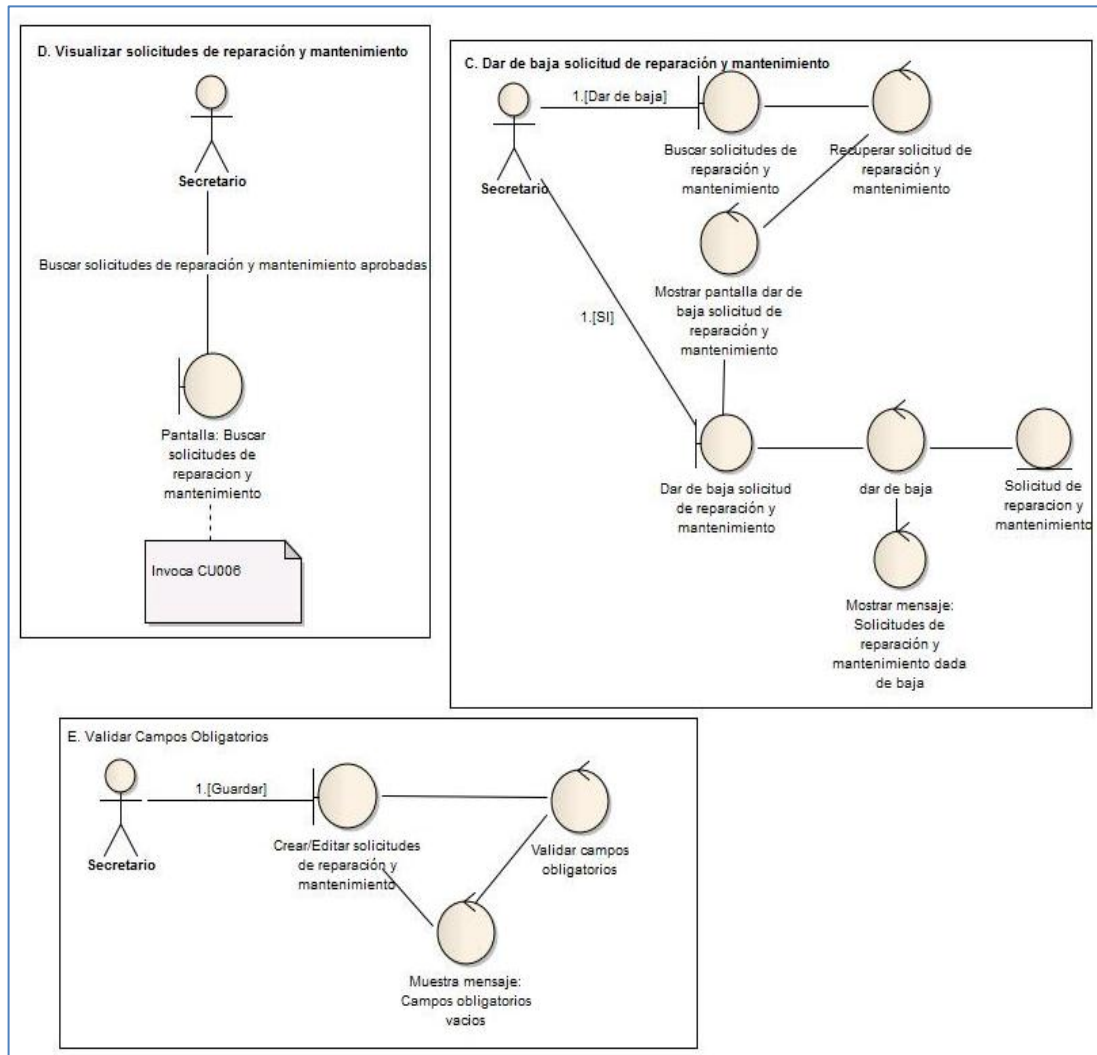


Figura 70. Diagrama de robustez CU004 curso alternativo eventos(b)

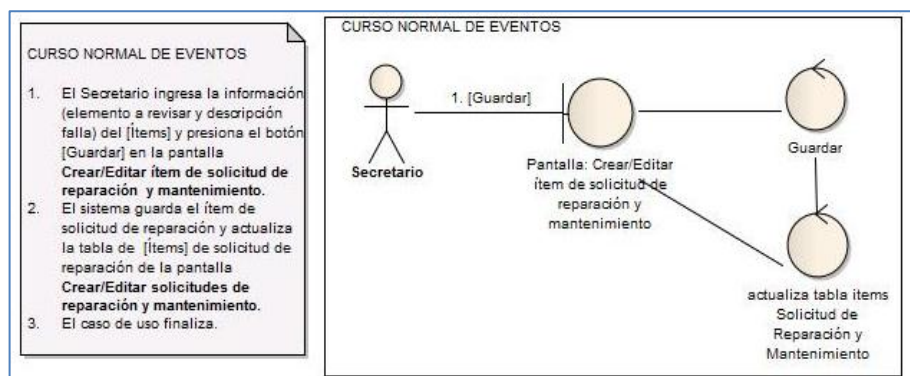


Figura 71. Diagrama de robustez CU005 curso normal eventos

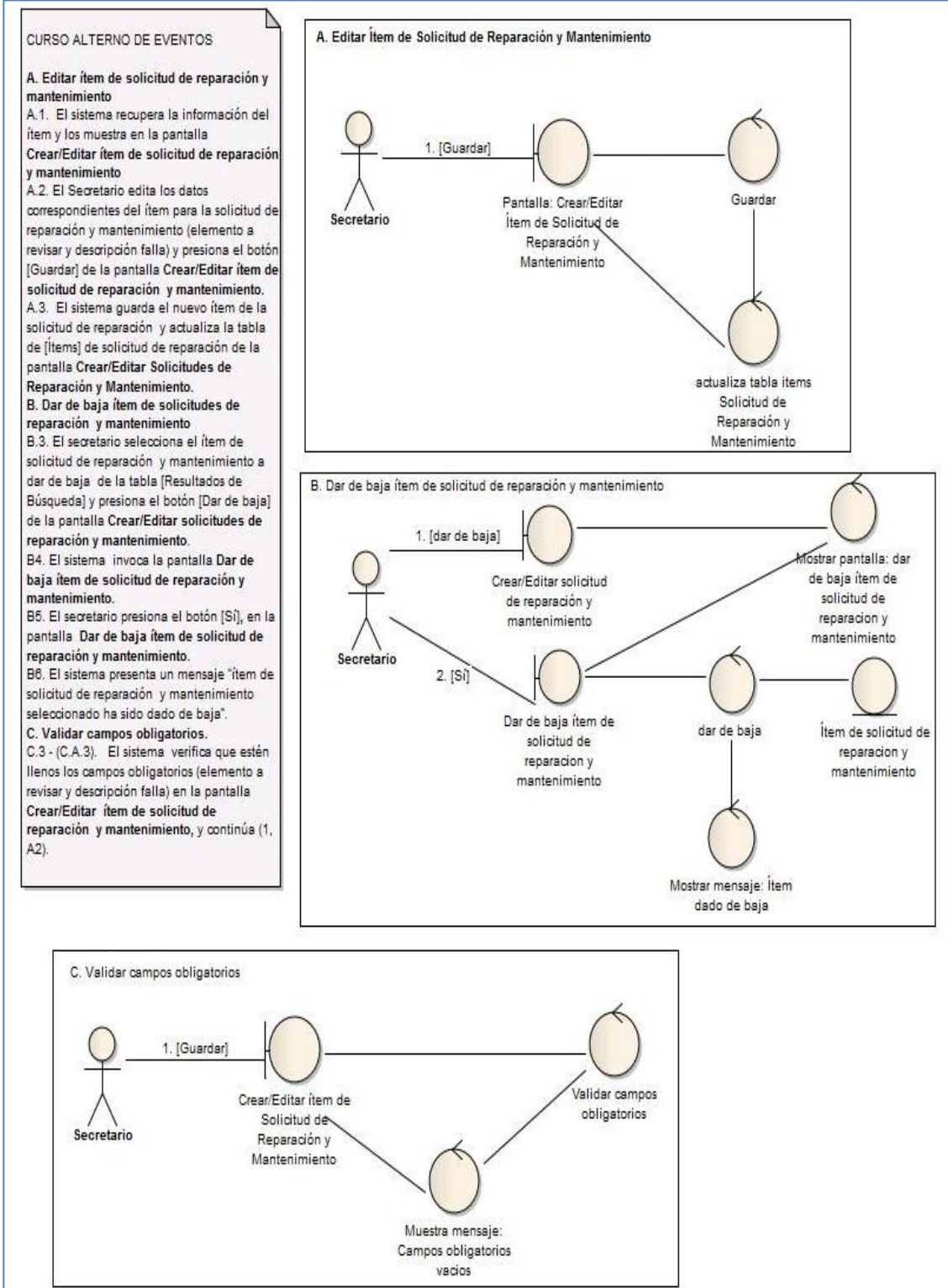
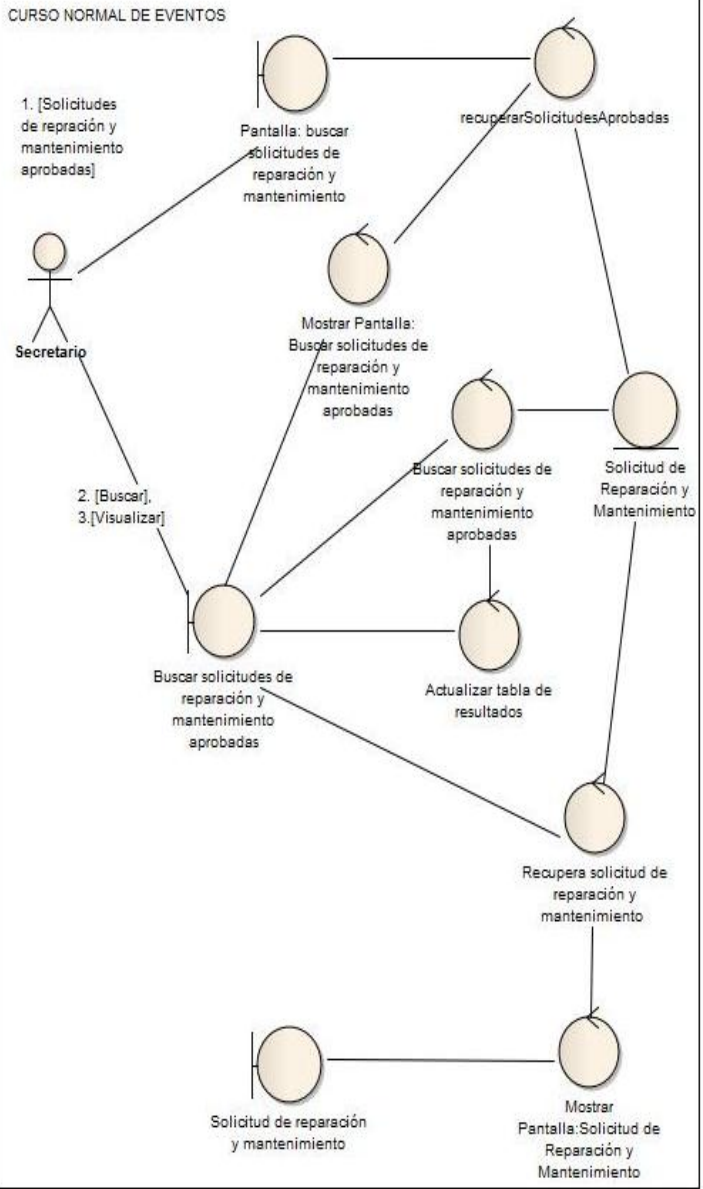


Figura 72. Diagrama de robustez CU005 curso alterno eventos.

- CURSO NORMAL DE EVENTOS**
1. El Secretario presiona el botón [Solicitudes de reparación y mantenimiento Aprobadas] en la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.
 2. El sistema invoca la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.
 3. El Secretario ingresa los criterios de búsqueda para una solicitud de reparación y mantenimiento aprobada (registro del vehículo, placa, serie de motor, serie de chasis, fecha de solicitud, registro de la solicitud) y presiona el botón [Buscar] en la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.
 4. El sistema actualiza la tabla [Resultados de búsqueda], con las solicitudes de reparación aprobadas que coincidan con el criterio de búsqueda.
 5. El secretario selecciona la solicitud de reparación a visualizar de la tabla [Resultados de Búsqueda] y presiona el botón [Visualizar] de la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.
 6. El sistema presenta la solicitud de reparación y mantenimiento seleccionada en la pantalla **solicitud de reparación y mantenimiento**.
 7. El caso de uso finaliza.



- CURSO ALTERNO DE EVENTOS**
1. **Limpiar Búsqueda.**
 - A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.
 - A.2. El sistema limpia el nombre de la solicitud de reparación y mantenimiento a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar solicitudes de reparación y mantenimiento aprobadas**.

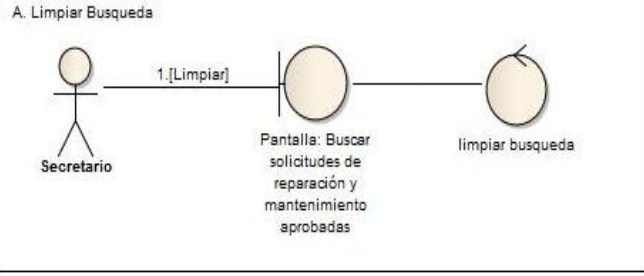


Figura 73. Diagrama de robustez CU006 curso normal y alterno eventos

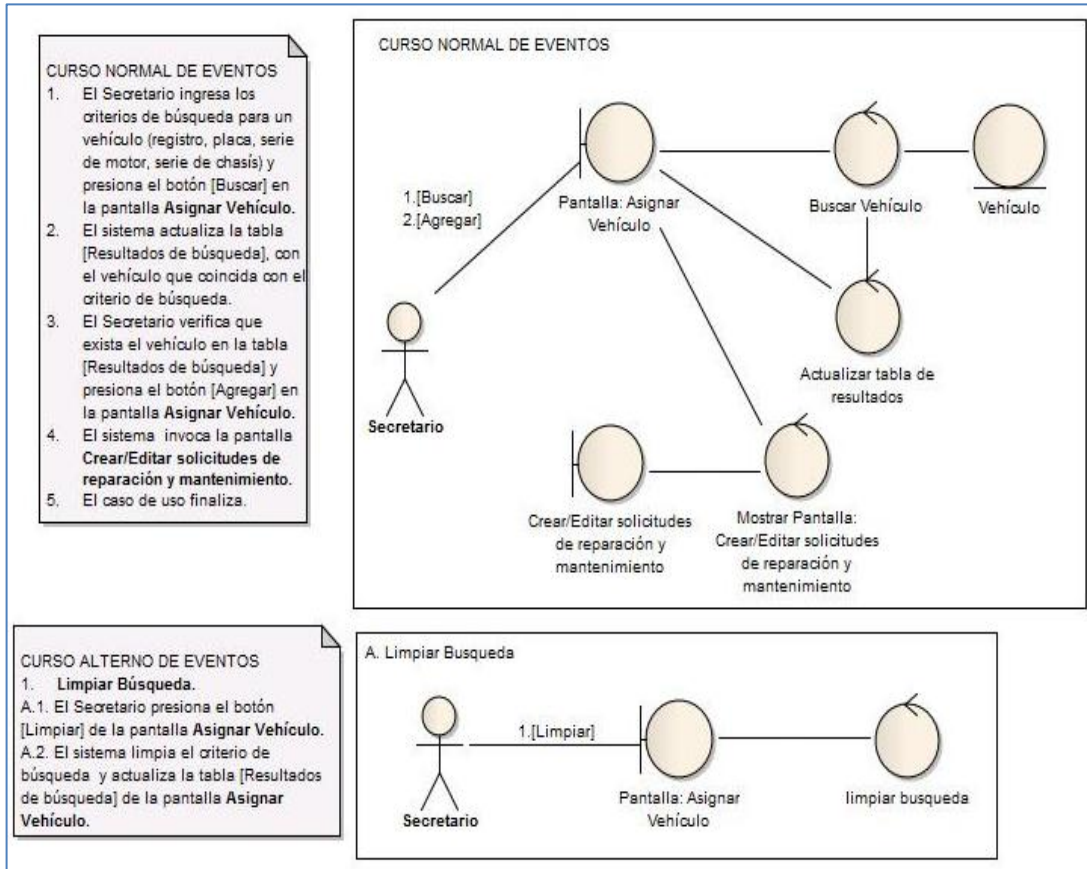


Figura 74. Diagrama de robustez CU007 curso normal y alterno eventos

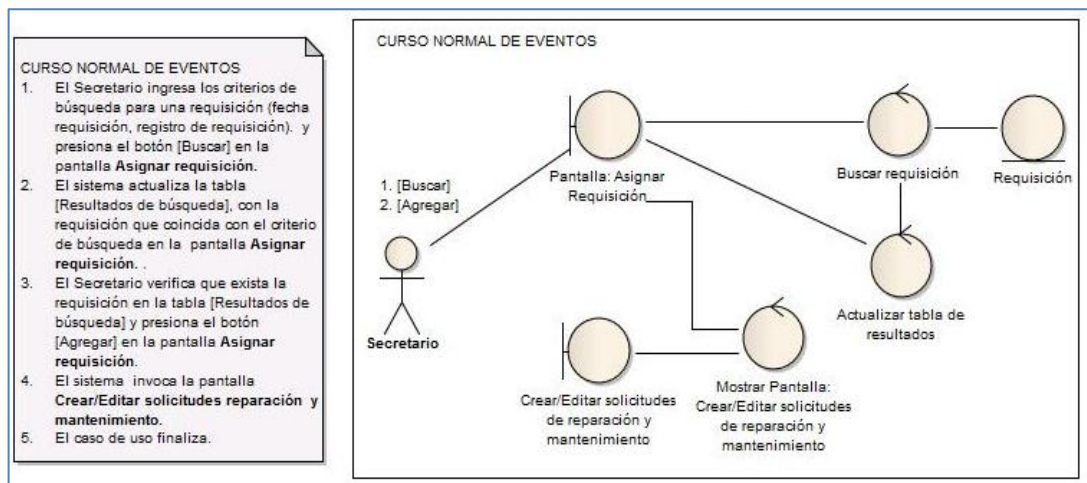


Figura 75. Diagrama de robustez CU008 curso normal eventos

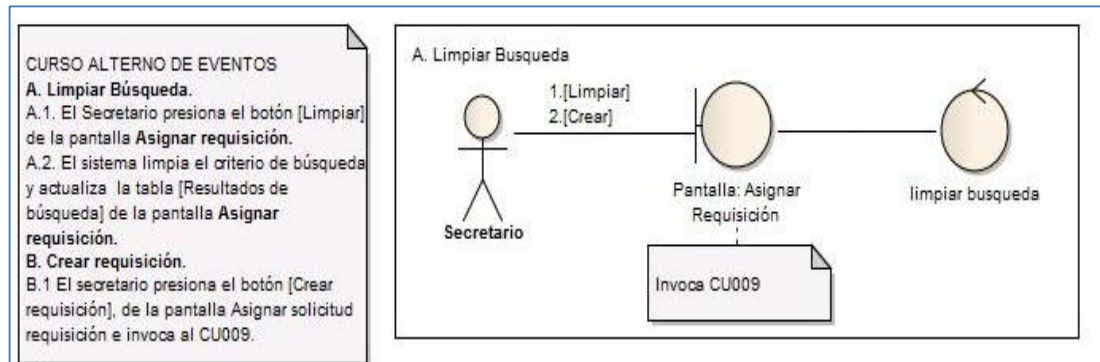


Figura 76. Diagrama de robustez CU008 curso alternativo eventos

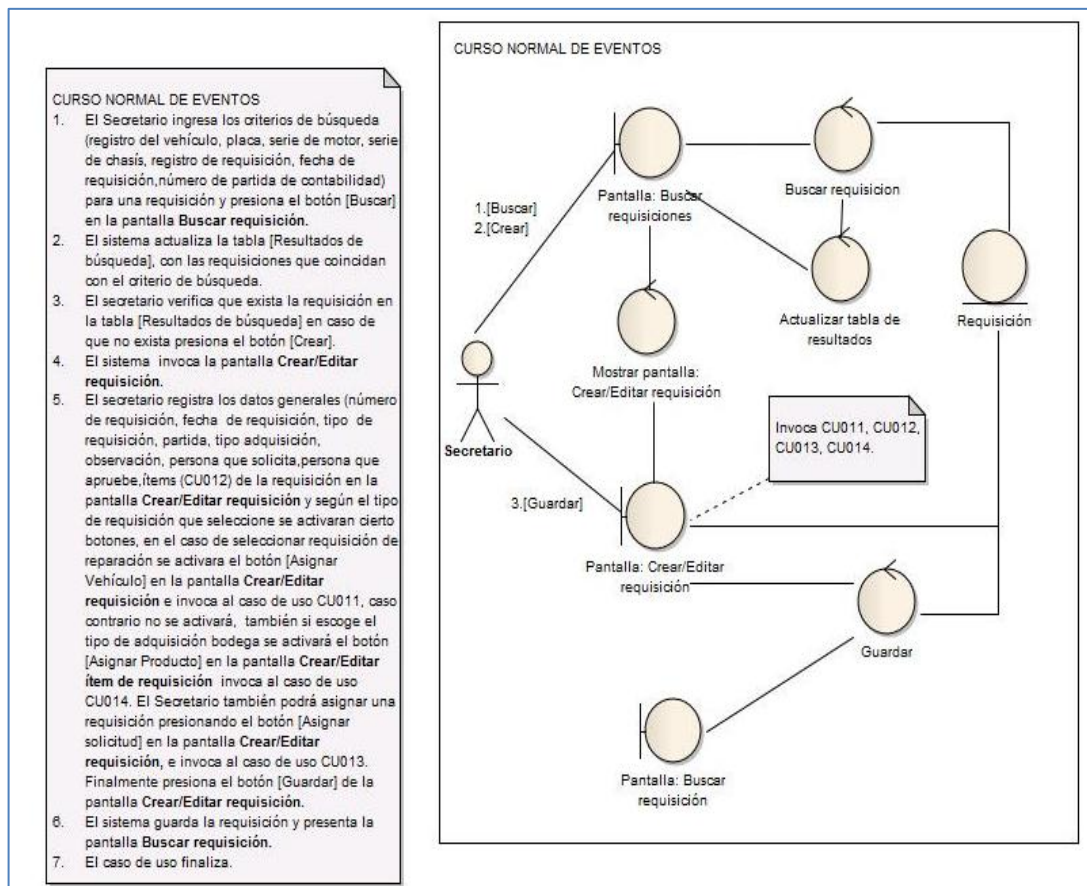


Figura 77. Diagrama de robustez CU009 curso normal eventos

CURSO ALTERNO DE EVENTOS

A. Limpiar Búsqueda.

A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar requisición**.

A.2. El sistema limpia el criterio de búsqueda y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar requisición**.

B. Editar Requisición.

B.3. El Secretario selecciona la requisición a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar requisición**.

B.4. El sistema recupera la información de la requisición y los muestra en la pantalla **Crear/Editar requisición**.

B.5. El Secretario edita los datos correspondientes de la requisición generales (número de requisición, fecha de requisición, tipo de requisición, partida, tipo adquisición, observación, persona que solicita, persona que aprueba, ítems (CU012) de la requisición en la pantalla **Crear/Editar requisición**, si elige el tipo de requisición de reparación se activará el botón [Asignar Vehículo] en la pantalla **Crear/Editar requisición** e invoca al caso de uso CU011, también si escoge el tipo de adquisición bodega se activará el botón [Asignar Producto] en la pantalla **Crear/Editar ítem de requisición** invoca al caso de uso CU014, caso contrario no se activará. El Secretario también podrá asignar una requisición presionando el botón [Asignar solicitud] en la pantalla **Crear/Editar requisición**, e invoca al caso de uso CU013) y presiona el botón [Guardar] de la pantalla **Crear/Editar requisición**.

B.6. El sistema guarda la nueva requisición con su respectiva información y presenta la pantalla **Buscar requisición** con el mensaje "Requisición editada".

C. Dar de baja requisición.

C.3. El secretario selecciona la requisición a dar de baja de la tabla [Resultados de Búsqueda] y presiona el botón [dar de baja] de la pantalla **Buscar requisición**.

C.4. El sistema invoca la pantalla **Dar de baja Requisición**.

C.5. El secretario presiona el botón [Sí], de la pantalla **Dar de baja Requisición**.

C.6. El sistema presenta un mensaje "Requisición seleccionada ha sido dado de baja".

D. Visualizar requisiciones aprobadas.

D.1. El secretario presiona el botón [Requisiciones Aprobadas] de la pantalla **Buscar requisición** e invoca al caso de uso CU010.

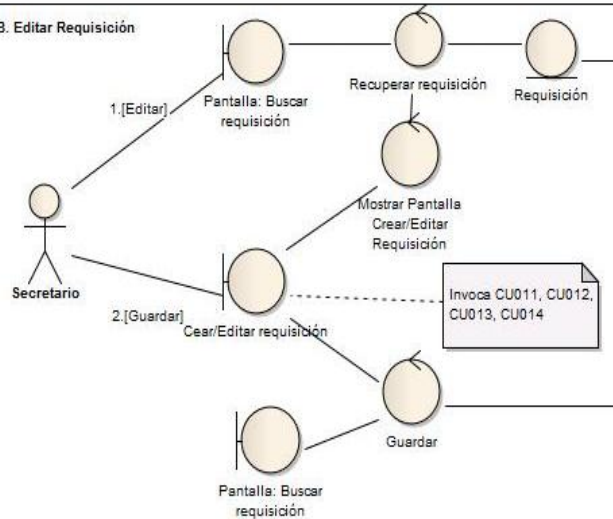
E. Validar Campos Obligatorios.

E.6 (E.B.6). El sistema verifica que estén llenos los campos obligatorios (número de requisición, fecha de requisición, tipo de requisición, partida, tipo adquisición, personal que solicita, el vehículo y al menos un ítem) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Requisición**, y continúa (5, B5).

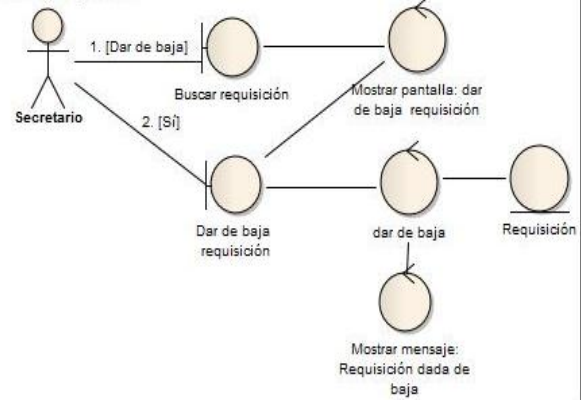
A. Limpiar Búsqueda



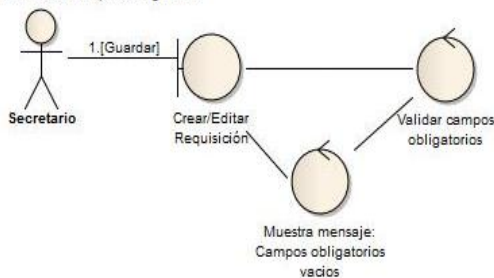
B. Editar Requisición



C. Dar de baja requisición



E. Validar Campos Obligatorios



D. Visualizar requisiciones aprobadas

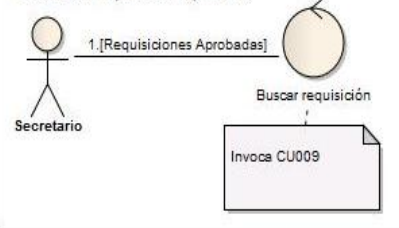
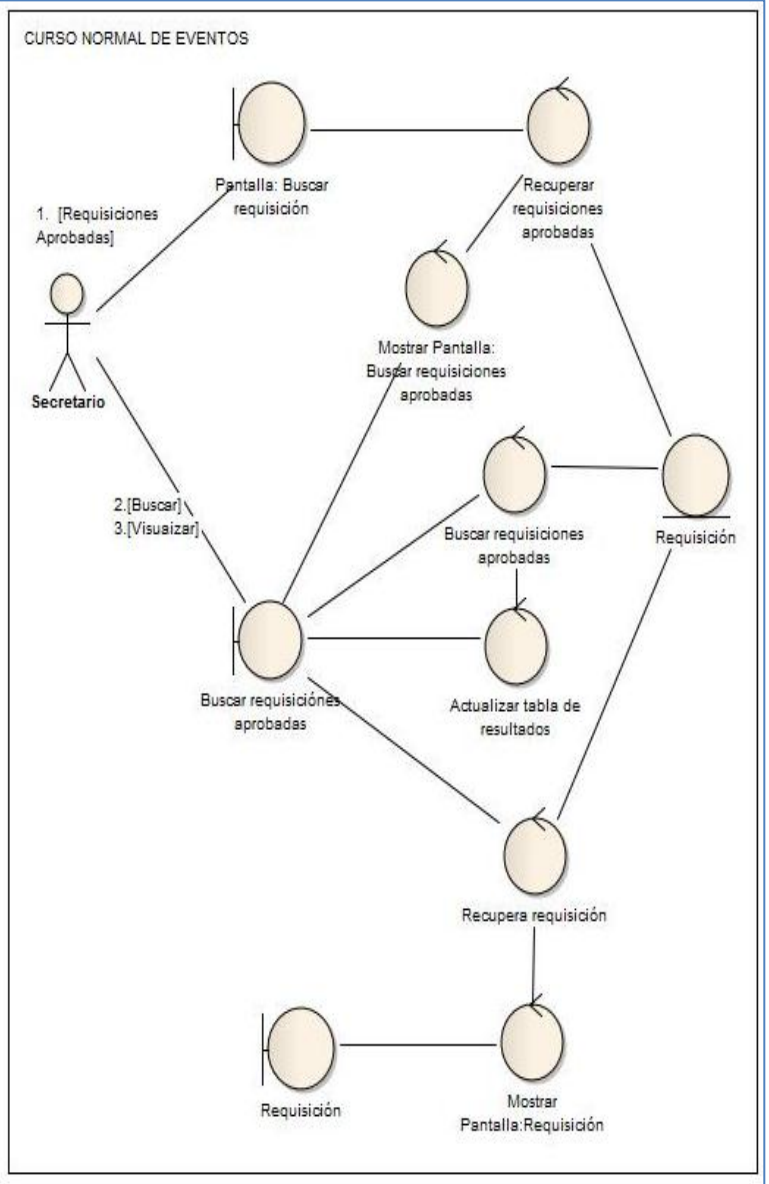


Figura 78. Diagrama de robustez CU009 curso alterno eventos

- CURSO NORMAL DE EVENTOS**
1. El Secretario presiona el botón [Requisiciones Aprobadas] en la pantalla **Buscar requisición**.
 2. El sistema presenta la pantalla **Buscar requisiciones aprobadas**.
 3. El Secretario ingresa los criterios de búsqueda (placa, serie chasis, serie motor, registro del vehículo, registro de requisición, fecha de requisición) para una requisición aprobada y presiona el botón [Buscar] en la pantalla **Buscar Requisiciones Aprobadas**.
 4. El sistema actualiza la tabla [Resultados de búsqueda], con las requisiciones aprobadas que coincidan con el criterio de búsqueda.
 5. El secretario selecciona la requisición a visualizar de la tabla [Resultados de Búsqueda] y presiona el botón [Visualizar] de la pantalla **Buscar requisiciones aprobadas**.
 6. El sistema presenta la Requisición seleccionada en la pantalla **Requisición**.
 7. El caso de uso finaliza.



- CURSO ALTERNO DE EVENTOS**
1. **Limpiar Búsqueda.**
 - A.1. El Secretario presiona el botón [Limpiar] de la pantalla **Buscar requisiciones aprobadas**.
 - A.2. El sistema limpia el nombre de la requisición a buscar y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar requisiciones aprobadas**.

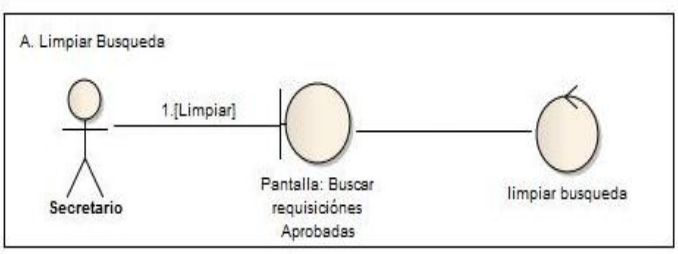


Figura 79. Diagrama de robustez CU010 curso normal y alterno eventos

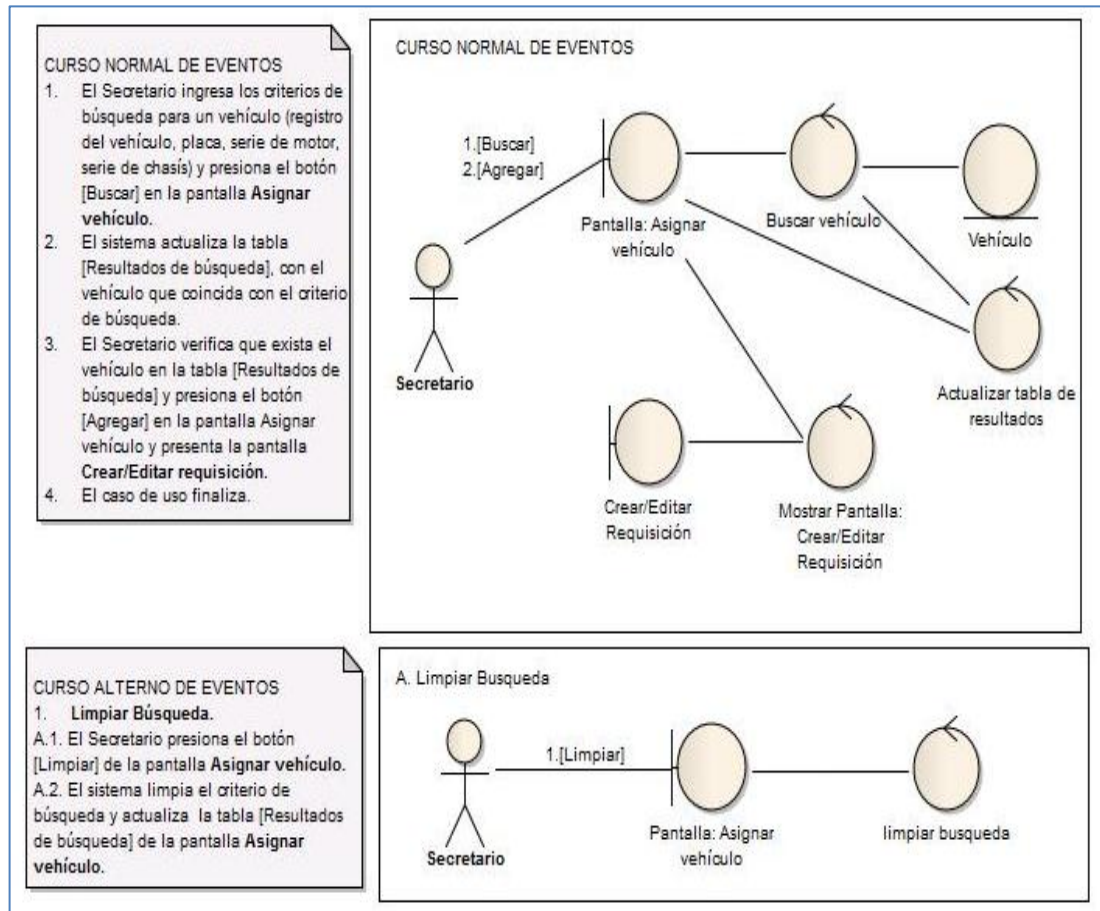


Figura 80. Diagrama de robustez CU011 curso normal y alterno eventos

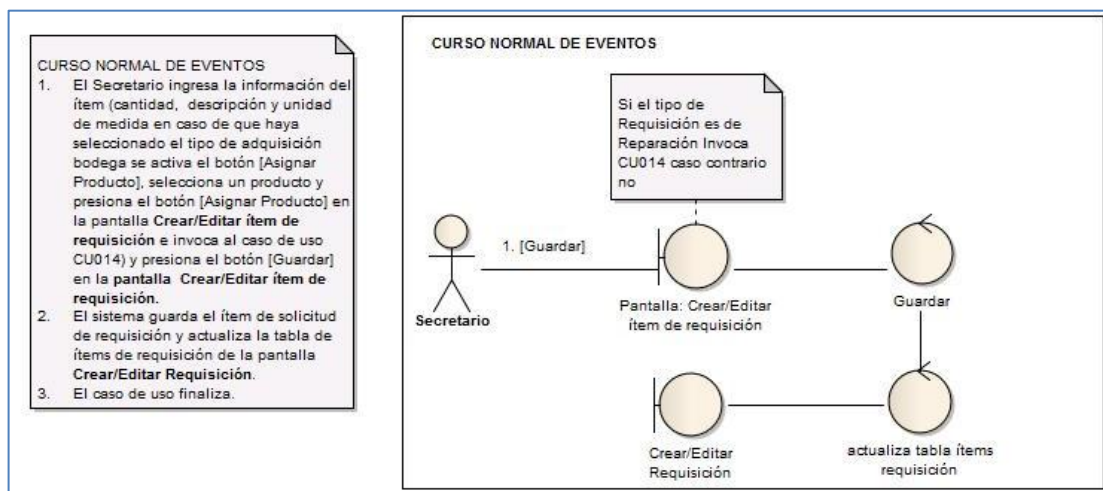


Figura 81. Diagrama de robustez CU012 curso normal eventos

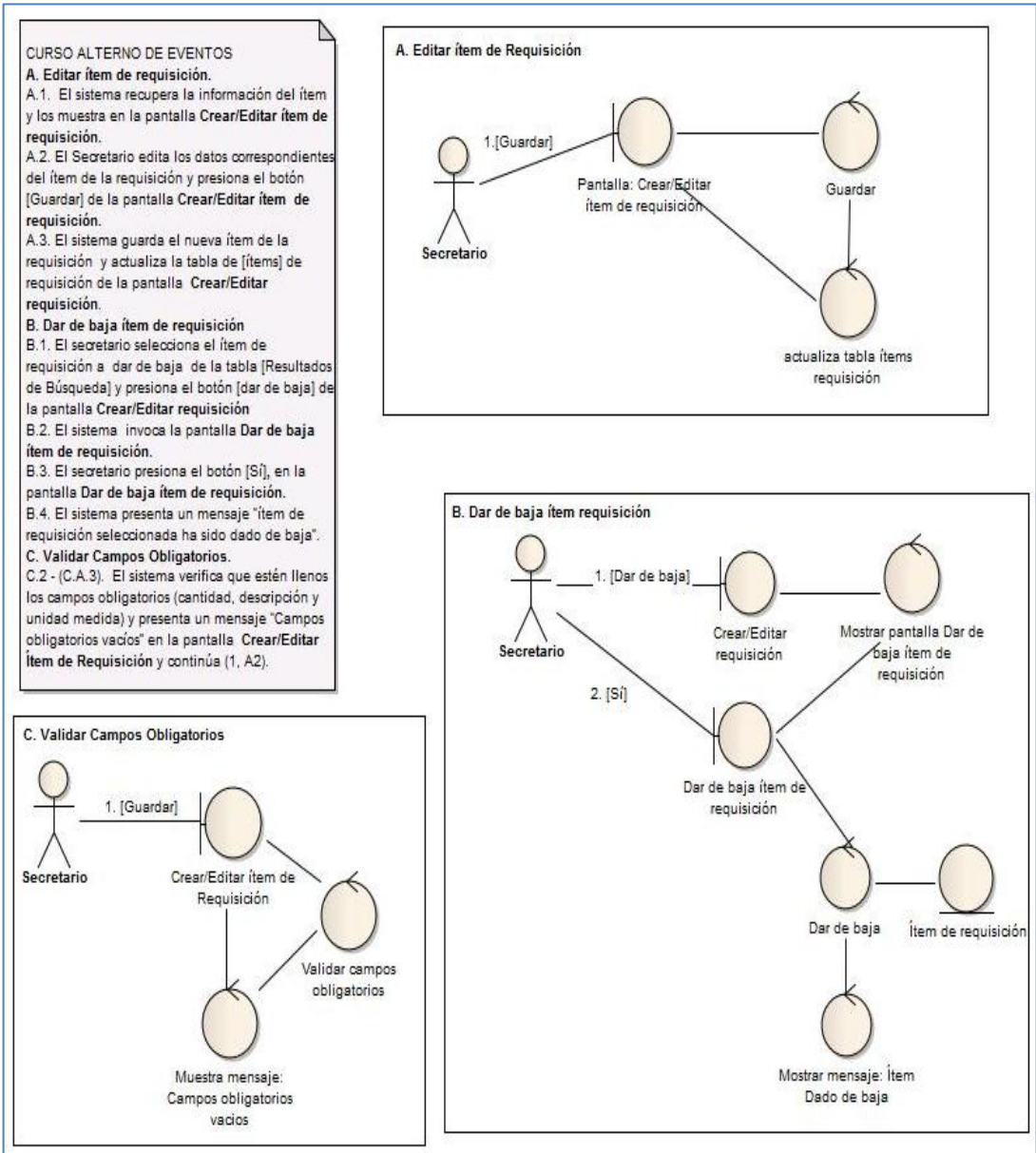


Figura 82. Diagrama de robustez CU012 curso alternativo eventos

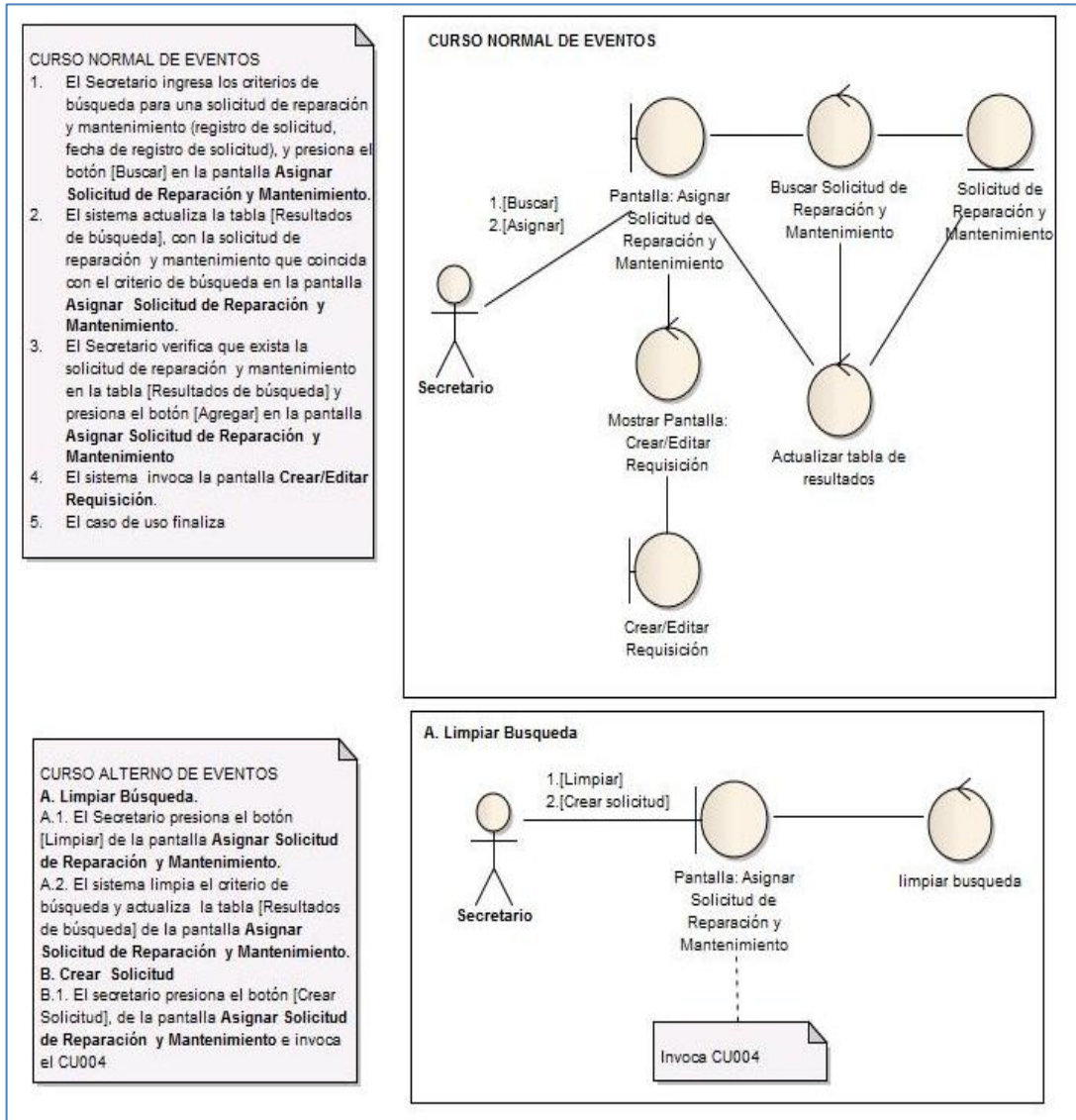


Figura 83. Diagrama de robustez CU013 curso normal y alterno eventos

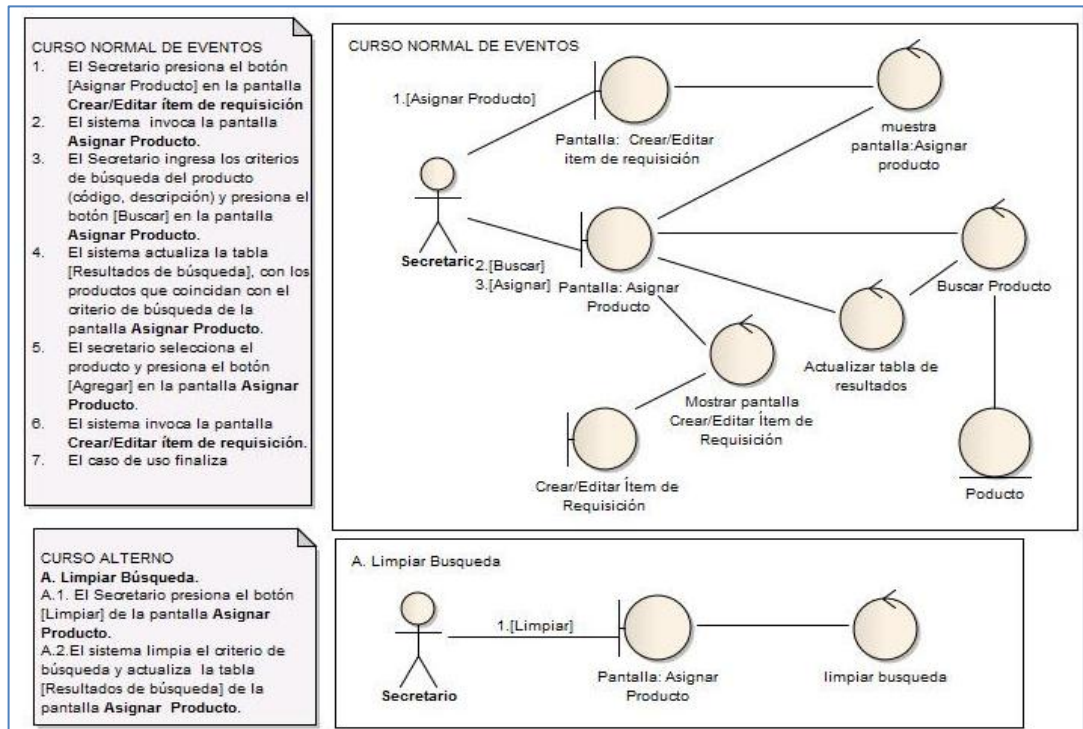


Figura 84. Diagrama de robustez CU014 curso normal y alterno eventos.

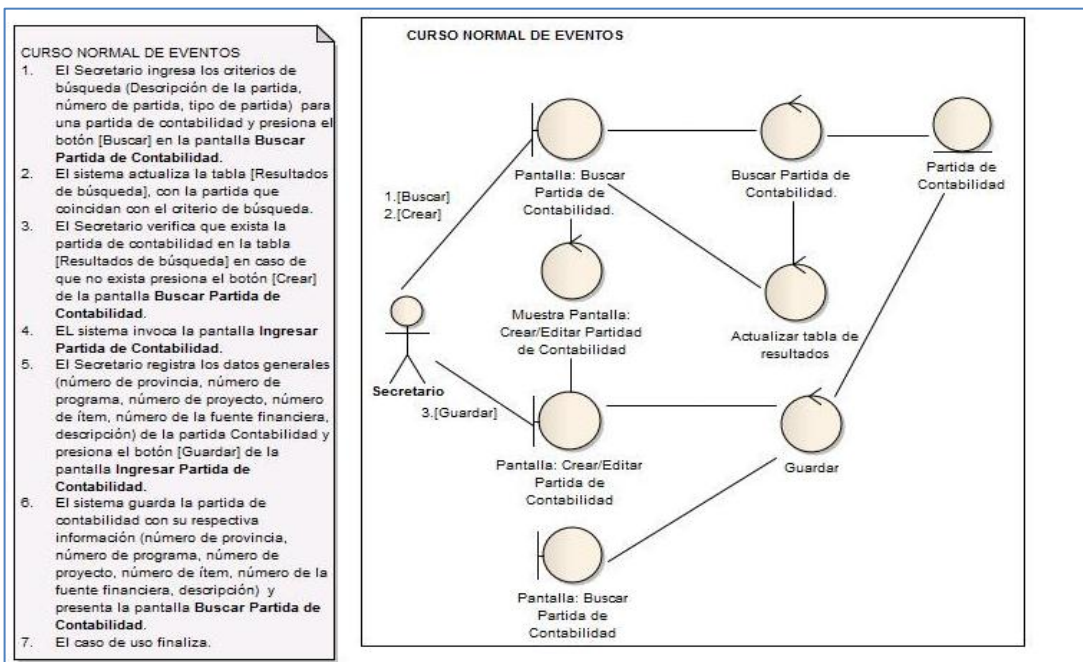


Figura 85. Diagrama de robustez CU015 curso normal eventos.

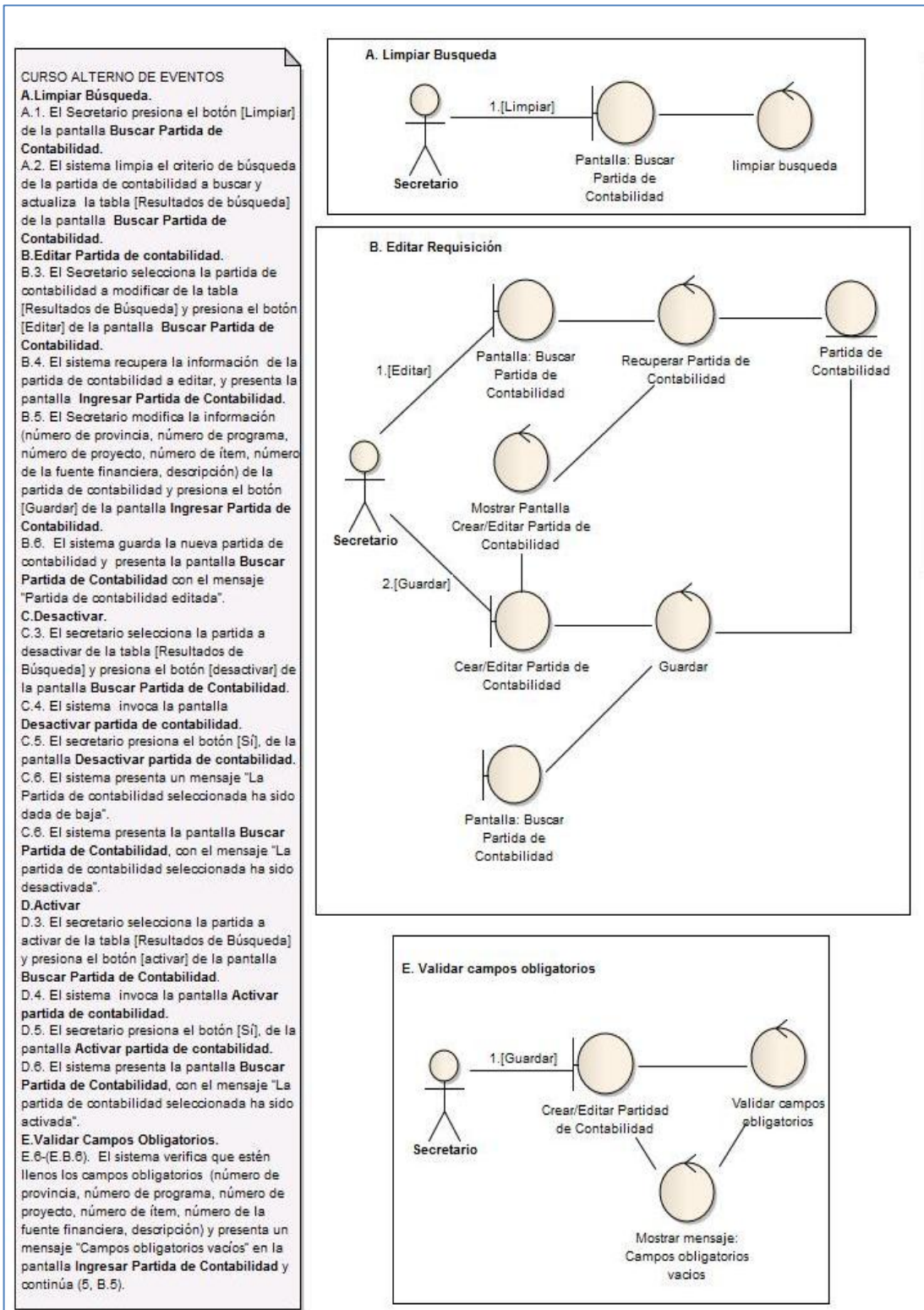


Figura 86. Diagrama de robustez CU015 curso alterno eventos (a).

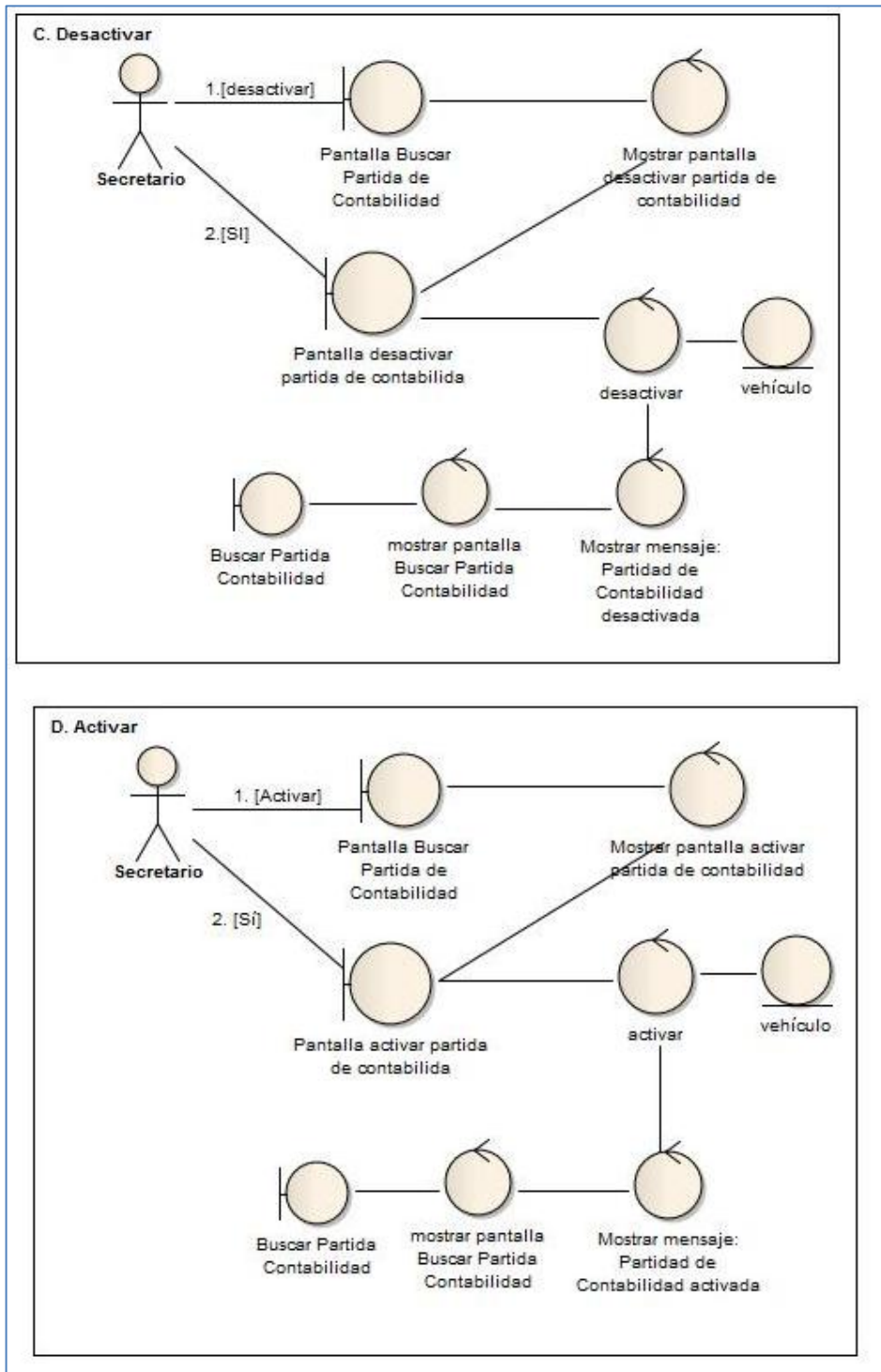


Figura 87. Diagrama de robustez CU015 curso alternativo eventos (b).

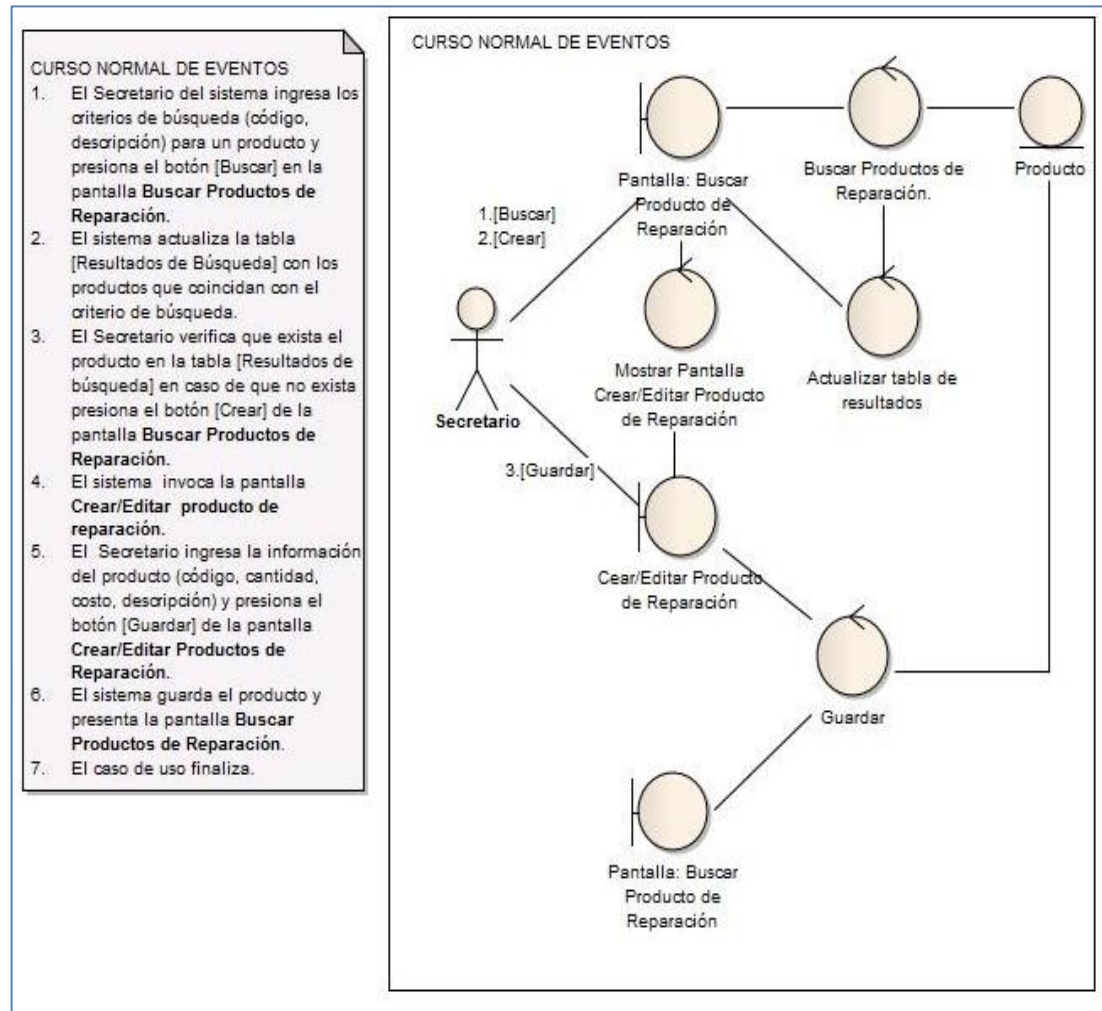


Figura 88. Diagrama de robustez CU016 curso normal eventos.

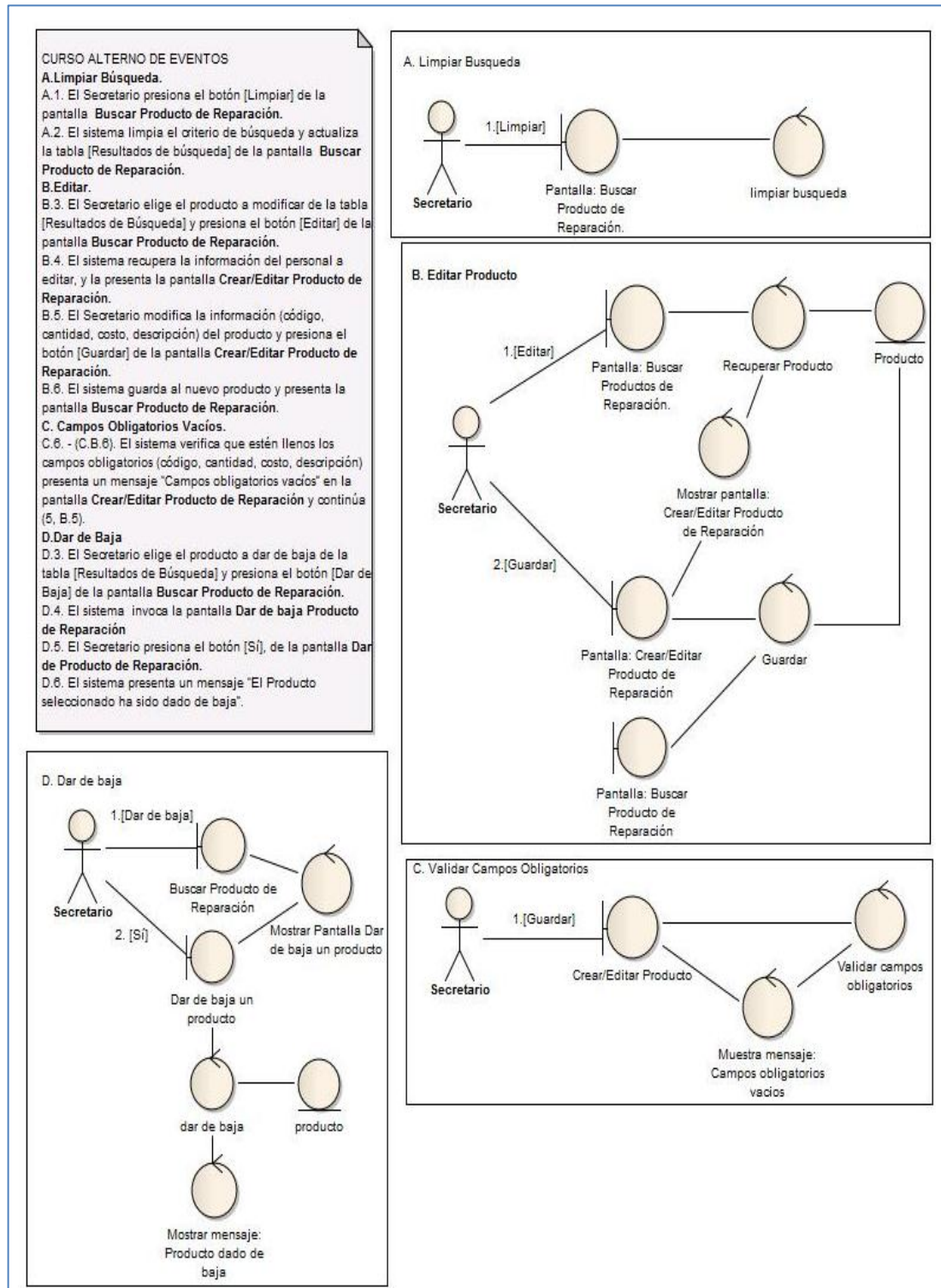


Figura 89. Diagrama de robustez CU016 curso alterno eventos.

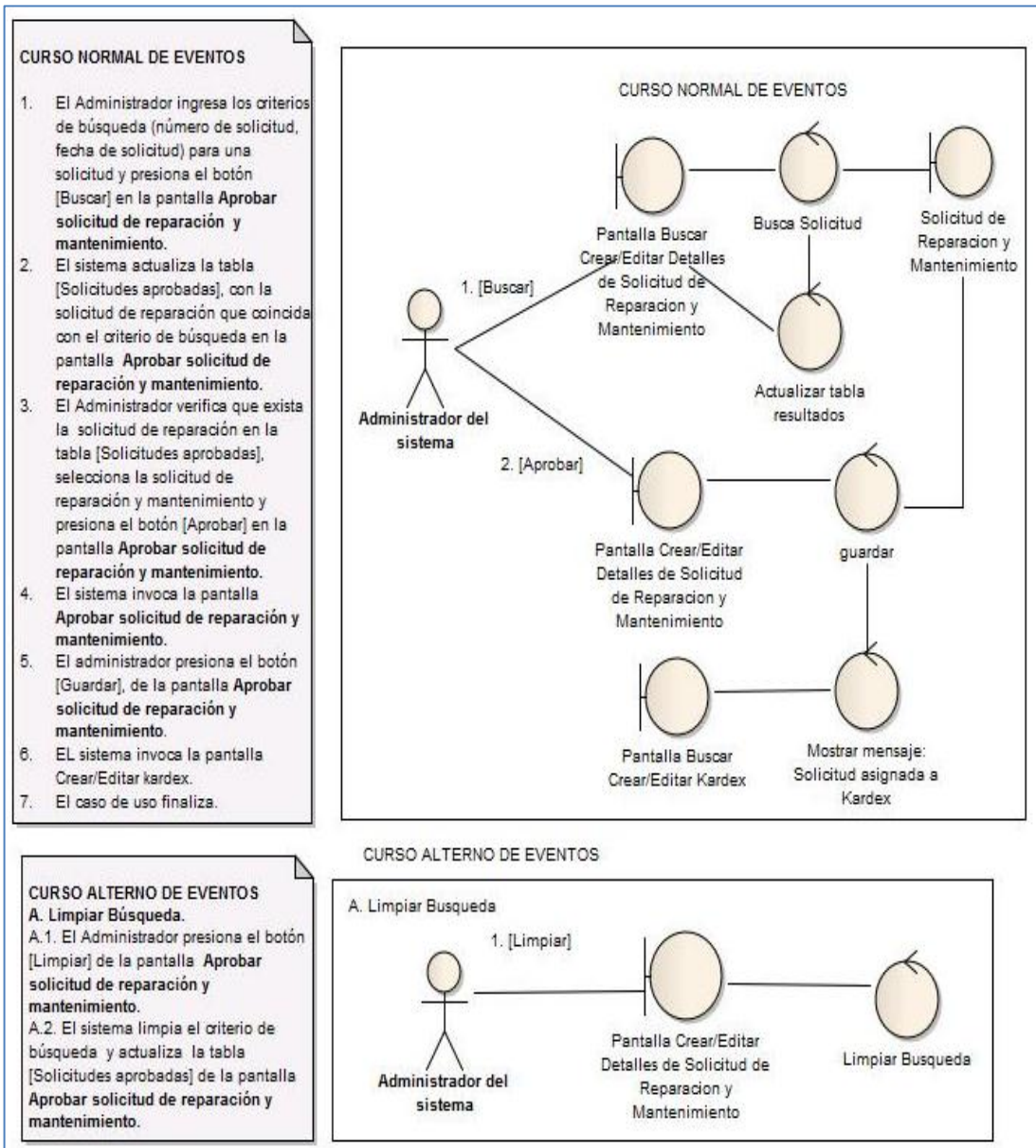


Figura 90. Diagrama de robustez CU018 curso normal y alterno eventos.

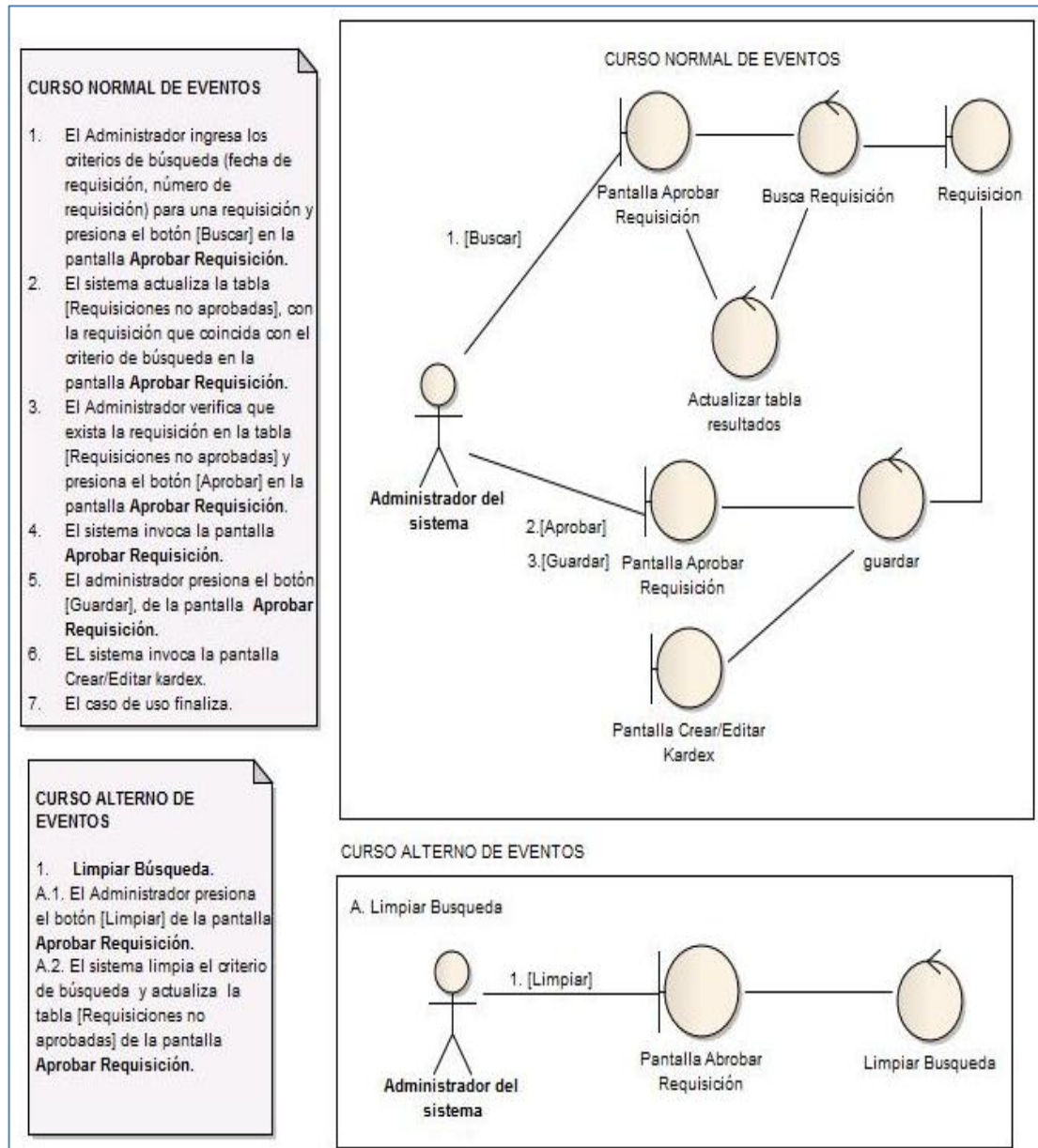


Figura 91. Diagrama de robustez CU019 curso normal y alterno eventos.

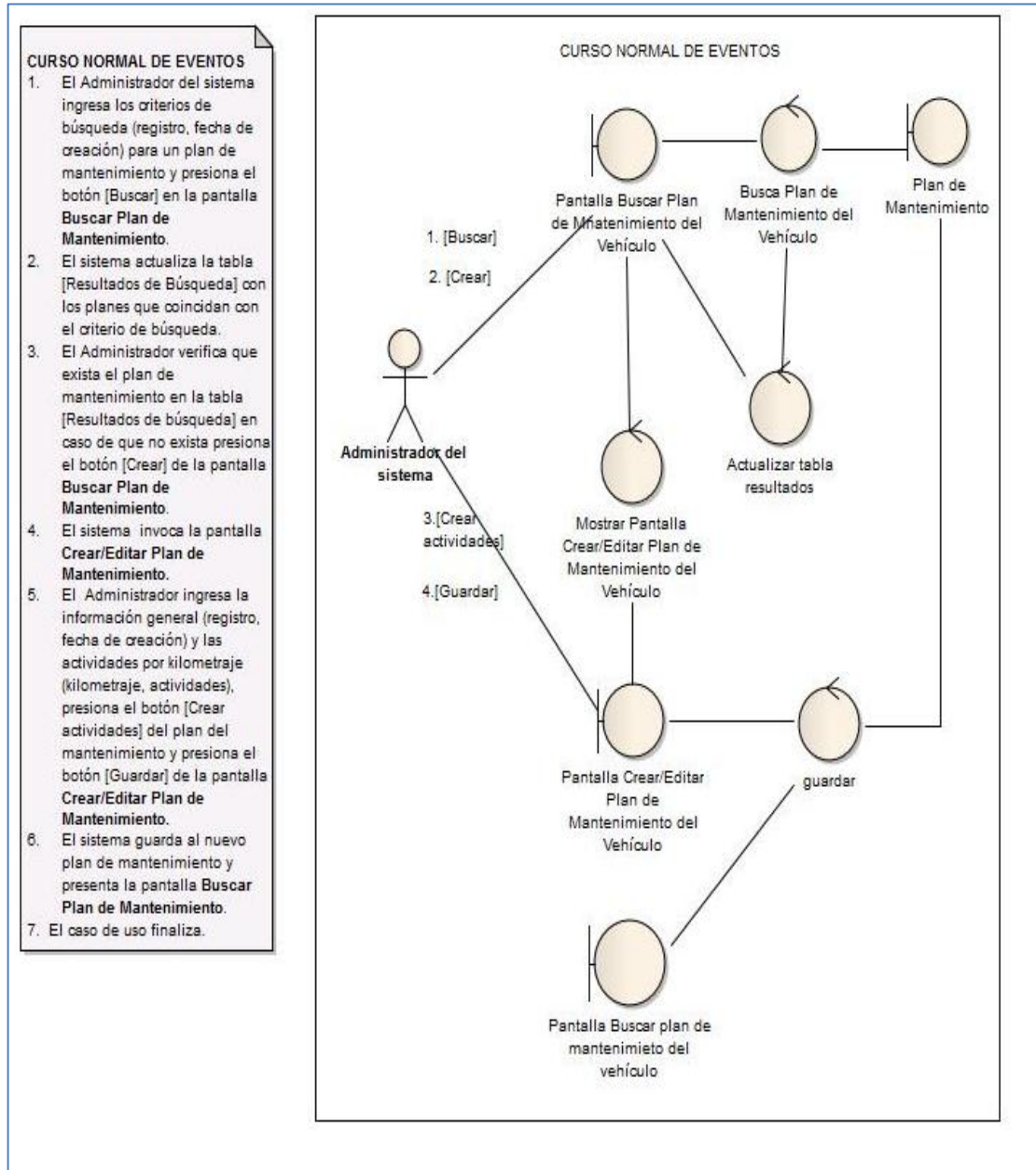


Figura 92. Diagrama de robustez CU020 curso normal eventos.

CURSO ALTERNO DE EVENTOS

A. Limpiar Búsqueda.

A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Plan de Mantenimiento**.

A.2. El sistema limpia el criterio de búsqueda y actualiza y actualiza la tabla [Resultados de búsqueda] de la pantalla **Buscar Plan de Mantenimiento**.

B. Editar.

B.3. El administrador elige el plan de mantenimiento a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Plan de Mantenimiento**.

B.4. El sistema recupera la información del plan de mantenimiento a editar, y la presenta en la pantalla **Crear/Editar Plan de Mantenimiento**.

B.5. El Administrador modifica la información general (registro, fecha de creación) y las actividades por kilometraje (kilometraje, actividades), presiona el botón [Crear actividades] del plan del mantenimiento y presiona el botón [Guardar] de la pantalla **Crear/Editar Plan de Mantenimiento**.

B.6. El sistema guarda al nuevo plan de mantenimiento presenta la pantalla **Buscar Plan de Mantenimiento**.

C. Campos Obligatorios Vacíos Plan de Mantenimiento.

C.6 - (C.6, C.B.6). El sistema verifica que estén llenos los campos obligatorios (registro, fecha de creación, kilometraje, actividades) y presenta un mensaje "Campos obligatorios vacíos" continúa (5, B.5).

D. Activar Plan.

D.3. El administrador elige el plan de mantenimiento a activar de la tabla [Resultados de Búsqueda] y presiona el botón [Activar Plan] de la pantalla **Buscar Plan de Mantenimiento**.

D.4. El sistema invoca la pantalla **Activar Plan de Mantenimiento**.

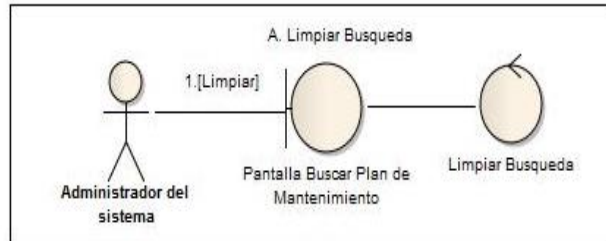
D.5. El administrador presiona el botón [Sí], de la pantalla **Activar Plan de Mantenimiento**.

D.6. El sistema presenta un mensaje "El Plan de mantenimiento seleccionado ha sido activado".

E. Visualizar plan activo.

E.3. El administrador presiona el botón [Visualizar plan activo] de la pantalla **Buscar Plan de Mantenimiento**.

E.4 El sistema muestra las próximas actividades del mantenimiento a realizar del vehículo seleccionado.



B. Editar Plan de Mantenimiento

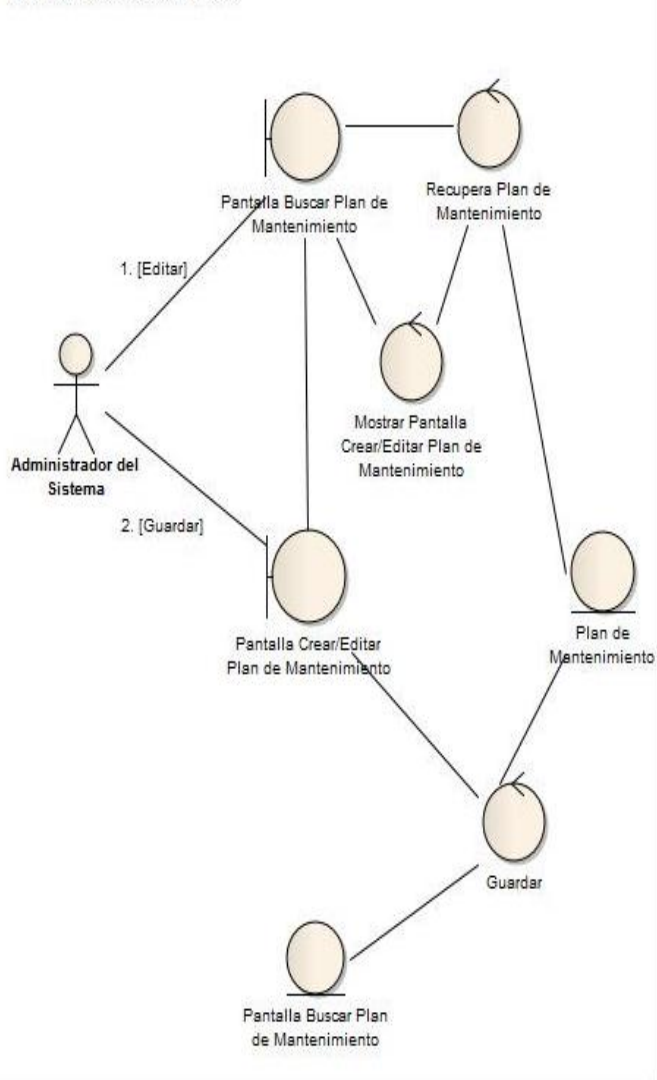


Figura 93. Diagrama de robustez CU020 curso alternativo de eventos (a).

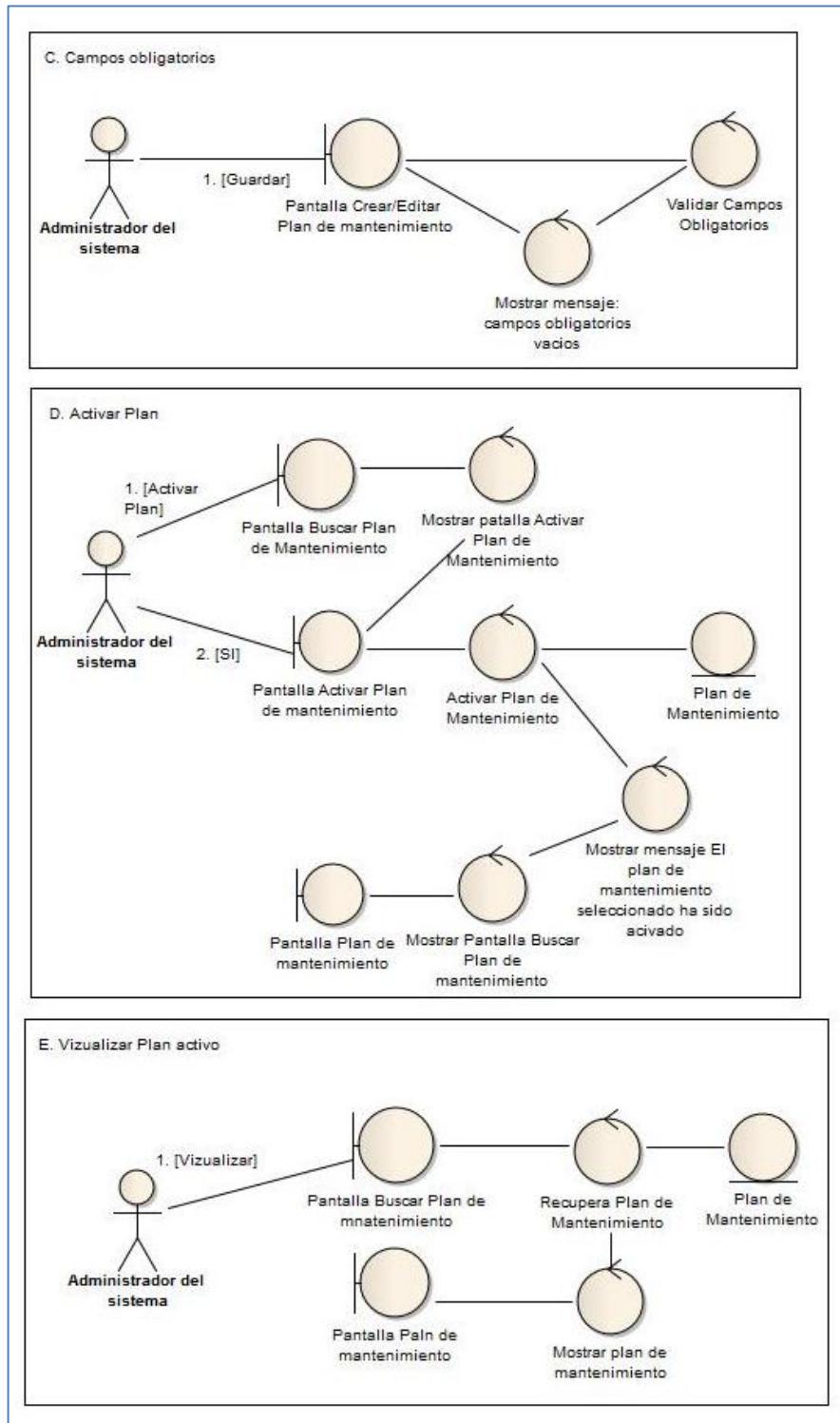


Figura 94. Diagrama de robustez CU020 curso alternativo de eventos (b).

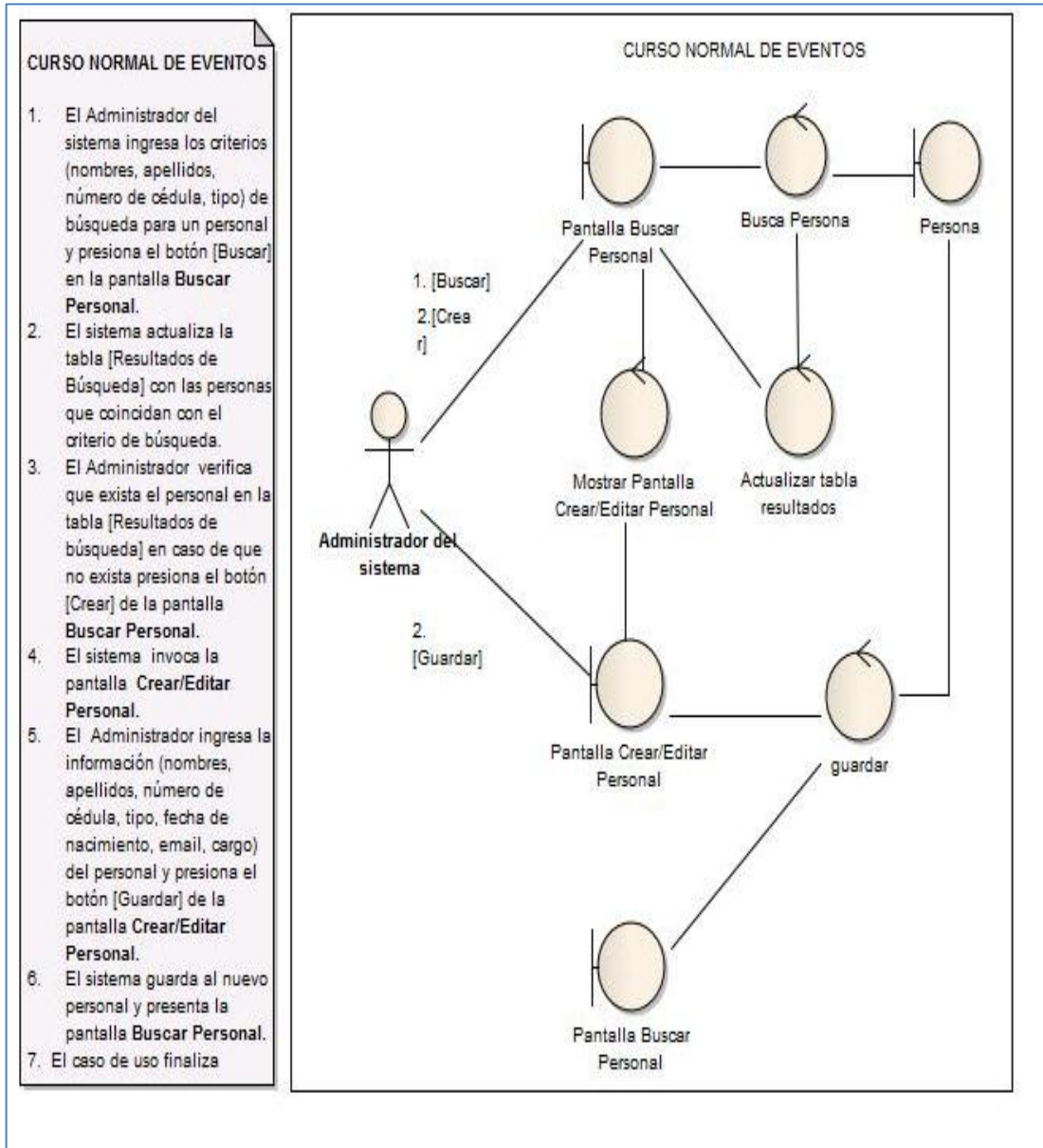


Figura 95. Diagrama de robustez CU022 curso normal de eventos.

CURSO ALTERNO DE EVENTOS

A. Limpiar Búsqueda.
 A.1. El Administrador presiona el botón [Limpiar] de la pantalla **Buscar Personal**.
 A.2. El sistema limpia el criterio de búsqueda y actualiza en la tabla [Resultados de búsqueda] de la pantalla **Buscar Personal**.

B. Editar personal.
 B.3. El administrador elige el personal a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Editar] de la pantalla **Buscar Personal**.
 B.4. El sistema recupera la información del personal a editar, y la presenta en la pantalla **Crear/Editar Personal**.
 B.5. El Administrador modifica la información (nombres, apellidos, número de cédula, tipo, fecha de nacimiento, email, cargo) del personal y presiona el botón [Guardar] de la pantalla **Crear/Editar Personal**.
 B.6. El sistema guarda el nuevo personal

C. Campos obligatorios vacíos del personal.
 C.6. - (C.B.6) El sistema verifica que estén llenos los campos obligatorios (nombres, apellidos, número de cédula, tipo, fecha de nacimiento, email) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Personal** y continúa (5, B.5).

D. Número de cédula duplicado del personal.
 D.5 - (D.B.5) El sistema verifica si el número de cédula del personal esta duplicado y presenta un mensaje "Número de cedula invalido o El DNI ingresado pertenece a otra persona" en la pantalla **Crear/Editar Personal** y continúa (5, B.5).

E. Crear Cuenta de Usuario.
 E.3. El administrador elige el personal de la tabla [Resultados de Búsqueda] y presiona el botón [Crear Cuenta de Usuario] de la pantalla **Buscar Personal**.
 E.4. El sistema invoca la pantalla **Crear/Editar Usuario**.
 E.5. El Administrador ingresa la información (usuario, contraseña, confirmar contraseña) para el usuario presiona el botón [Guardar] de la pantalla **Crear/Editar Usuario**.
 E.6. El sistema guarda al nuevo usuario y presenta la pantalla **Buscar Personal**.

F. Editar Usuario.
 F.3. El administrador elige el personal a modificar de la tabla [Resultados de Búsqueda] y presiona el botón [Crear Cuenta de Usuario] de la pantalla **Buscar Personal**.
 F.4. El sistema recupera la información del usuario a editar, y presenta la pantalla **Crear/Editar Cuenta de usuario**.
 F.5. El Administrador modifica la información (contraseña, confirmar contraseña) del personal y presiona el botón [Guardar] de la pantalla **Crear/Editar Cuenta de Usuario**.
 F.6. El sistema guarda al nuevo usuario y presenta la pantalla **Buscar Personal**.

G. Campos obligatorios vacíos de usuario.
 G.6. - (G.F.6) El sistema verifica que estén llenos los campos obligatorios (usuario, contraseña, confirmar contraseña) y presenta un mensaje "Campos obligatorios vacíos" en la pantalla **Crear/Editar Personal** y continúa (5, F.5).

H. Usuario duplicado.
 H.6 - (H.F.6) El sistema verifica si el usuario esta duplicado y presenta un mensaje "El nombre de usuario no está disponible. Pertenecce a otro usuario" en la pantalla **Crear/Editar Personal** y continúa (5, F.5).

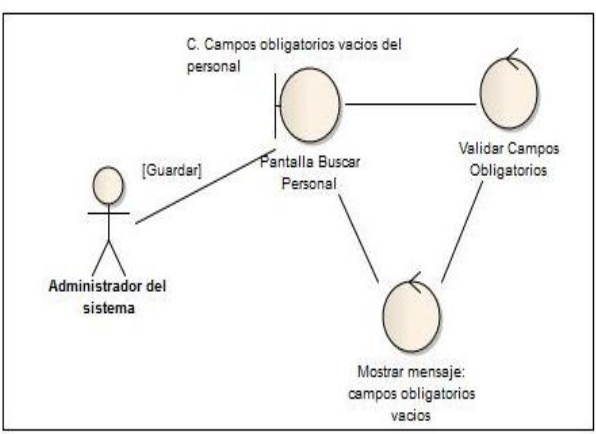
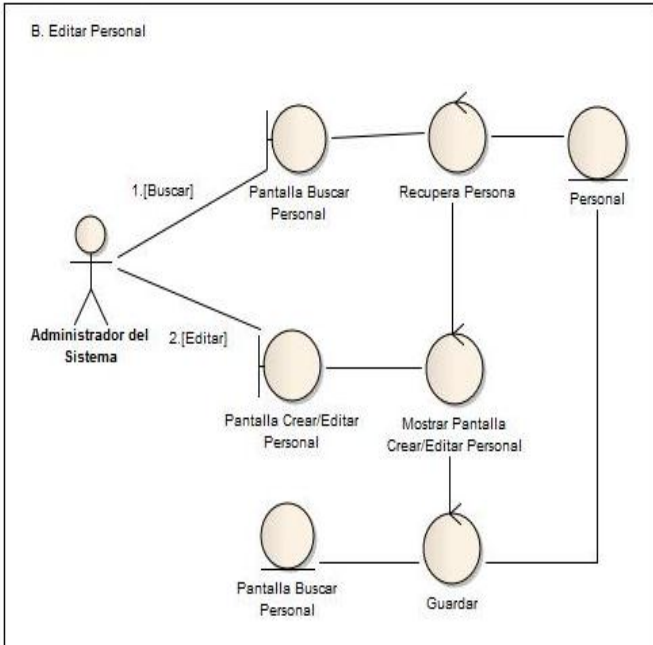
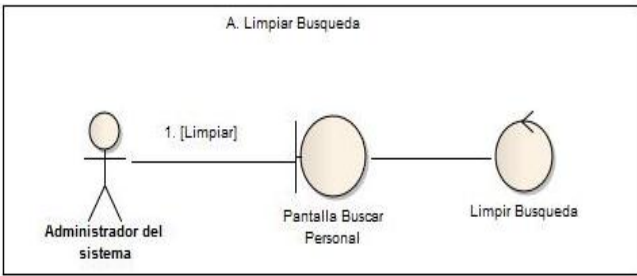


Figura 96. Diagrama de robustez CU022 curso alterno de eventos(a).

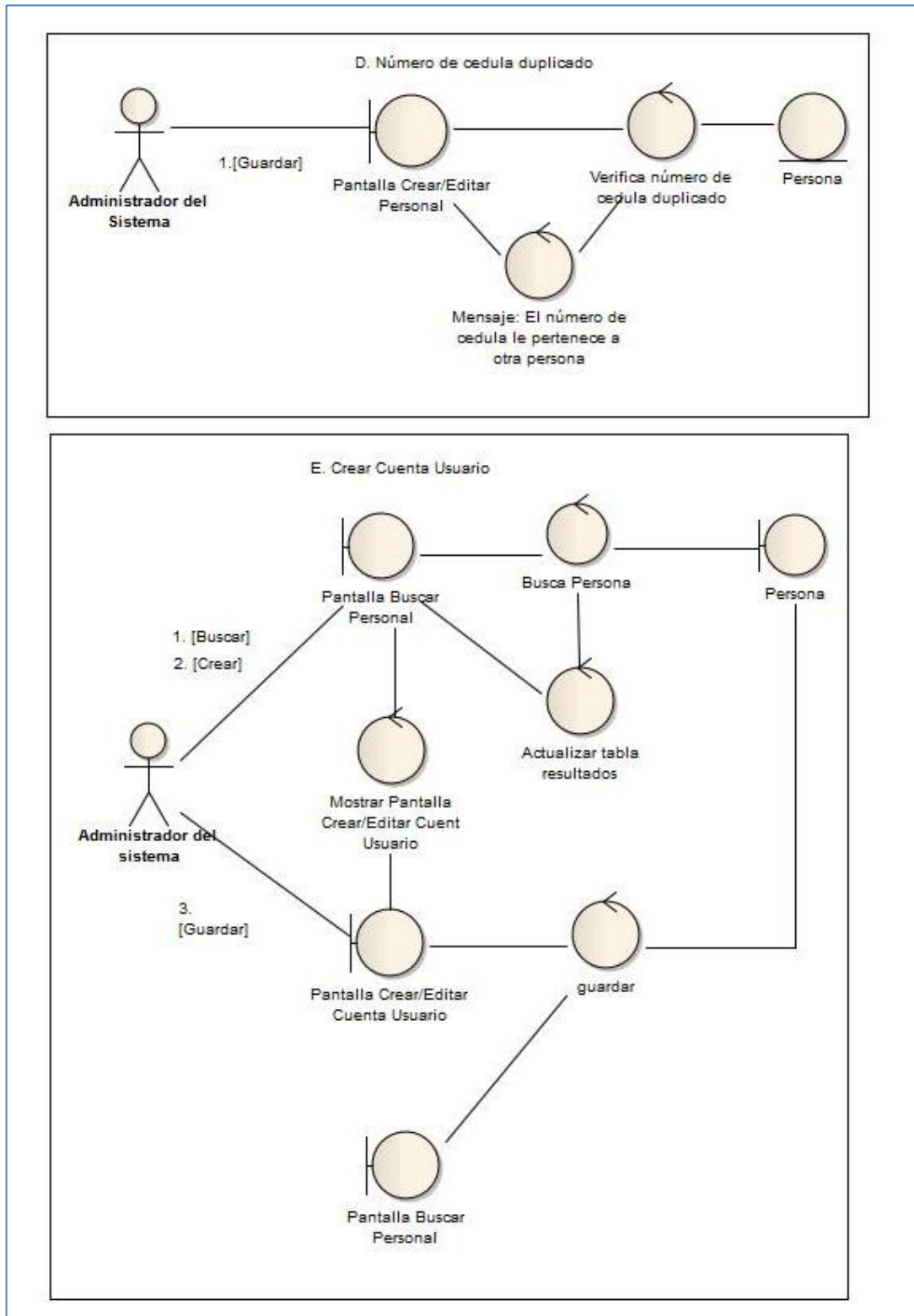


Figura 97. Diagrama de robustez CU022 curso alternativo de eventos(b).

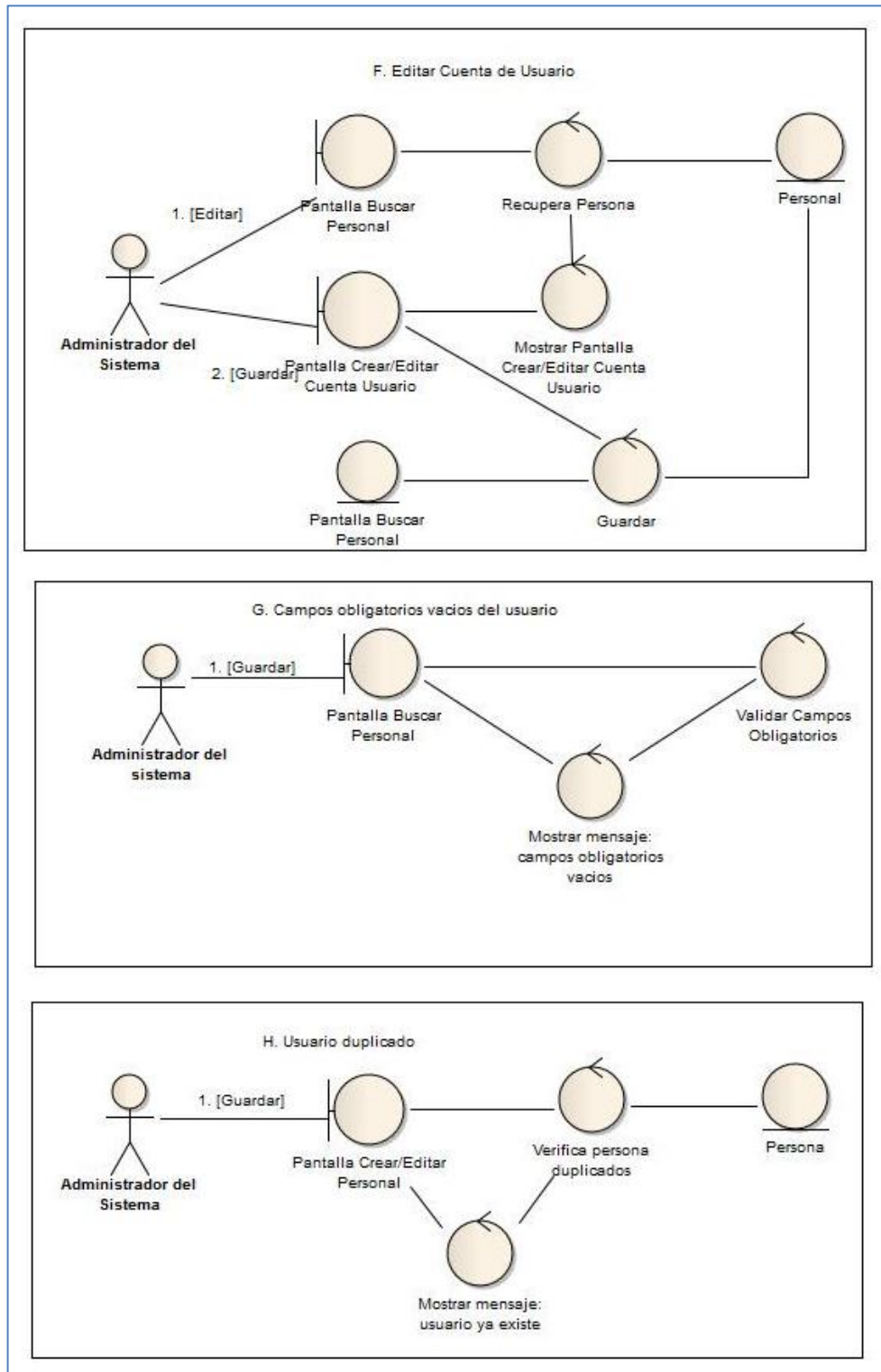


Figura 98. Diagrama de robustez CU022 curso alternativo de eventos(c).

Anexo 6: Diagrama de secuencias

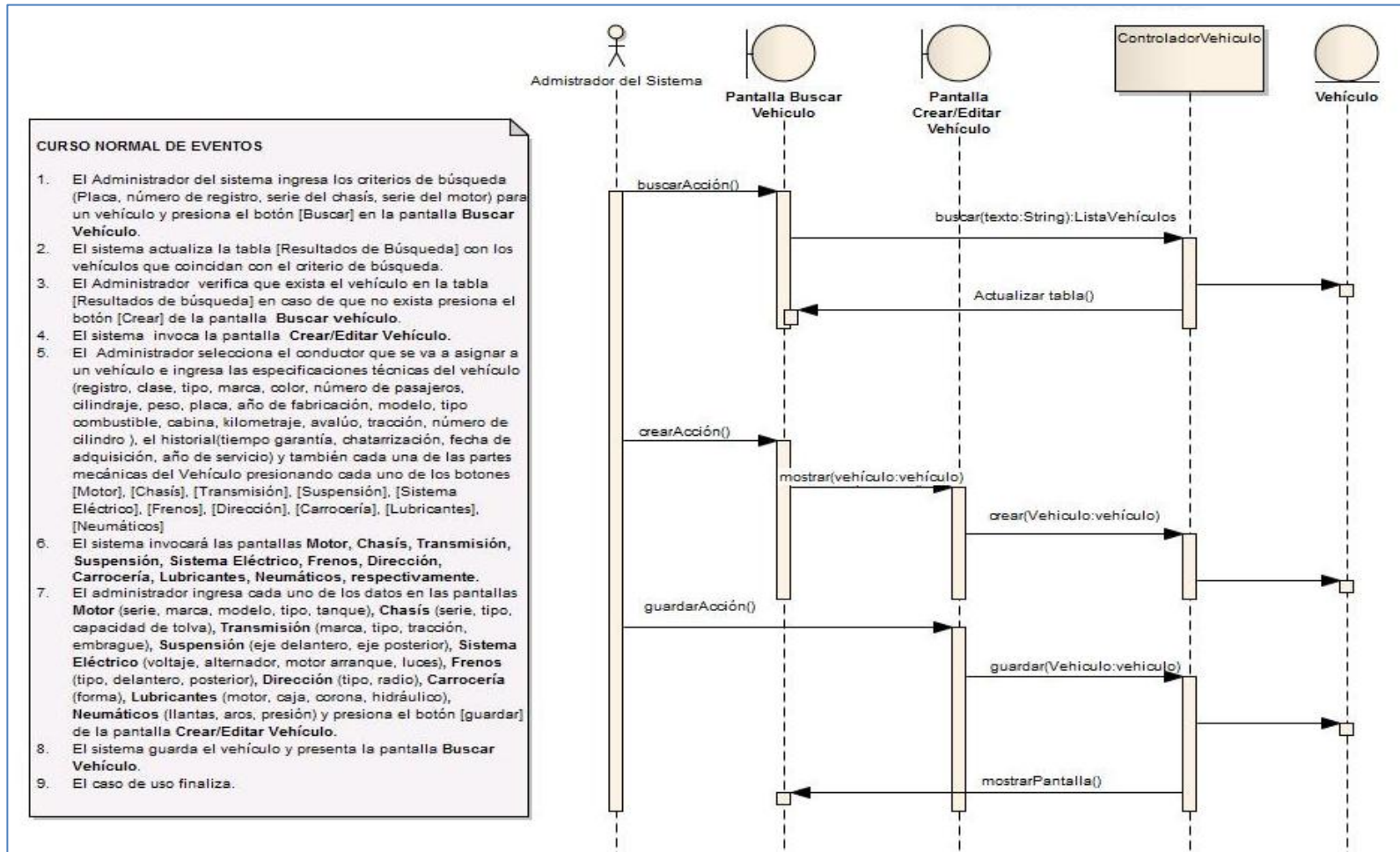


Figura 99. Diagrama de secuencia del CU001 curso normal de eventos.

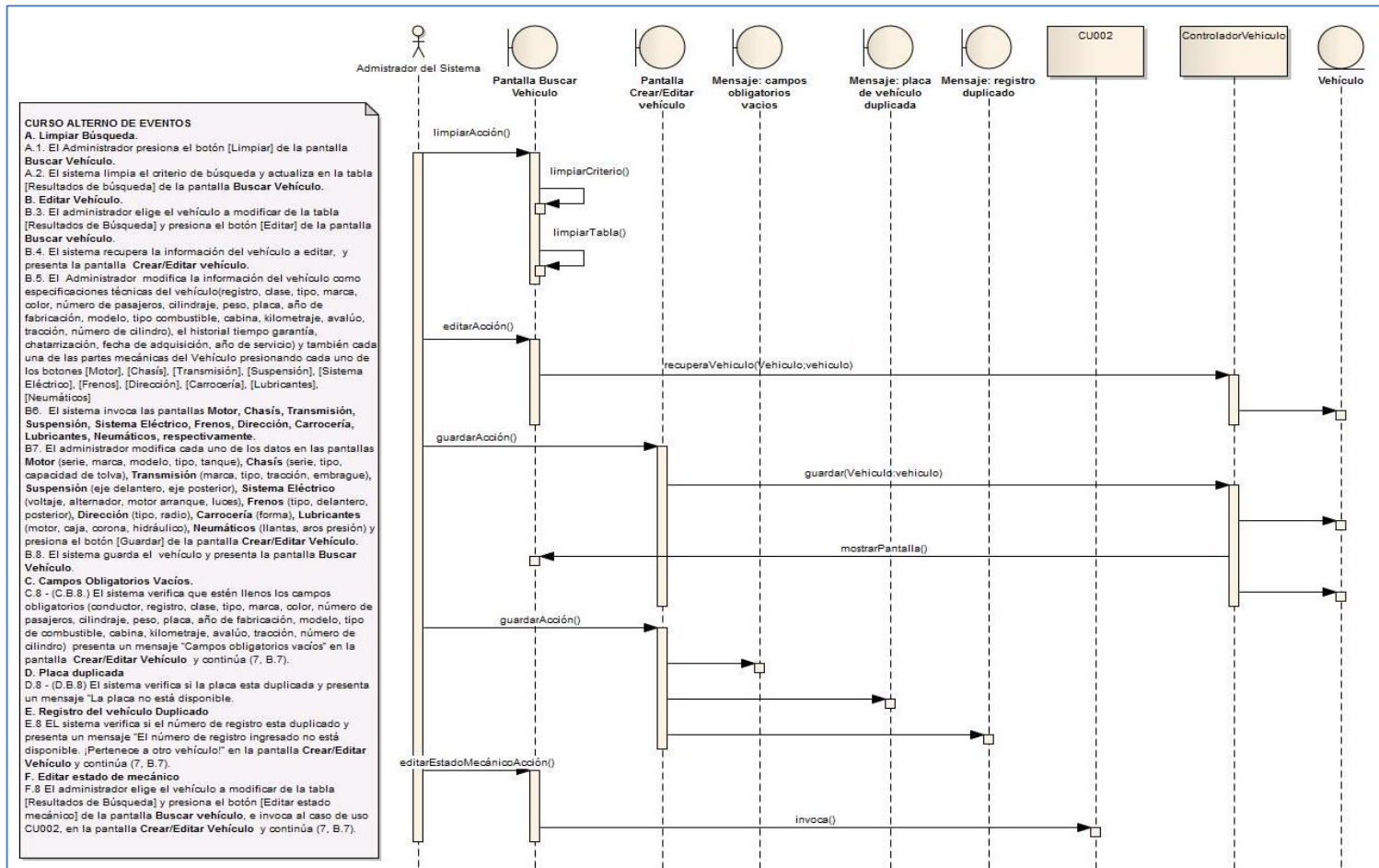


Figura 100. Diagrama de secuencia del CU001 curso alterno de eventos.

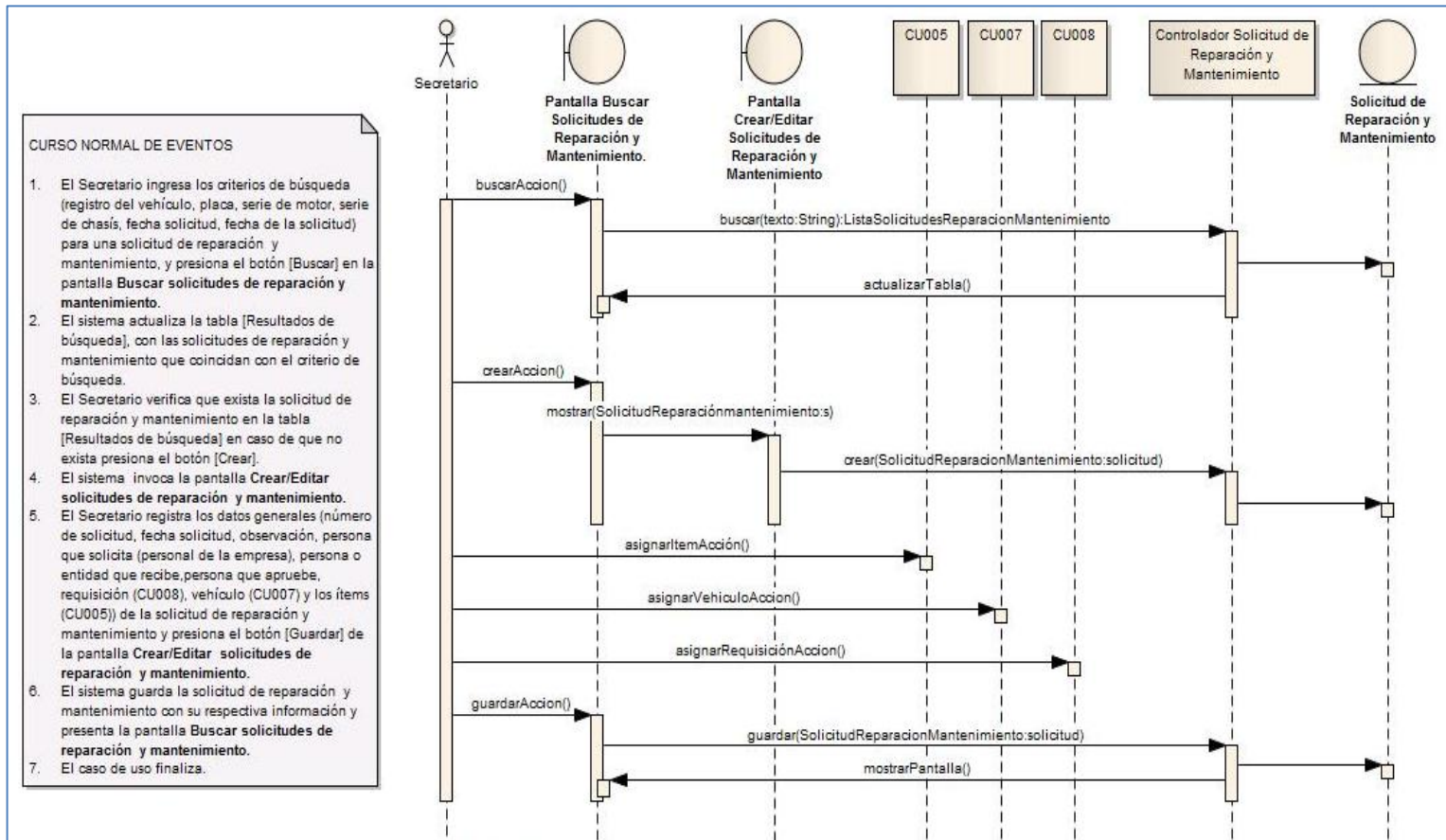


Figura 101. Diagrama de secuencia del CU004 curso normal de eventos.

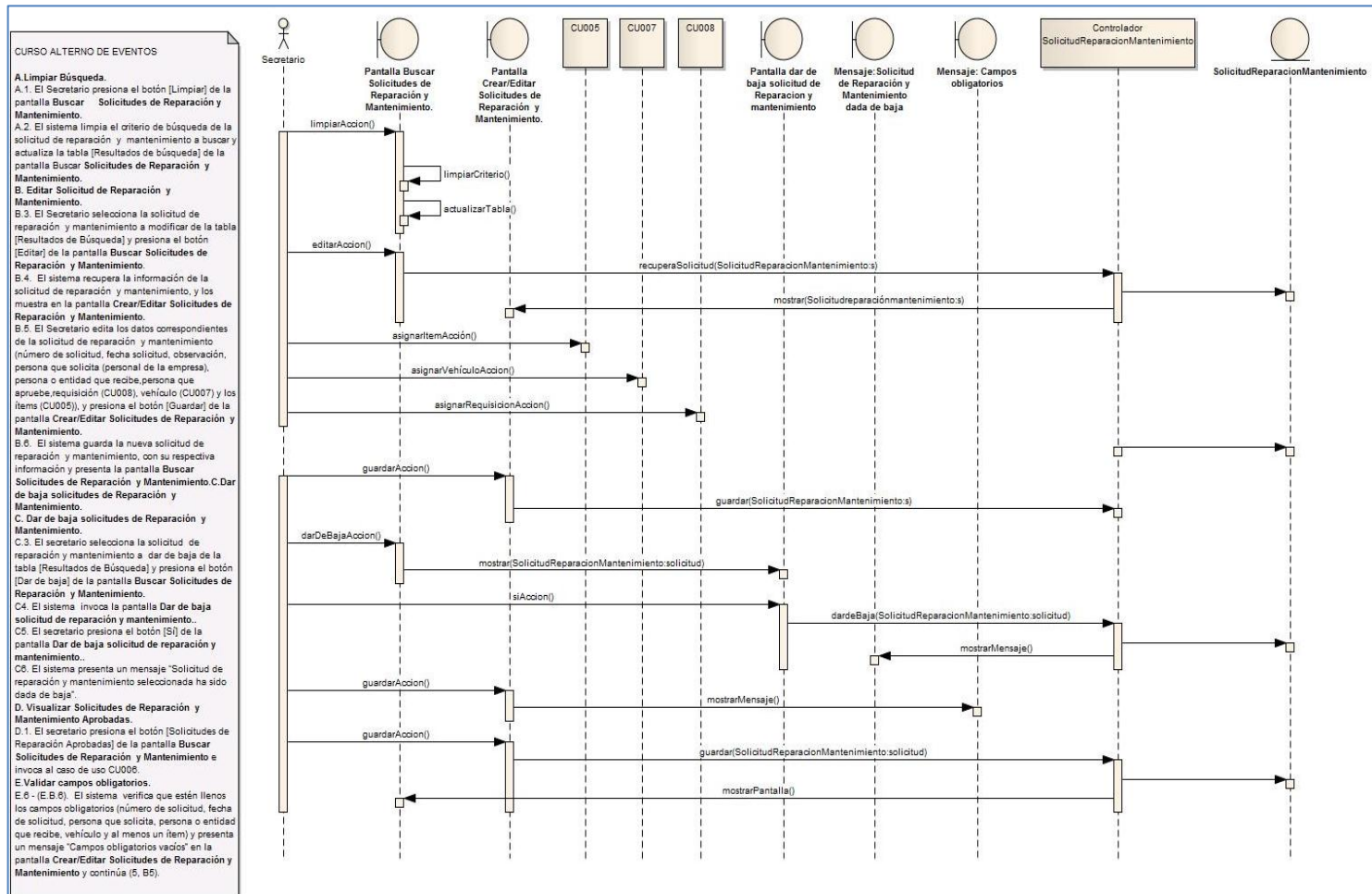


Figura 102. Diagrama de secuencia del CU004 curso alterno de eventos.

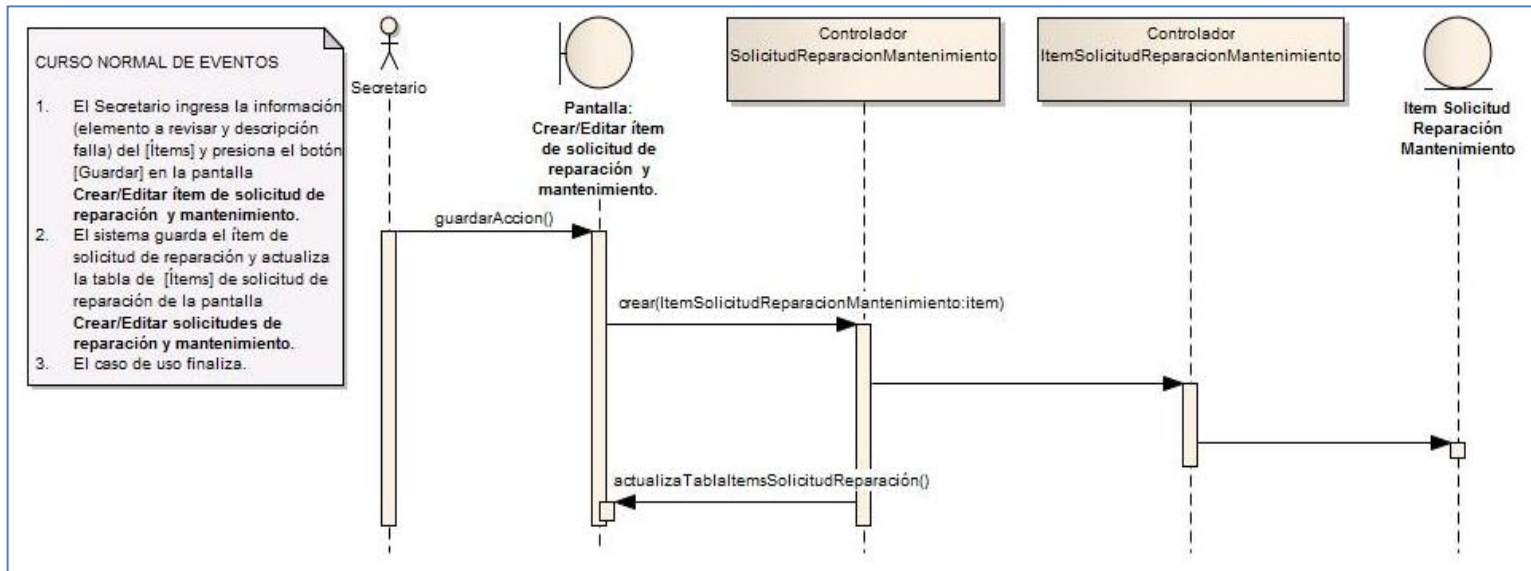


Figura 103. Diagrama de secuencia del CU005 curso normal de eventos.

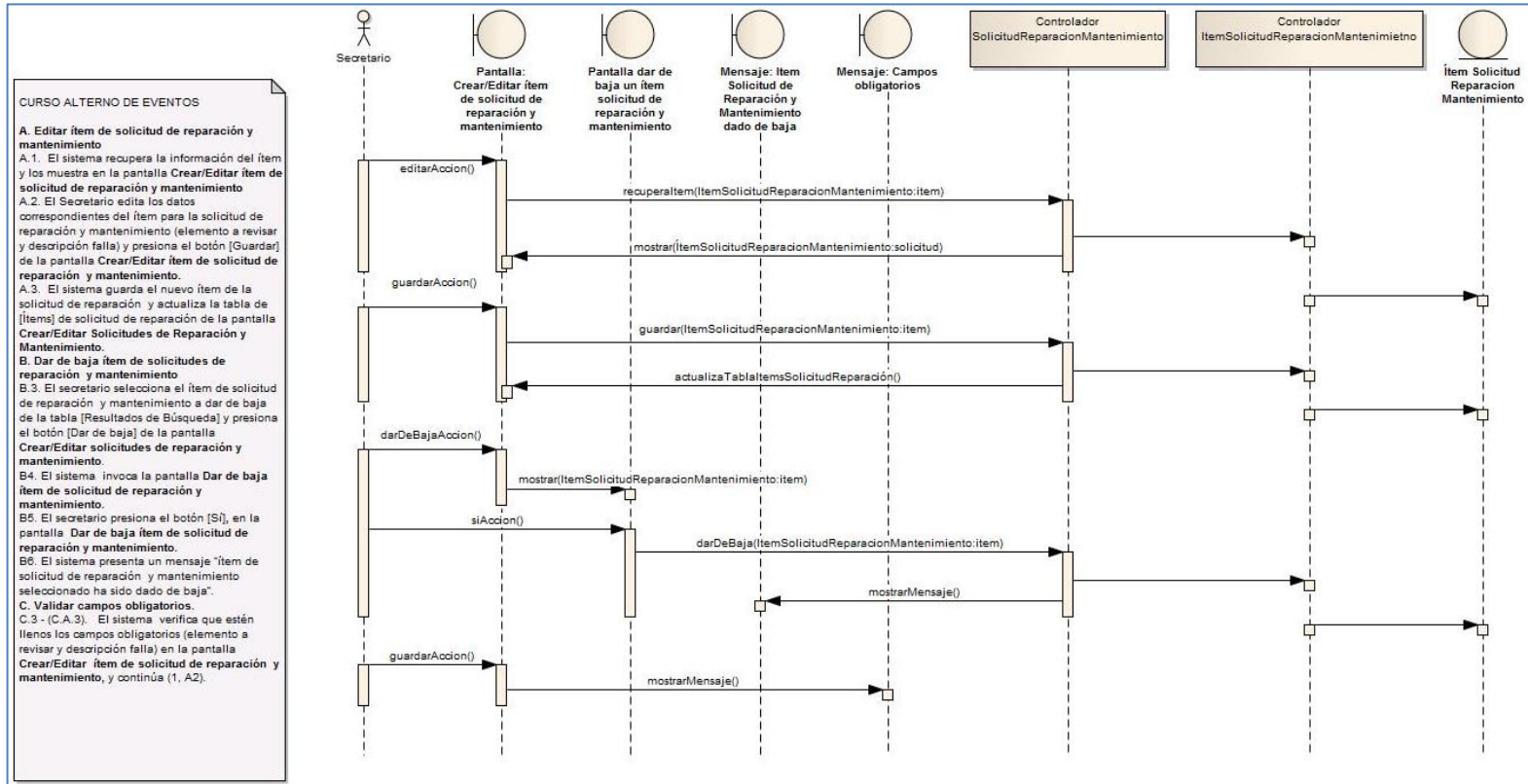


Figura 104. Diagrama de secuencia del CU005 curso alterno de eventos.

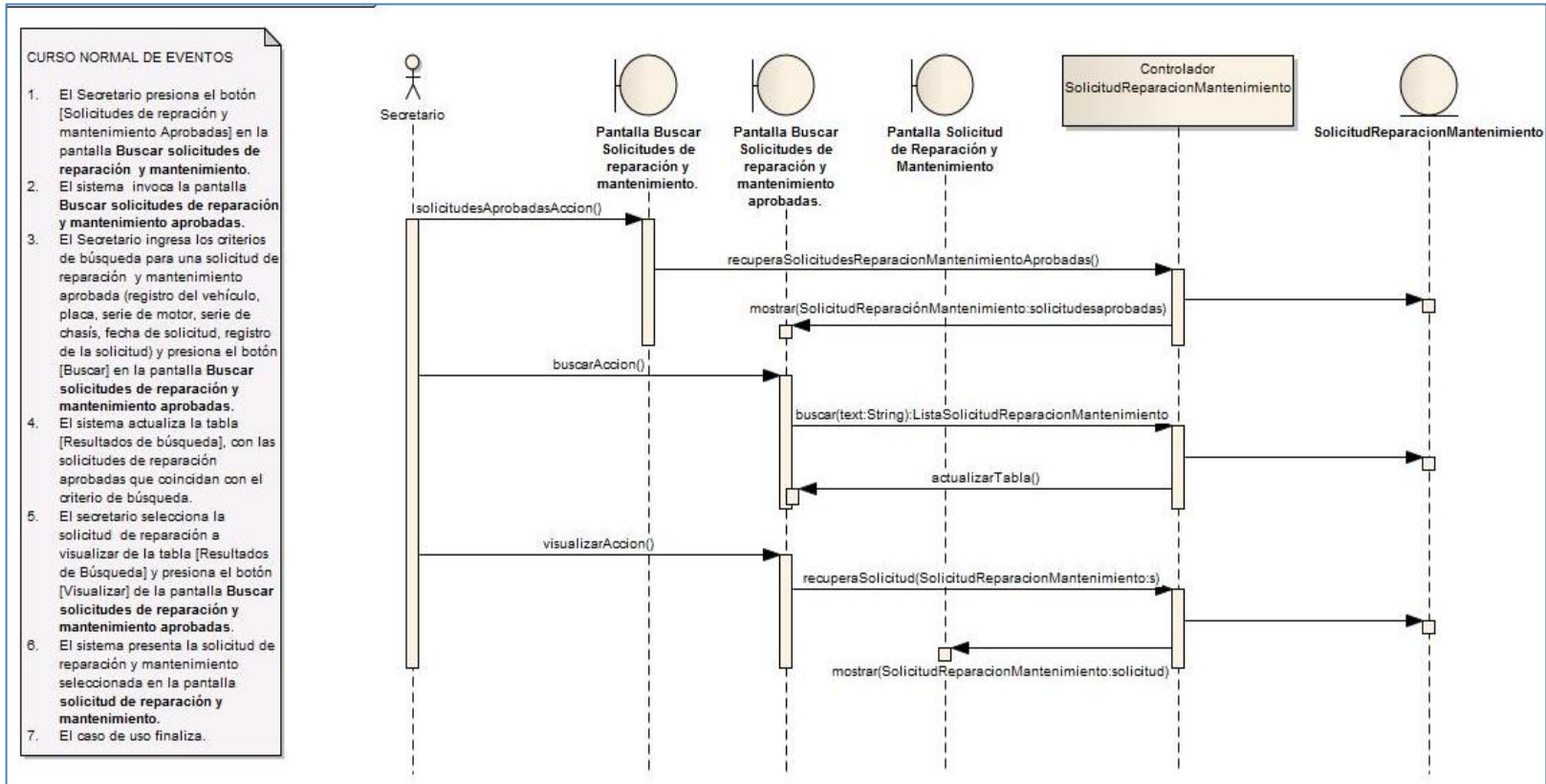


Figura 105. Diagrama de secuencia del CU006 curso normal de eventos.

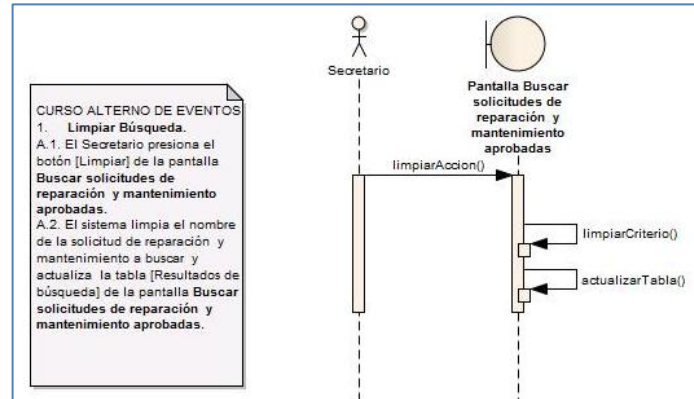


Figura 106. Diagrama de secuencia del CU006 curso alternativo de eventos.

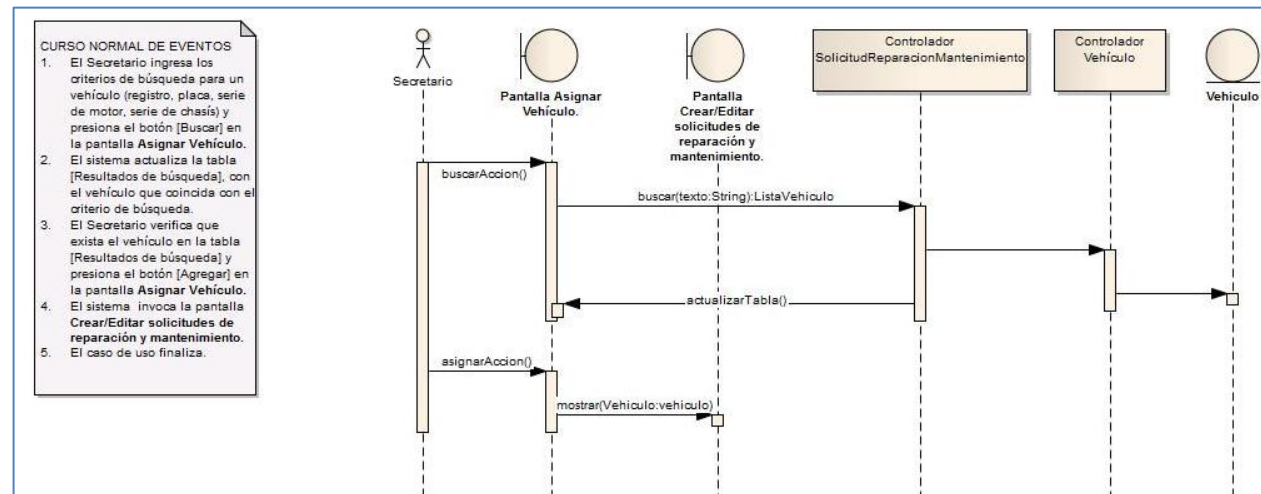


Figura 107. Diagrama de secuencia del CU007 curso normal de eventos.

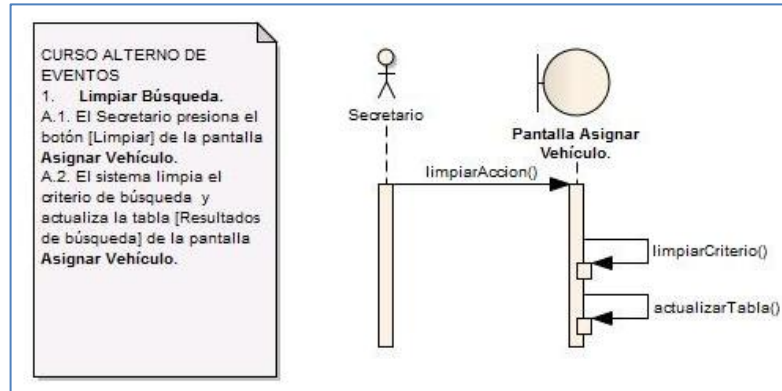


Figura 108. Diagrama de secuencia del CU007 curso alterno de eventos.

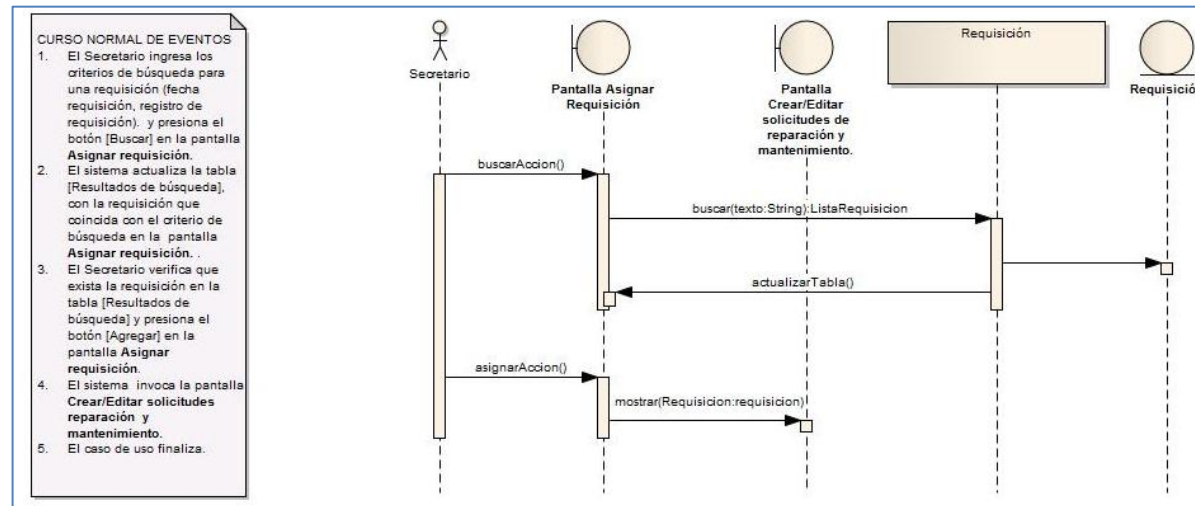


Figura 109. Diagrama de secuencia del CU008 curso normal de eventos.

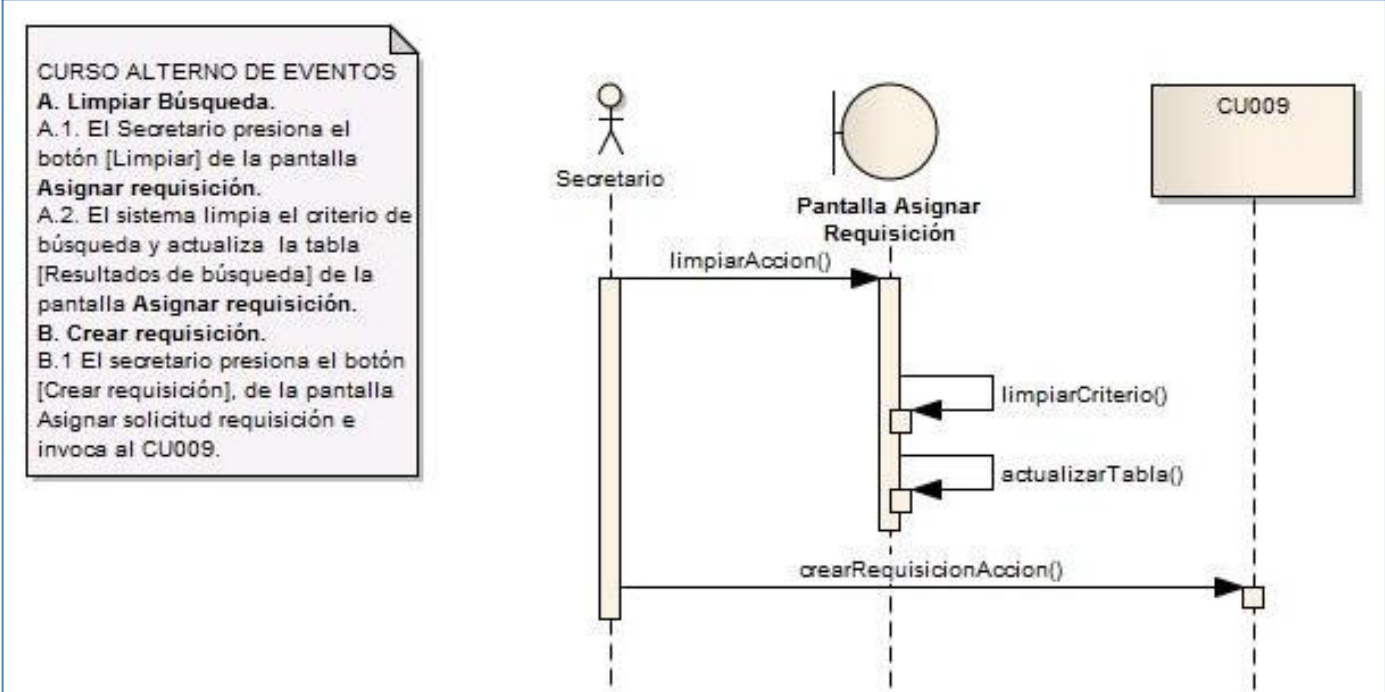


Figura 110. Diagrama de secuencia del CU008 curso alternativo de eventos.

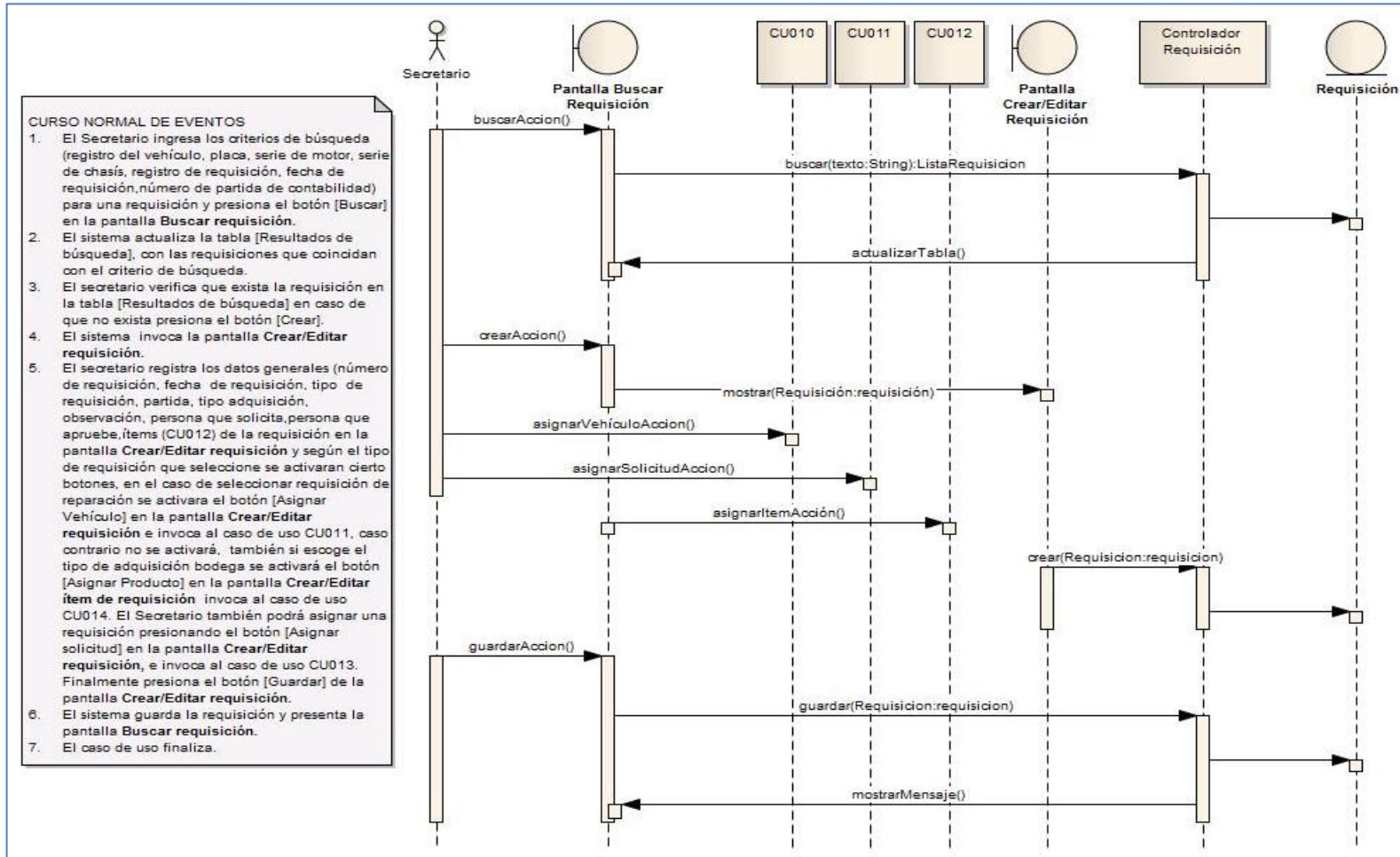


Figura 111. Diagrama de secuencia del CU009 curso normal de eventos.

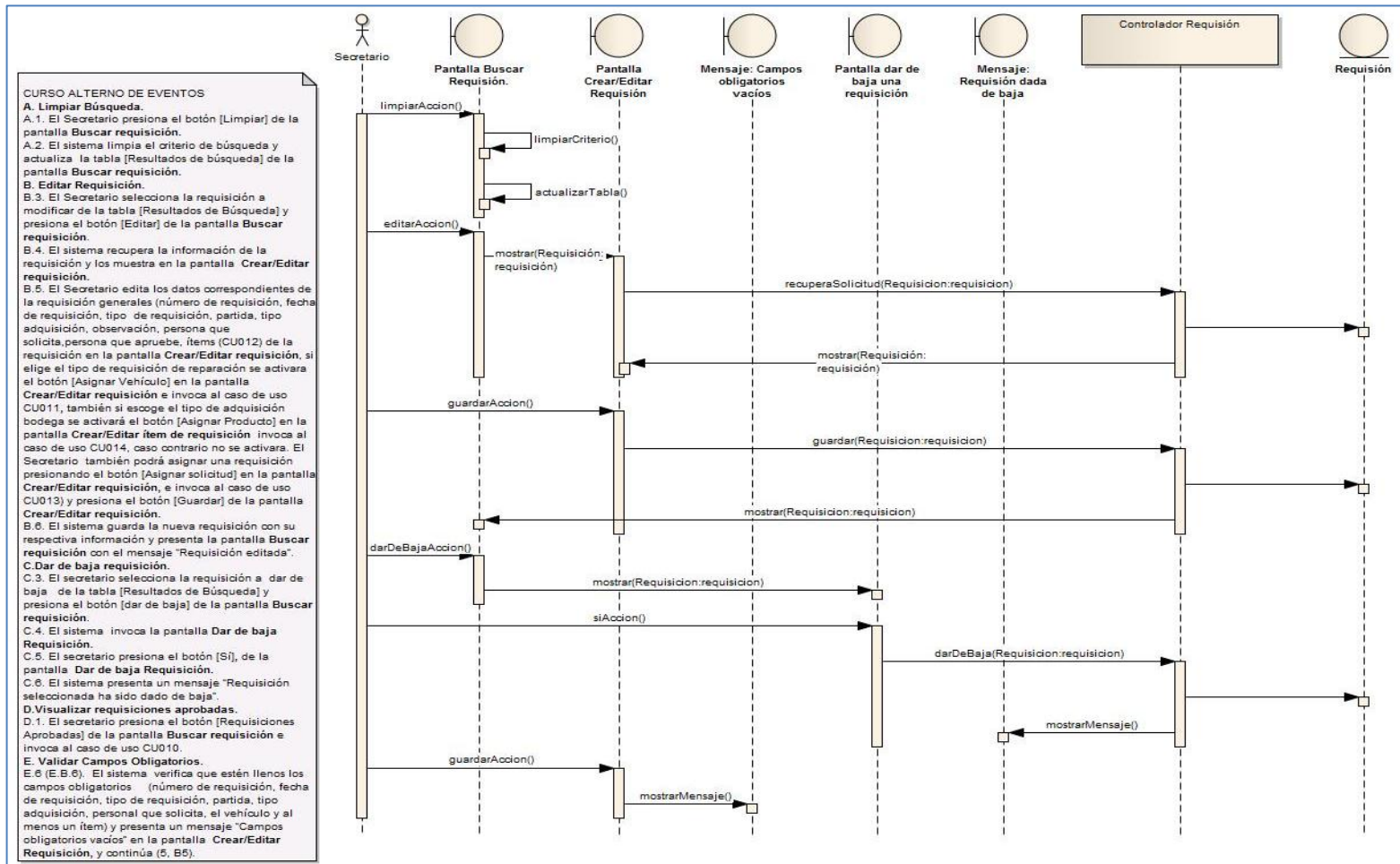


Figura 112. Diagrama de secuencia del CU009 curso alterno de eventos.

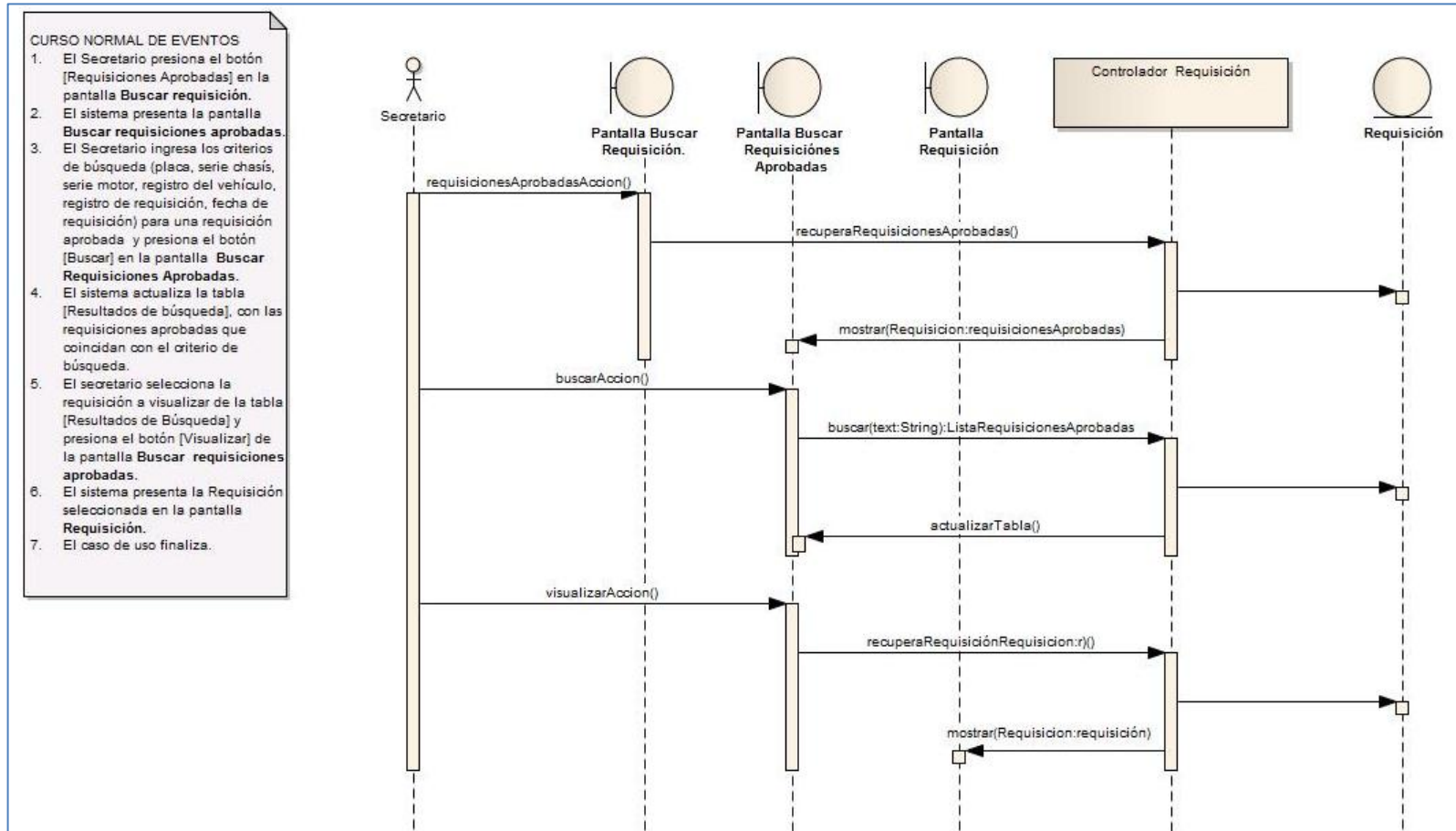


Figura 113. Diagrama de secuencia del CU010 curso normal de eventos.

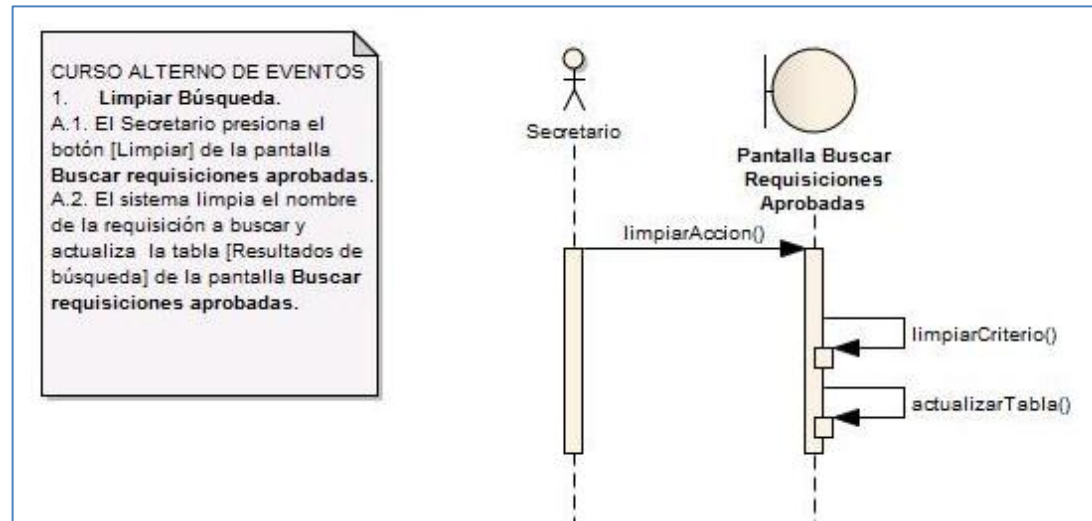


Figura 114. Diagrama de secuencia del CU010 curso alternativo de eventos.

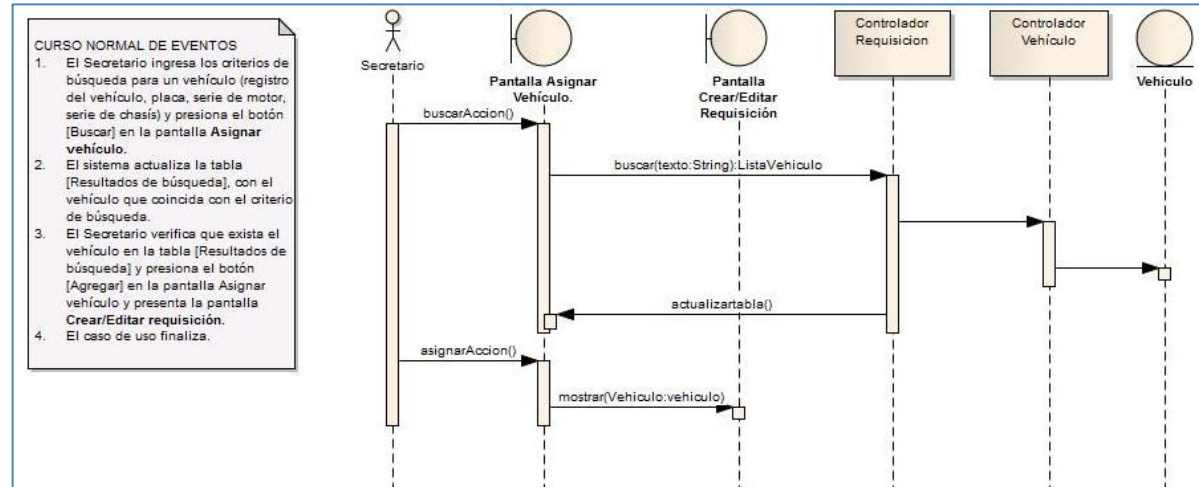


Figura 115. Diagrama de secuencia del CU011 curso normal de eventos.

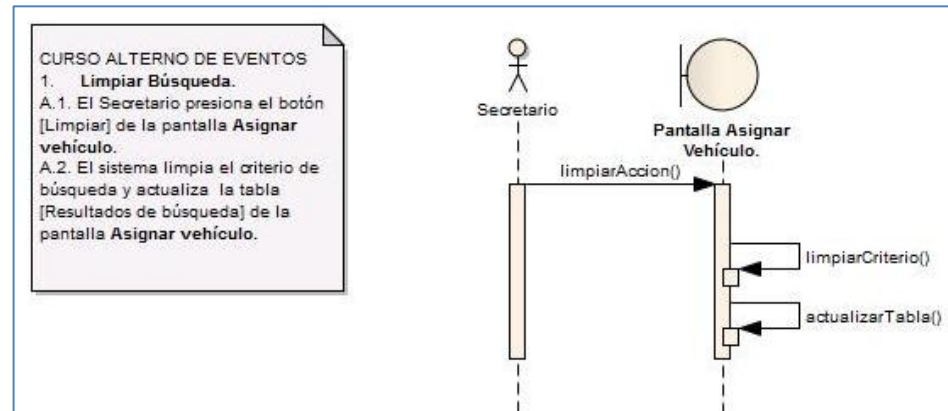


Figura 116. . Diagrama de secuencia del CU011 curso alterno de eventos.

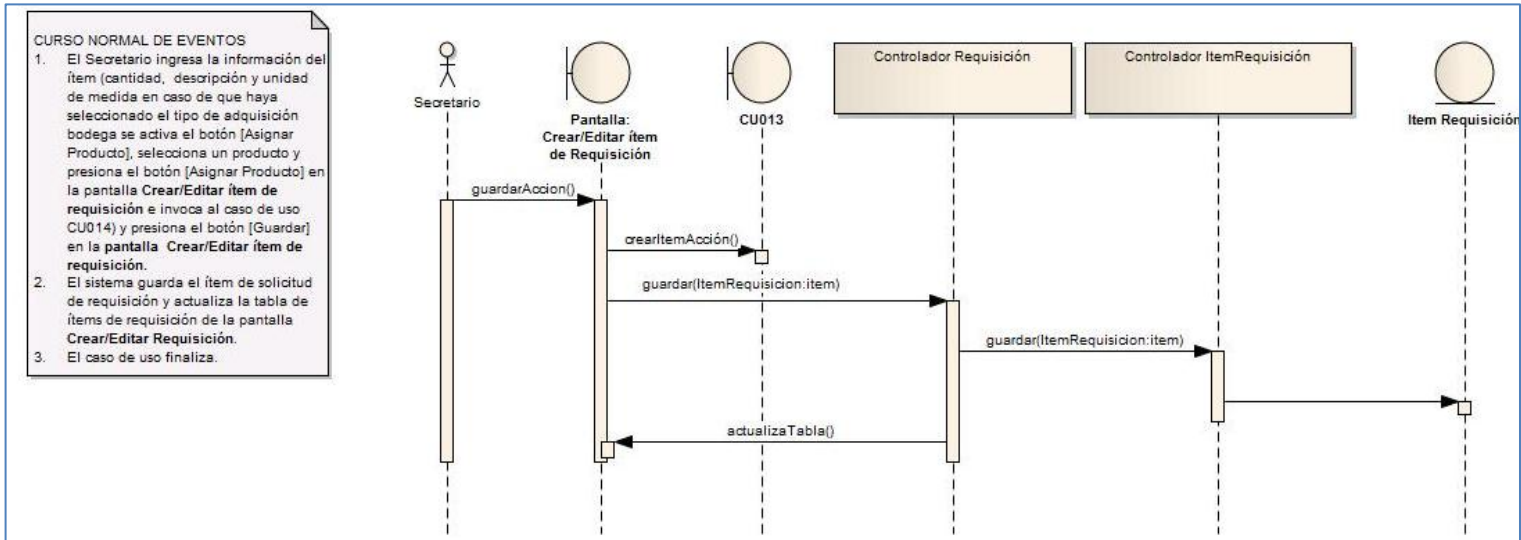


Figura 117. Diagrama de secuencia del CU012 curso normal de eventos.

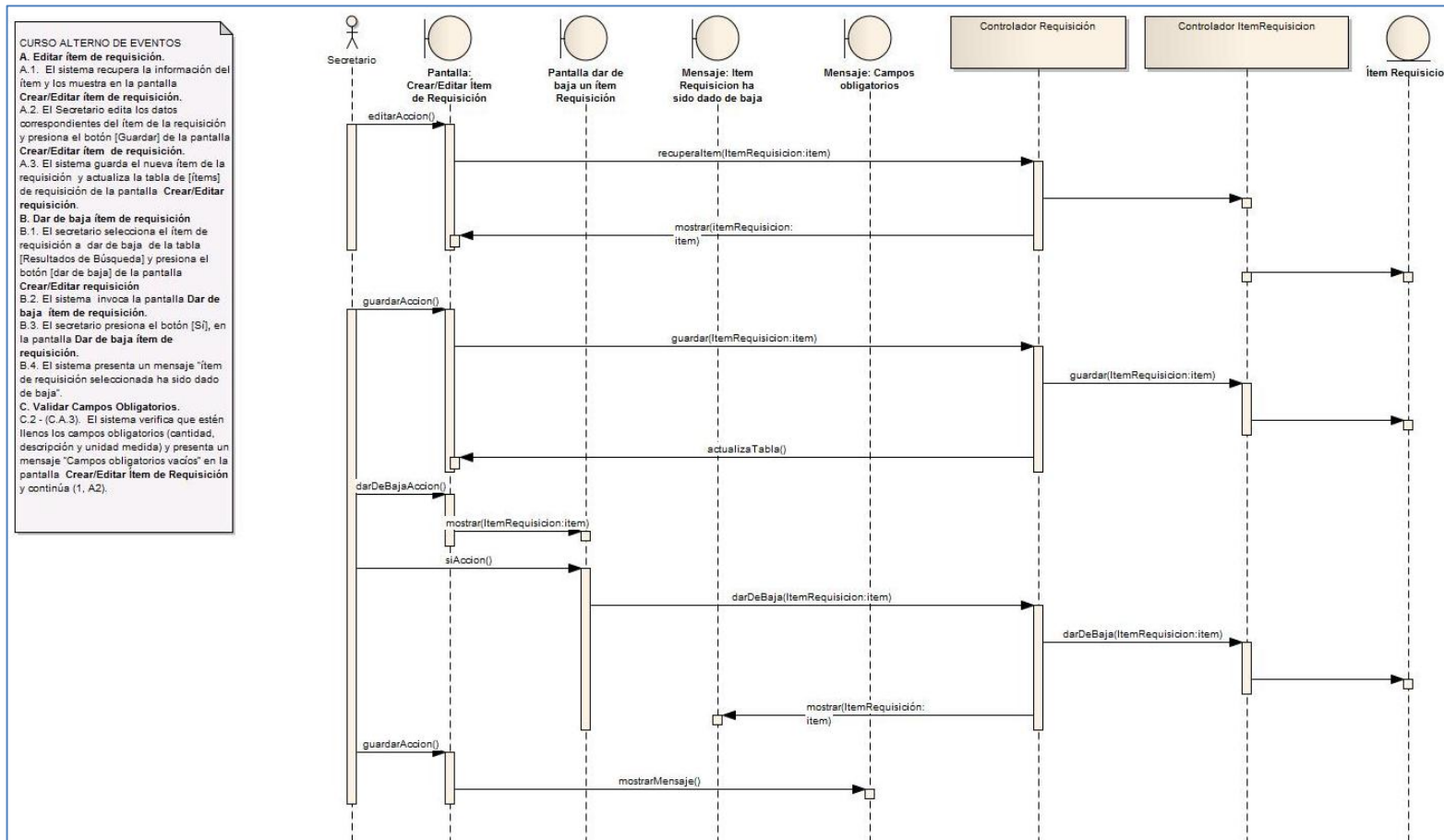


Figura 118. Diagrama de secuencia del CU012 curso alterno de eventos.

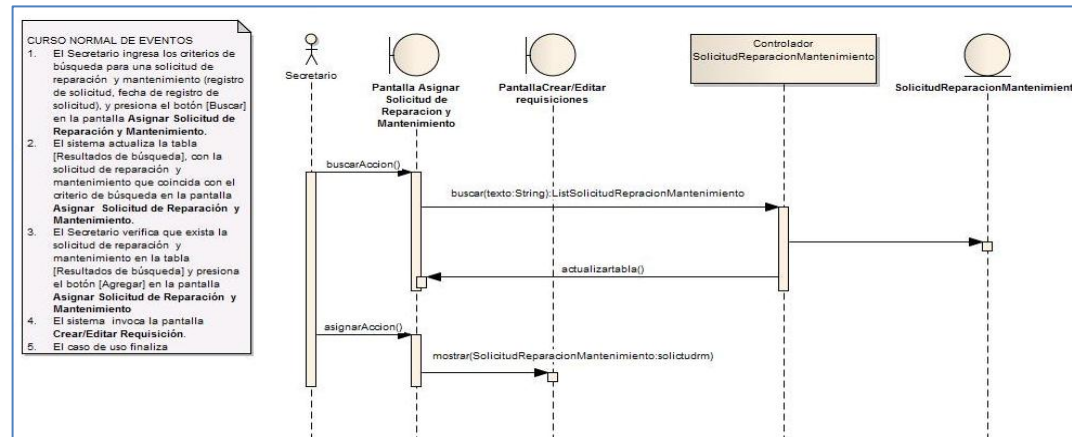


Figura 119. Diagrama de secuencia del CU013 curso normal de eventos.

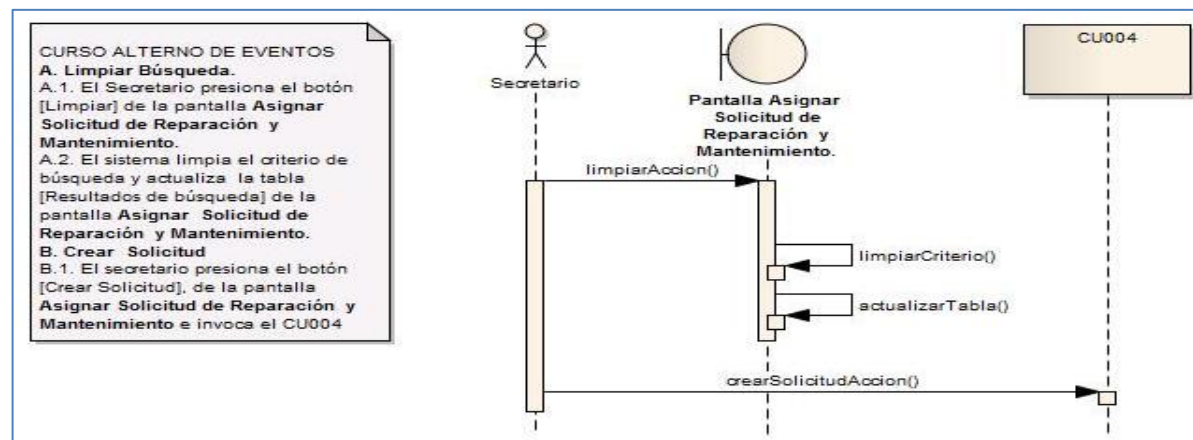


Figura 120. Diagrama de secuencia del CU013 curso alterno de eventos.

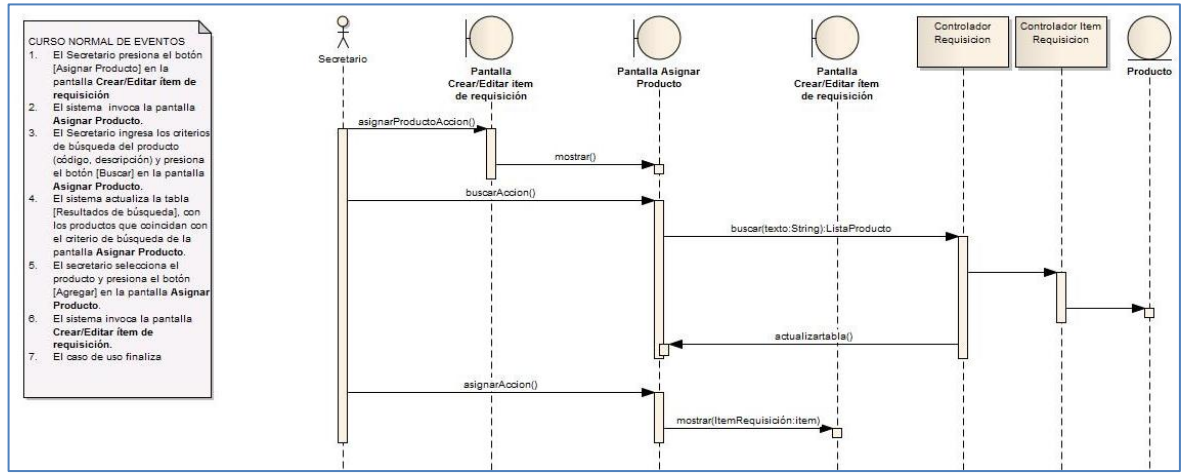


Figura 121. Diagrama de secuencia del CU014 curso normal de eventos.

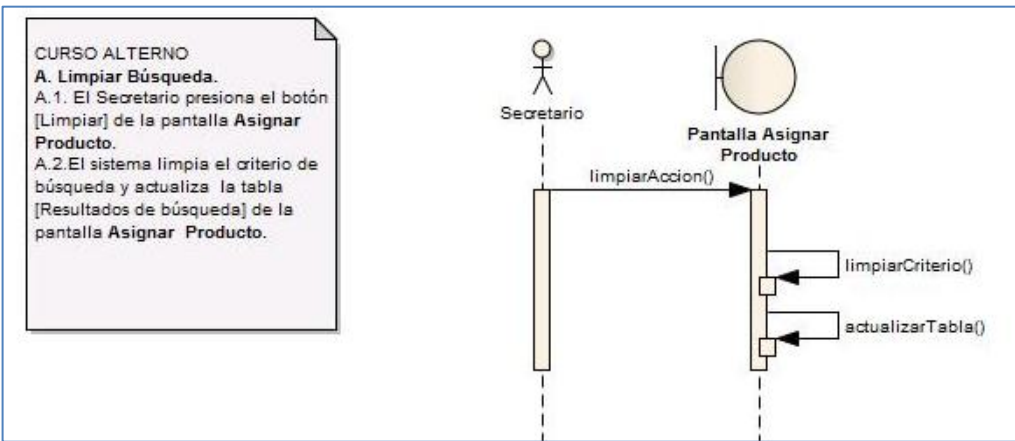


Figura 122. Diagrama de secuencia del CU014 curso alterno de eventos

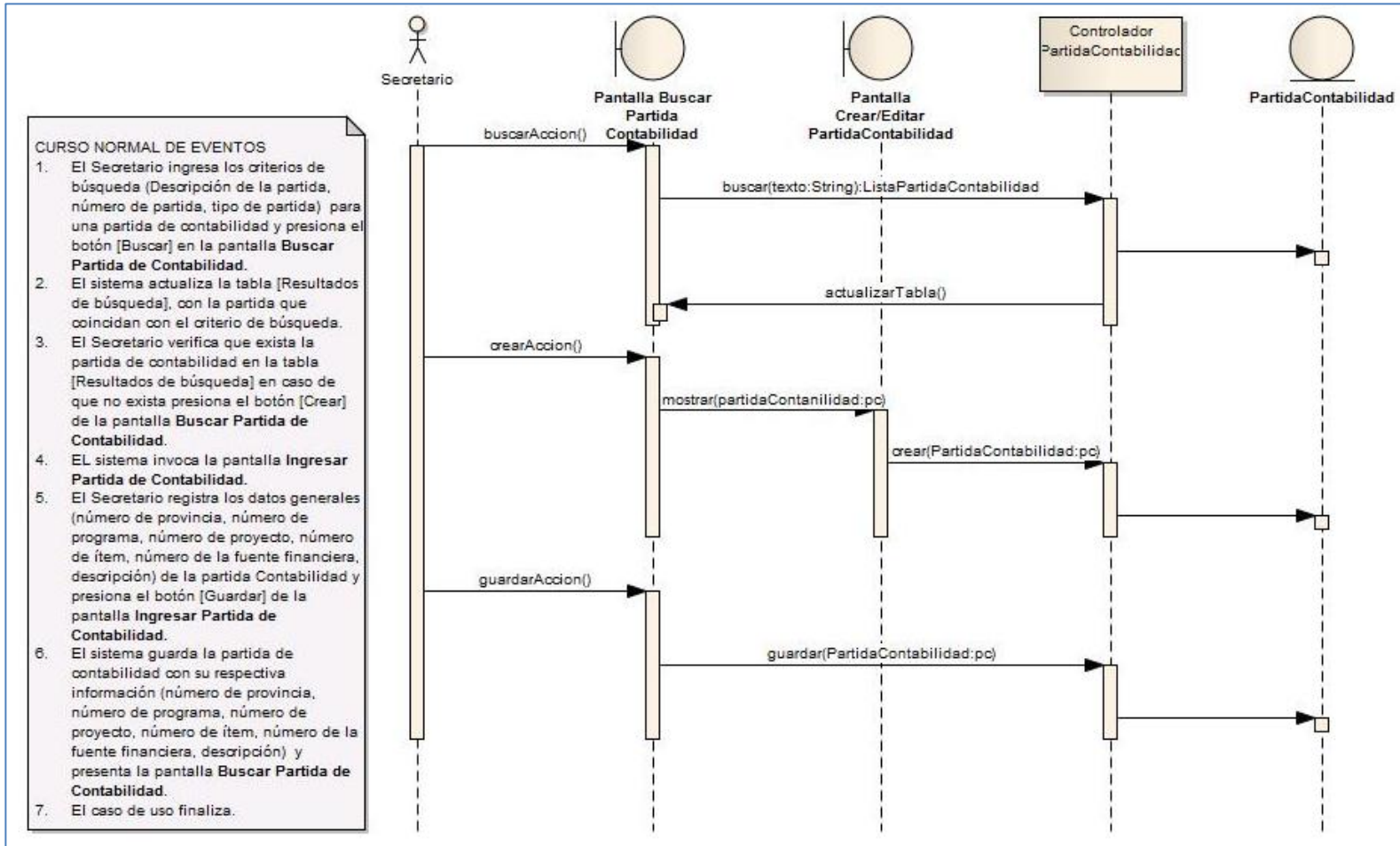


Figura 123. Diagrama de secuencia CU015 curso normal de eventos.

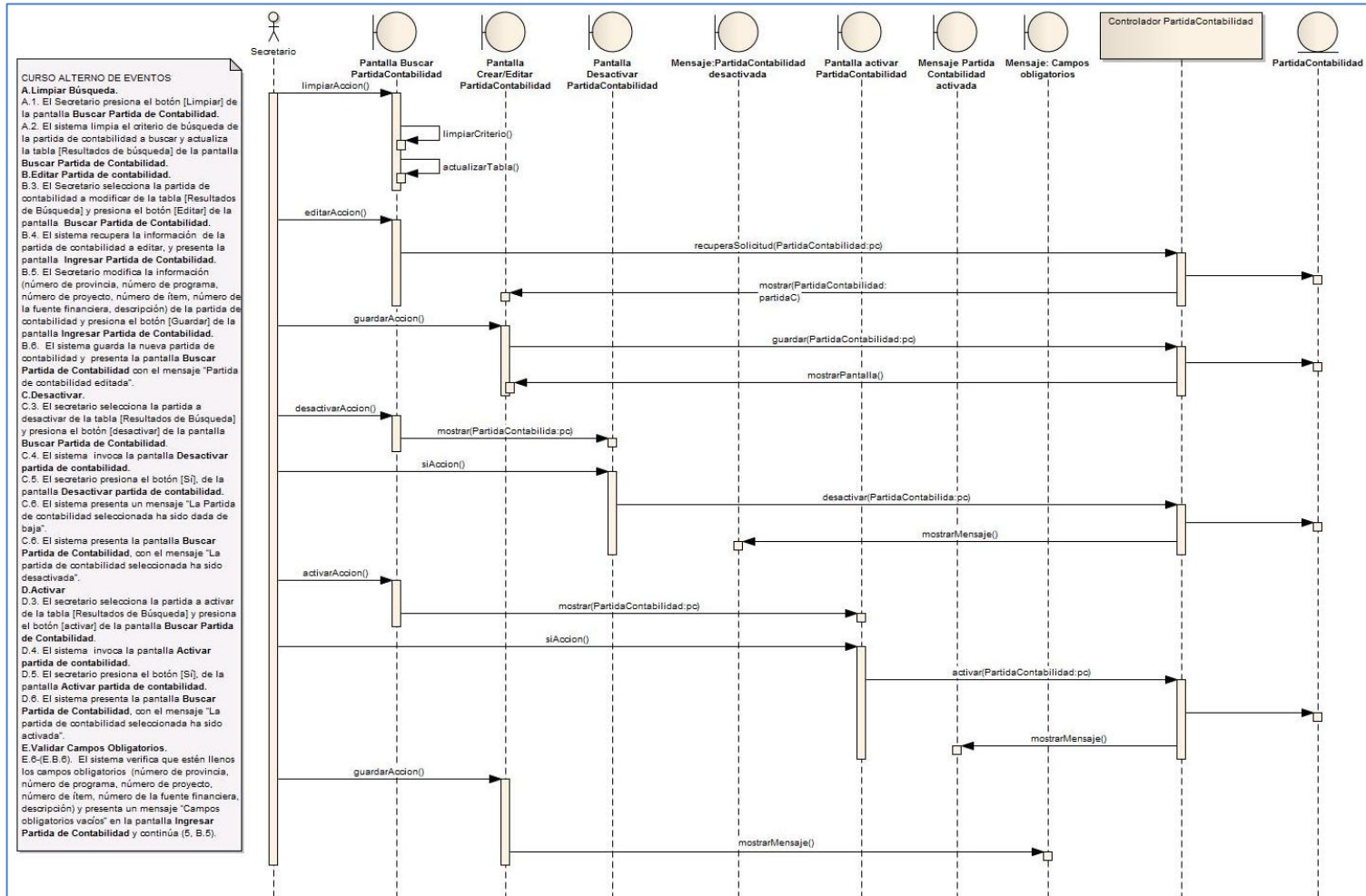


Figura 124. Diagrama de secuencia CU015 curso alterno de eventos.

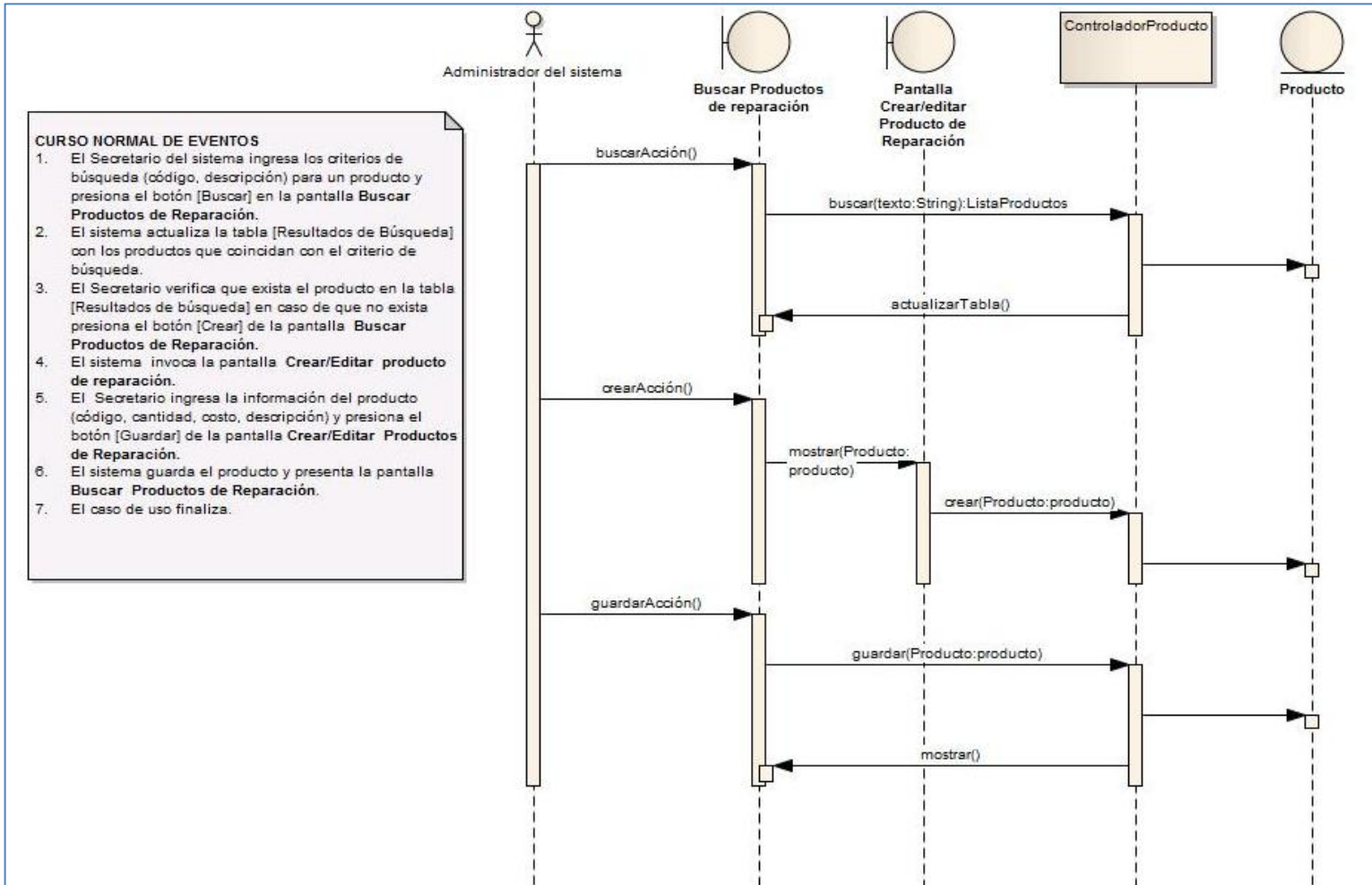


Figura 125. Diagrama de secuencia CU016 curso normal de eventos.

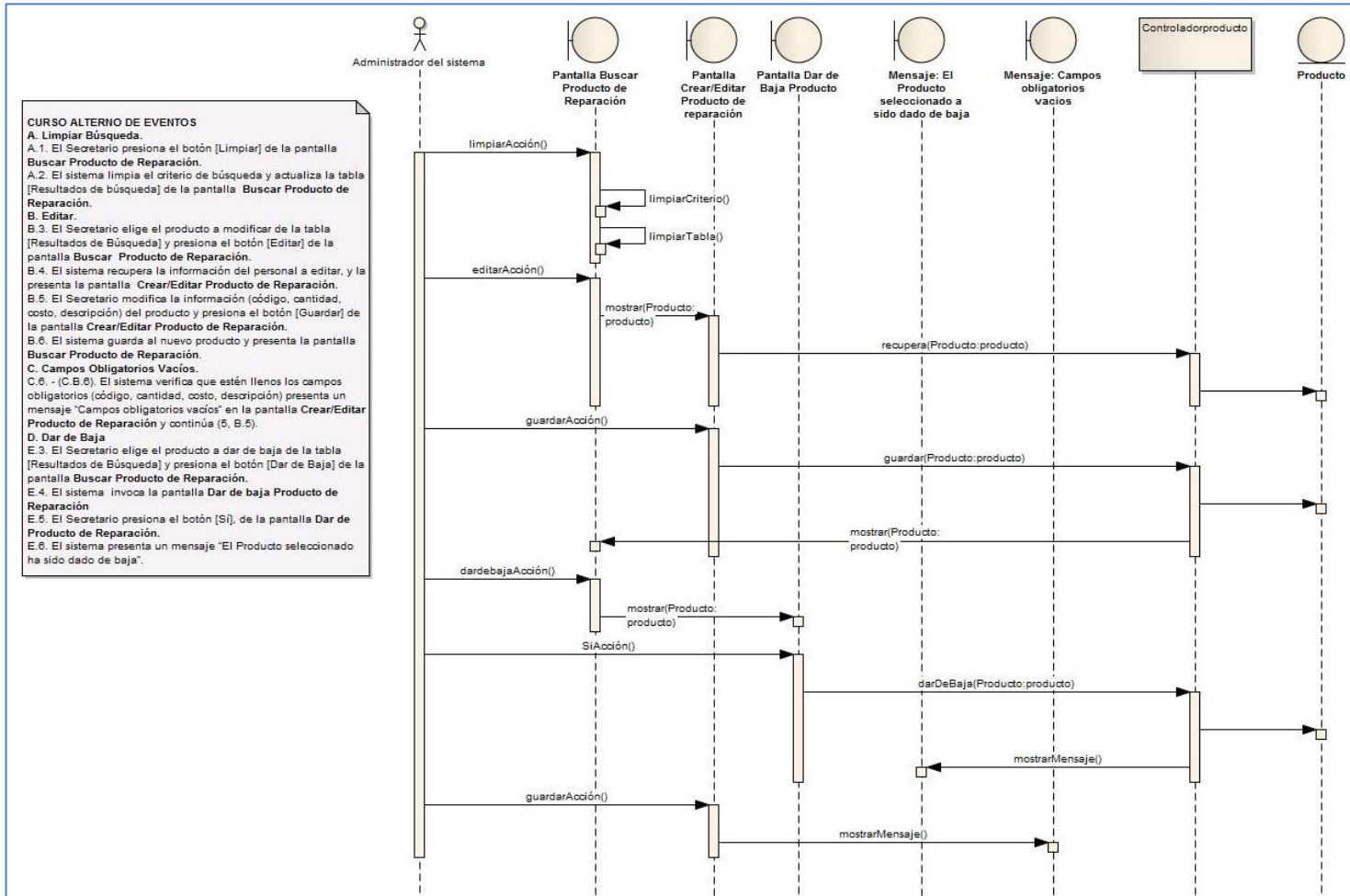


Figura 126. Diagrama de secuencia CU016 curso alterno de eventos.

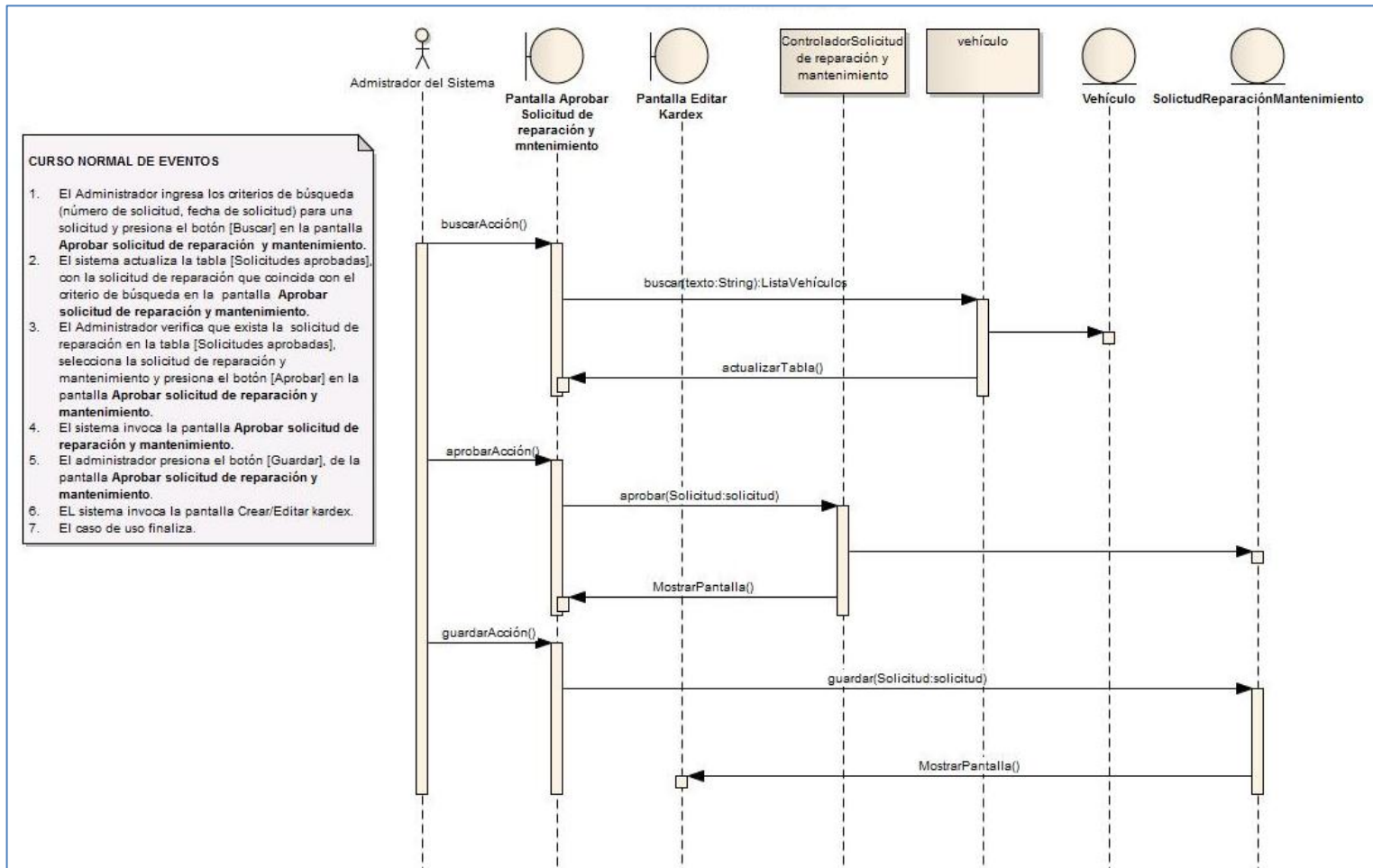


Figura 127. Diagrama de secuencia CU018 curso normal de eventos.

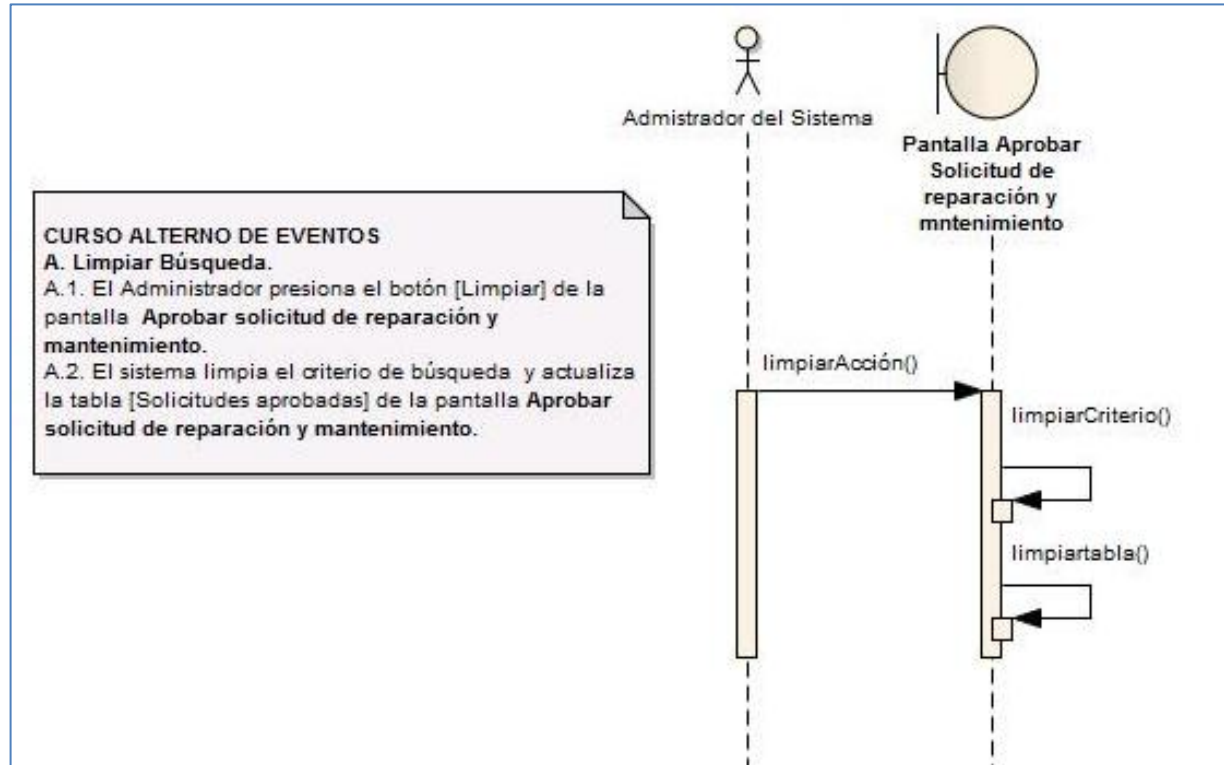


Figura 128. Diagrama de secuencia CU018 curso alterno de eventos.

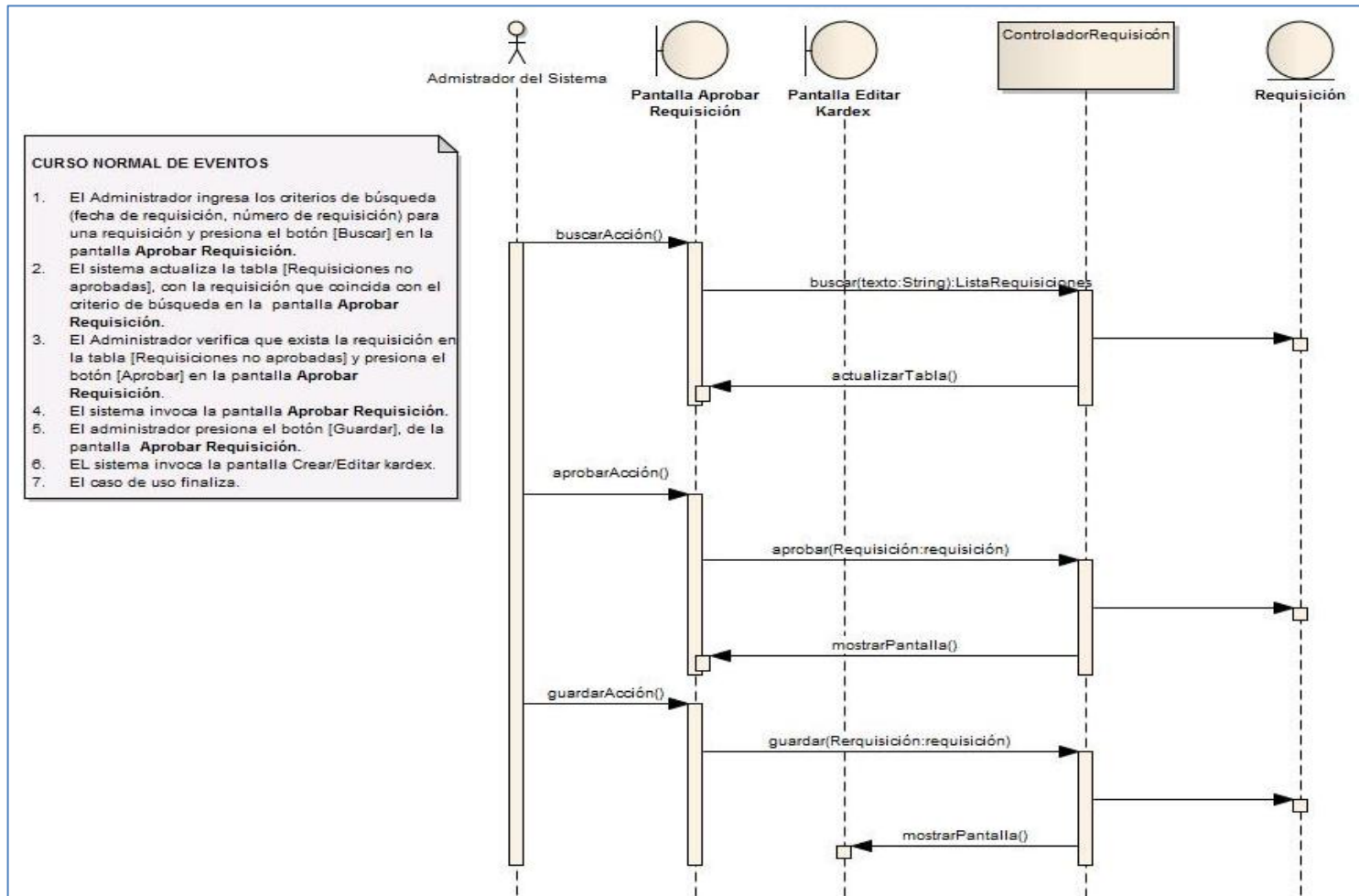


Figura 129. Diagrama de secuencia CU019 curso normal de eventos.

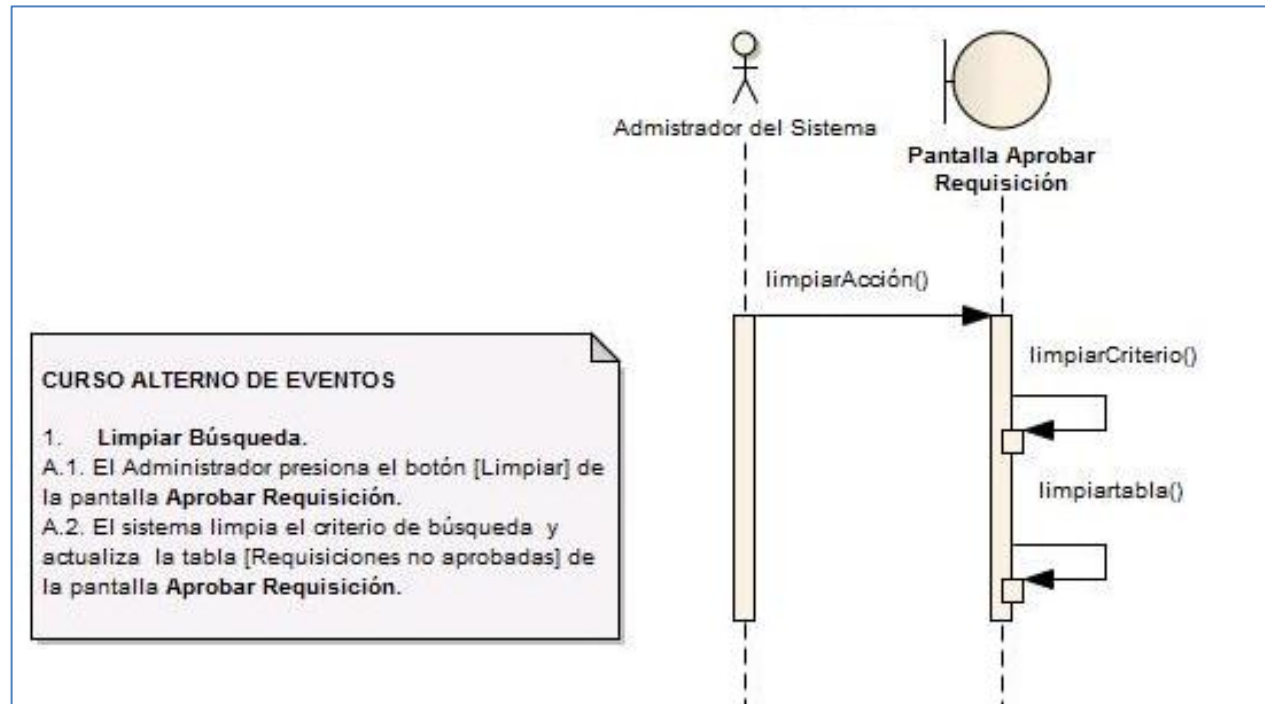


Figura 130. Diagrama de secuencia CU019 curso alterno de eventos.

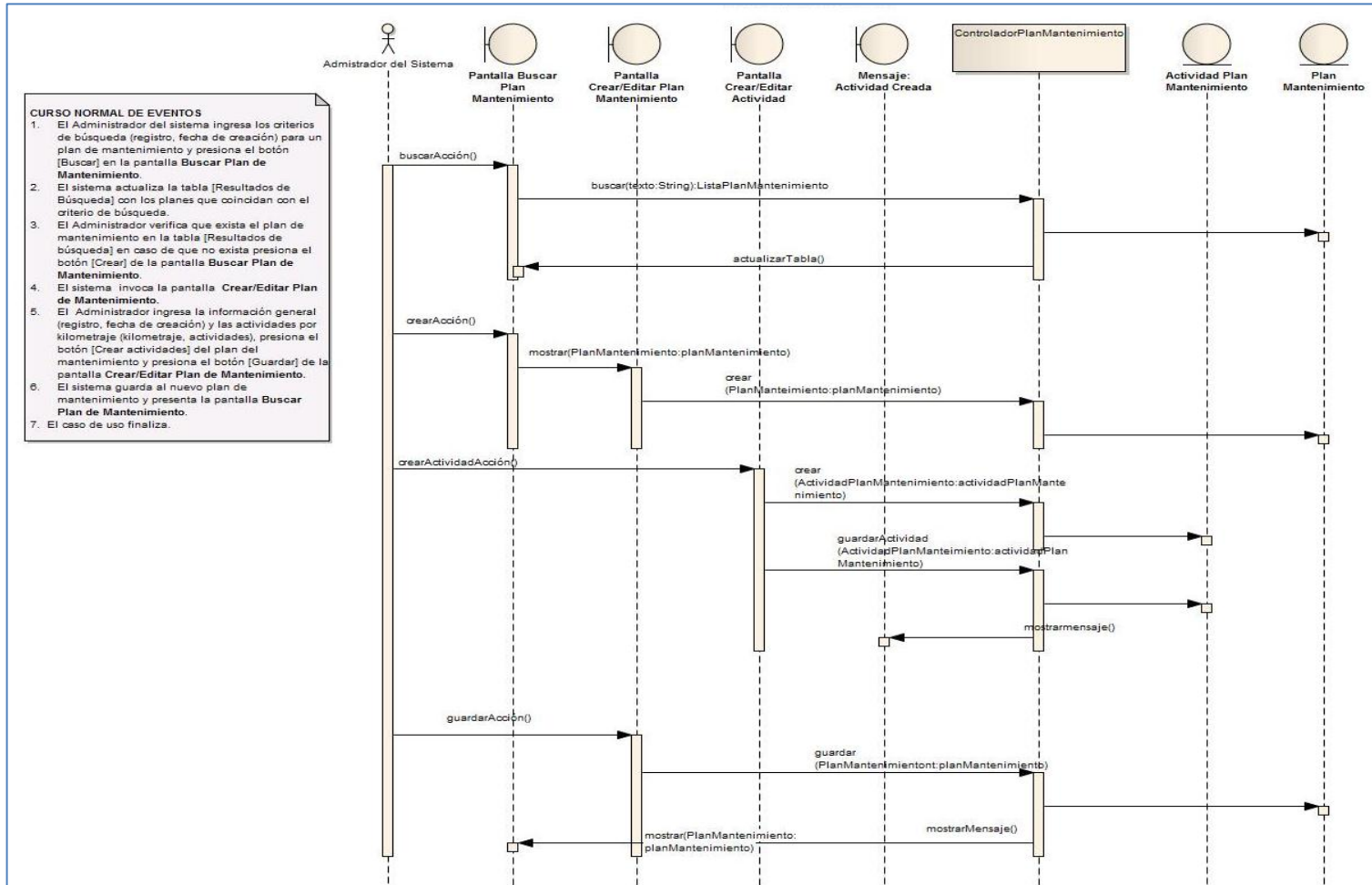


Figura 131. Diagrama de secuencia CU020 curso normal de eventos.

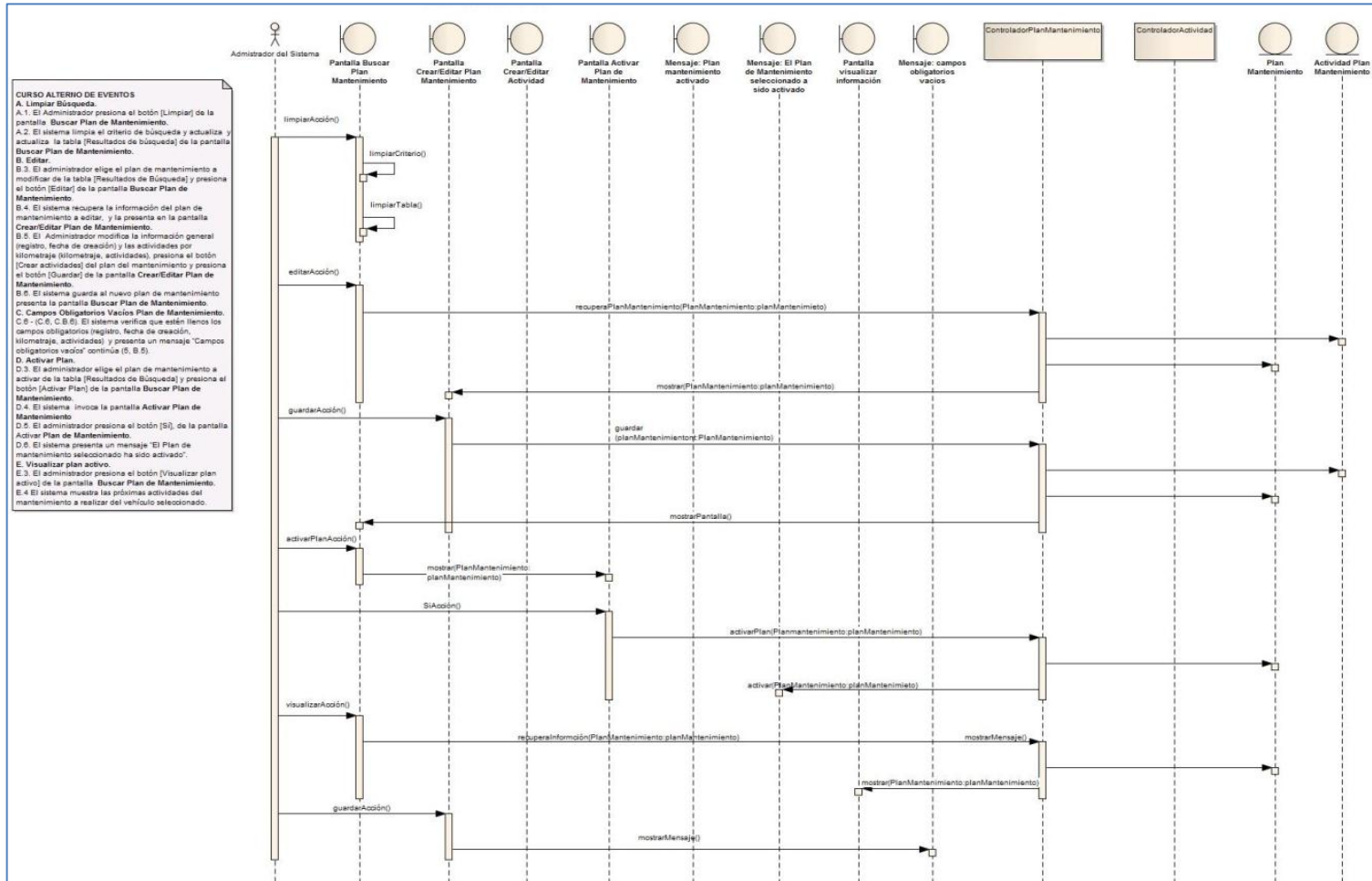


Figura 132. Diagrama de secuencia CU020 curso alterno de eventos.

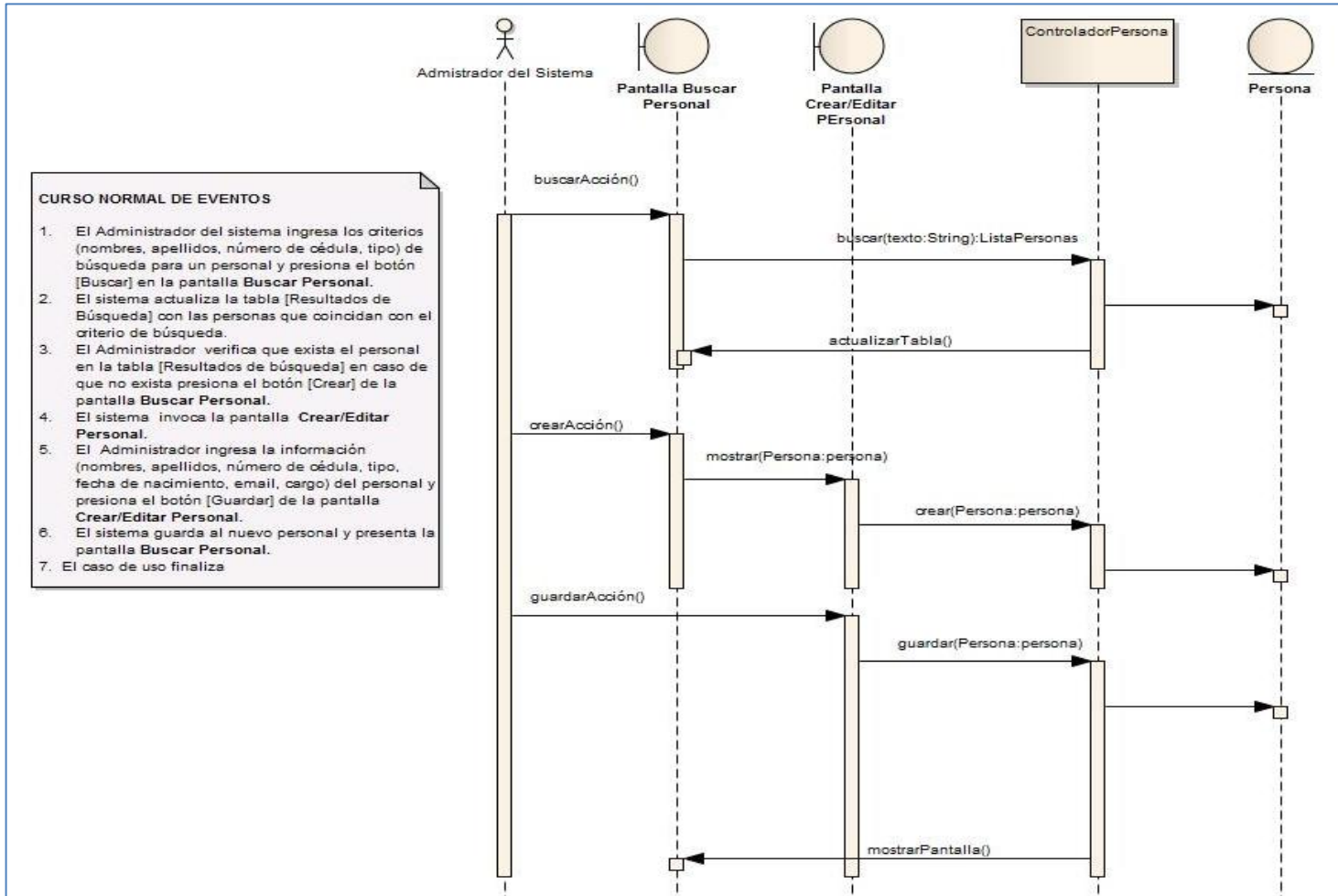


Figura 133. Diagrama de secuencia CU022 curso normal de eventos.

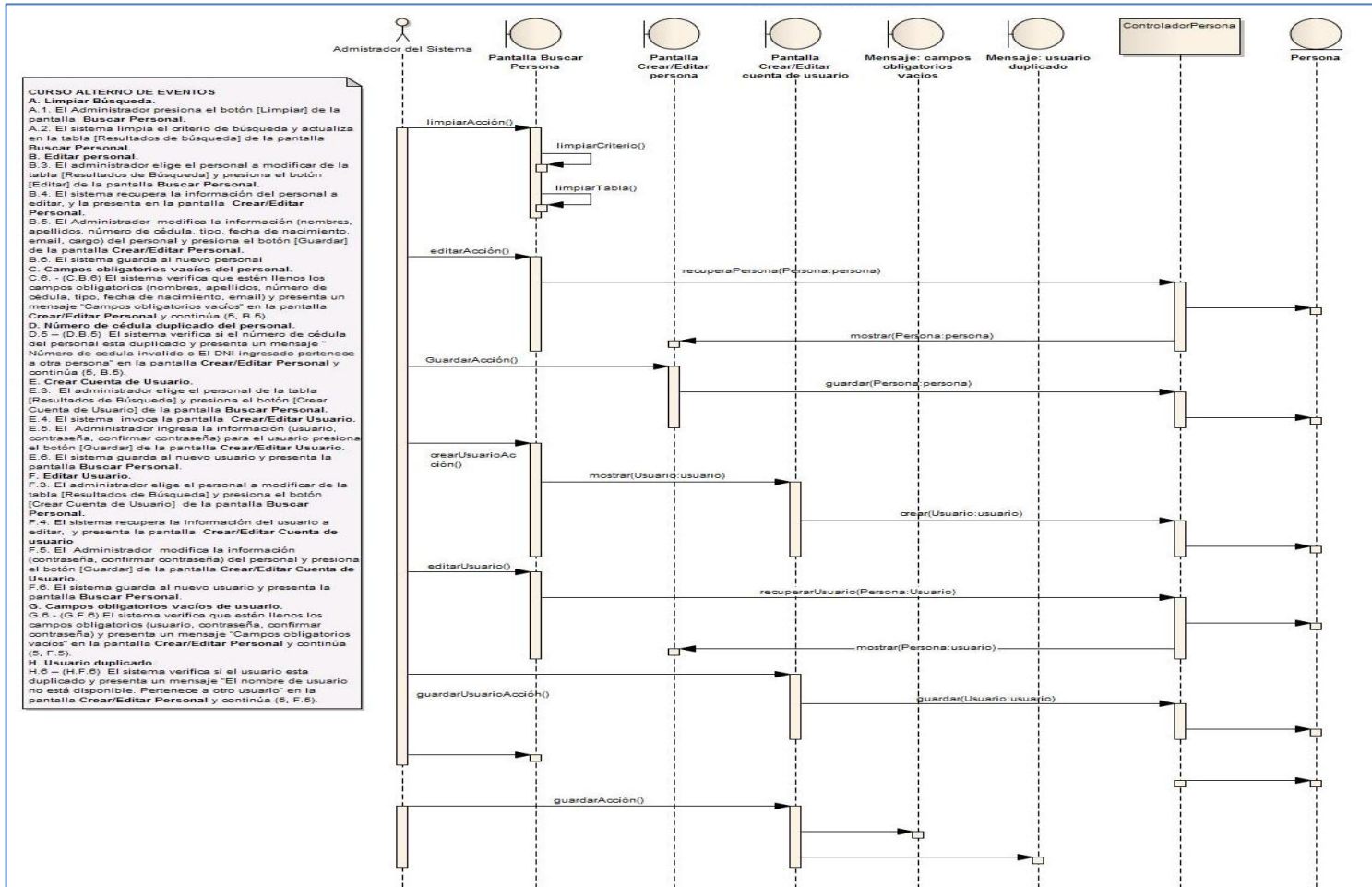


Figura 134. Diagrama de secuencia CU022 curso alterno de eventos.

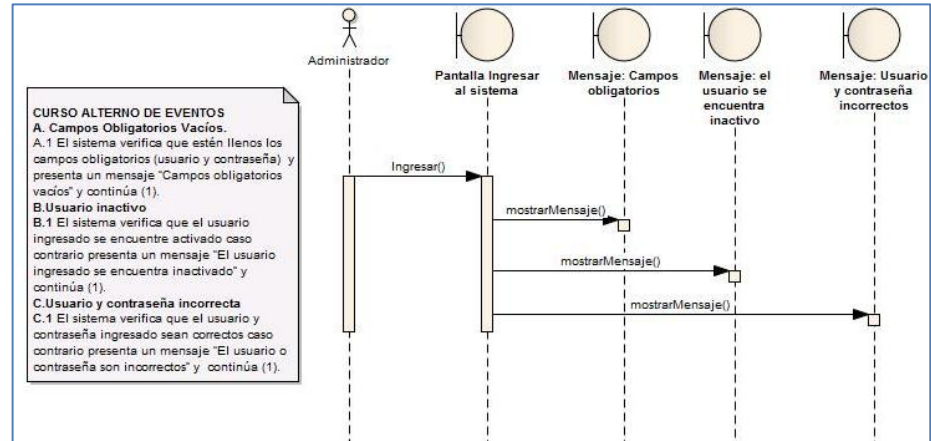


Figura 135. Diagrama de secuencia CU024 curso normal de eventos.

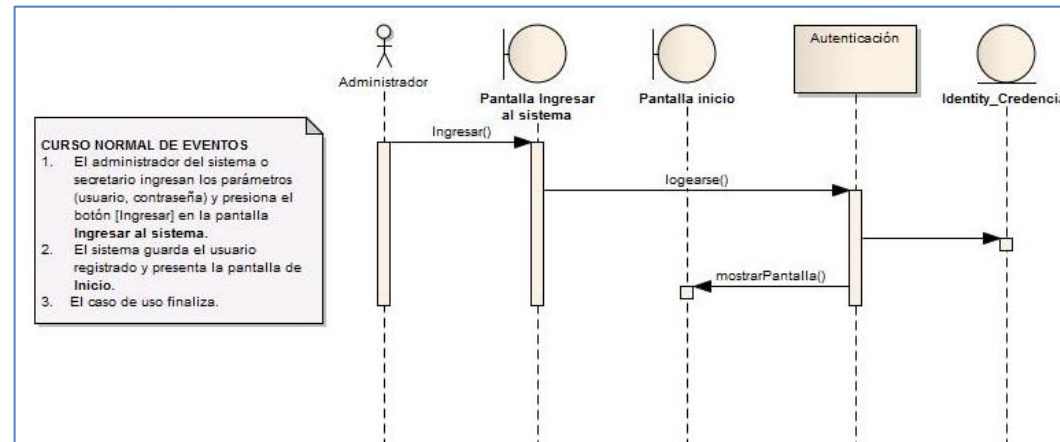


Figura 136. Diagrama de secuencia CU024 curso alterno de eventos.

Anexo 7: Entrevista dirigida a Ing. Juan Carlos Vinueza



**UNIVERSIDAD
NACIONAL
DE LOJA**



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

RESULTADO DE ENTREVISTA

Jueves 17 de Julio del 2014, a partir de 15h00 a 16h00 se realizó una entrevista con el Ing. Juan Carlos Vinueza administrador del control del estado mecánico de los vehículos del Departamento de Mantenimiento Vial del Ministerio de Transporte y Obras Públicas de la ciudad de Loja. La entrevista consistió en precisar ciertos detalles relacionados con el número de registro del vehículo, número de registro de kardex, número de registro de solicitud de reparación y mantenimiento, número de registro de requisición, estados de evaluación de los vehículos. Es así como se concluyó que el registro del vehículo lo digitará el administrador por el motivo que el número viene establecido desde la central de Quito, el número de la kardex será el mismo del registro del vehículo, el número de registro de requisición de bienes y servicios deberá continuar desde la numeración 900, el número registro de requisición de reparación deberá continuar desde la numeración 200, el número de registro de la solicitud de reparación y mantenimiento deberá continuar desde la numeración 8000, así también se determinó que serán generadas de manera secuencialmente. De igual forma se acordó que para la evaluación (ficha técnica) de los vehículos se utilizará como estados bueno y malo.



Ing. Juan Carlos Vinueza

Entrevistado.



**UNIVERSIDAD
NACIONAL
DE LOJA**



*Área de la Energía, las Industrias y los Recursos Naturales No
Renovables*

CARRERA DE INGENIERÍA EN SISTEMAS

RESULTADO DE ENTREVISTA

Martes 15 de abril del 2014, a partir de 13h30 a 15h30 se realizó una entrevista con el Ing. Juan Carlos Vinueza administrador del control del estado mecánico de los vehículos del Departamento de Mantenimiento Vial del Ministerio de Transporte y Obras Públicas de la ciudad de Loja. La entrevista consistió en aclarar varias interrogantes que se tenía del sistema a realizar para de esta manera comprender y conocer a fondo el funcionamiento de las tareas que se realizan en el control de los vehículos.

Ing. Juan Carlos Vinueza.
Administrador.

Carla Lisbeth Romero Córdova.
Responsable

Yesica Norrelly Saquicela Celi.
Responsable

Anexo 8: Entrevista dirigida a Sr. Danny Castillo



**UNIVERSIDAD
NACIONAL
DE LOJA**



*Área de la Energía, las Industrias y los Recursos Naturales No
Renovables*

CARRERA DE INGENIERÍA EN SISTEMAS

RESULTADO DE ENTREVISTA

Martes 22 de abril del 2014, a partir de 18h00 a 19h00 se realizó una entrevista al especialista en mecánica Dany Castillo, para conocer cómo se realiza un plan general del mantenimiento de vehículos, quien manifestó que dicho plan se realiza según el kilometraje del vehículo. A continuación se lista las actividades del plan de mantenimiento que son necesarias de acuerdo al tiempo de recorrido del vehículo en kilómetros:

KILOMETROS	ACTIVIDADES
5000 km	Reajuste de suspensión. Revisión de aceite. Revisión de niveles de agua del radiador. Revisión de niveles de líquidos de freno. Revisión de niveles de líquidos de plumas. Revisión de niveles de aceite hidráulico.
10000 km	Realizar un ABC de frenos Realizar un cambio de aceite de motor. Revisión de niveles de agua del radiador. Revisión de niveles de líquidos de freno. Revisión de niveles de líquidos de plumas. Revisión de niveles de aceite hidráulico.
15000 km	Realizar actividades de 5000 km + Cambio de filtro de aire.

20000 km	Realizar actividades de 10000 km
25000 km	Realizar actividades de 10000 km + Cambio de filtros de combustibles.
30000 km	Realizar actividades de 5000 km
35000 km	Realizar actividades de 10000 km
40000 km	Realizar actividades de 15000 km
45000 km	Realizar actividades de 20000 km
50000 km	Cambio de suspensión. Realizar engrasamiento de puntas. Realizar engrasamiento de ejes. Realizar engrasamiento de rulimanes. Realizar un ABC de frenos. Realizar cambio de banda de accesorios. Realizar cambio de banda de distribución. Cambio de filtro de combustible. Cambio de bujías.
55000 km	Realizar actividades de 5000 km
60000 km	Realizar actividades de 10000 km
65000 km	Realizar actividades de 15000 km
70000 km	Realizar actividades de 20000 km
75000 km	Realizar actividades de 25000 km
80000 km	Realizar actividades de 30000 km
85000 km	Realizar actividades de 35000 km
90000 km	Realizar actividades de 40000 km
95000 km	Realizar actividades de 45000 km
100000 km	Realizar actividades de 50000 km
105000 km	Realizar actividades de 5000 km
110000 km	Realizar actividades de 10000 km
115000 km	Realizar actividades de 15000 km
120000 km	Realizar actividades de 20000 km
125000 km	Realizar actividades de 25000 km
130000 km	Realizar actividades de 30000 km
135000 km	Realizar actividades de 35000 km

140000 km	Realizar actividades de 40000 km
145000 km	Realizar actividades de 45000 km
150000 km	Realizar cambio de motor



Dany Castillo.

Mecánico.

Anexo 9: Entrevista dirigida al Sr. Yovanni Alvarado



**UNIVERSIDAD
NACIONAL
DE LOJA**



*Área de la Energía, las Industrias y los Recursos Naturales No
Renovables*

CARRERA DE INGENIERÍA EN SISTEMAS

RESULTADO DE ENTREVISTA

Lunes 05 de mayo del 2014, a partir de 13h00 a 14h00 se realizó una entrevista al Sr. Yovany Alvarado encargado del registro de requisiciones y solicitudes de reparación de los vehículos para la adquisición de repuestos, quien informó que existen dos tipos de requisiciones una que sirve para requisiciones de repuesto de los vehículos y otra para bienes y servicios a continuación se explica el proceso para realizar dichos documentos:

Primero manifestó que cualquier mecánico u otra persona del mismo departamento realizan un pedido sobre algún repuesto que se requiere para la reparación de un vehículo, posteriormente el trámite es llevado por la persona interesada a bodega para confirmar la existencia del pedido, luego le informa al Sr. Yovany Alvarado si el repuesto se encuentra en bodega o si se necesita realizar la compra fuera. Para realizar las requisiciones de bienes y servicios se realiza el mismo proceso.

La solicitud de reparación denominada orden de trabajo es solicitada por cualquier mecánico u otra persona del mismo departamento, en las cuales se describe que elemento se debe revisar y cuál es la falla que tiene el vehículo, las mismas deben ser aprobadas posteriormente. Cuando las reparaciones se realizan fuera quien solicita es el mecánico y se debe registrar cual es la mecánica a la cual va a dirigirse el vehículo.

Se investigó que en el departamento de bodega el registro de inventarios lo realizan en hojas de Excel y que dicha información no se la puede facilitar debido a la confidencialidad llevada en dicho departamento, y que el encargado de bodega no está interesado en utilizar un software para el registro del inventario de los repuestos de los vehículos.

Por lo cual se concluyó que el Sr. Yovany Alvarado no es el encargado de revisar el trabajo que se realiza en bodega, por lo tanto no puede llevar registro de inventarios.



Sr. Yovany Alvarado.

Entrevistado.

Anexo 10: Firma de requerimientos

REQUERIMIENTOS DEL SISTEMA

Requerimientos funcionales.

NÚMERO	DESCRIPCIÓN	PRIORIDAD
RF001	El sistema permitirá al administrador buscar crear, editar vehículos.	Alta/Esencial
RF002	El sistema permitirá al administrador cambiar el estado de las partes mecánicas del vehículo.	Alta/Esencial
RF003	El sistema permitirá al administrador editar el estado de ubicación del vehículo (Equipo bueno inactivo, Equipo trabajando normalmente, Equipo trabajando con fallas, equipo en reparación, equipo para baja o remate), el lugar en donde se encuentra y la fecha por cada estado.	Alta/Esencial
RF004	El sistema permitirá al administrador asignar un conductor a un vehículo.	Alta/Esencial
RF005	El sistema permitirá a la secretaria buscar, crear una solicitud de reparación y mantenimiento de un vehículo, con sus respectivos, ítems y el vehículo a reparar; las puede editar y dar de baja siempre y cuando la solicitud no esté aprobada por el administrador caso contrario solamente podrá visualizarla.	Alta/Esencial
RF006	El sistema permitirá al secretario asignar requisición de reparación en una solicitud de reparación y mantenimiento.	Alta/Esencial
RF007	El sistema permitirá al secretario buscar y crear una requisición para las reparaciones de un vehículo, con sus respectivos ítems, vehículo y el número de partida que le corresponde; las podrá editar y dar de baja siempre y cuando no estén aprobadas por el administrador caso contrario solo podrá visualizarlas.	Alta/Esencial
RF008	El sistema permitirá al secretario asignar solicitud de	Alta/Esencial

	reparación y mantenimiento en una requisición de reparación.	
RF009	El sistema permitirá al secretario asignar un producto a un ítem de requisición de reparación.	Alta/Esencial
RF010	El sistema permitirá al secretario buscar, crear, editar, y desactivar/activar una partida de contabilidad.	Alta/Esencial
RF011	El sistema permitirá al secretario buscar, crear, editar, y dar de baja un producto de reparación del vehículo.	Alta/Esencial
RF012	El sistema permitirá al administrador buscar y editar kardex.	Alta/Esencial
RF013	El sistema permitirá al administrador aprobar solicitud de reparación y mantenimiento a una kardex.	Alta/Esencial
RF014	El sistema permitirá al administrador aprobar requisición de reparación a una kardex.	Alta/Esencial
RF015	El sistema permitirá al administrador buscar, crear, editar y activar planes de mantenimiento.	Alta/Esencial
RF016	El sistema permitirá al administrador conocer por medio de alertas las revisiones necesarias que debe hacer al vehículo según el plan de mantenimiento asignado.	Alta/Esencial
RF017	El sistema permitirá al administrador imprimir reportes del informe mensual del estado de los vehículos, resumen del estado general de los vehículos.	Alta/Esencial
RF018	El sistema permitirá al administrador buscar, crear, editar el personal con sus diferentes tipos (administrador, secretario, conductor, mecánico, etc).	Alta/Esencial
RF019	El sistema permitirá al administrador crear, editar y desactivar las funcionalidades de acuerdo a requerimientos futuros.	Alta/Esencial
RF020	Iniciar sesión en el sistema mediante el ingreso del usuario y contraseña	Alta/Esencial

Requerimientos no funcionales.

NUMERO	DESCRIPCIÓN	PRIORIDAD
RNF001	El sistema permitirá dos tipos de usuarios, permitiendo un manejo más eficiente del sistema.	Alta/Esencial
RNF002	El tiempo máximo de respuesta del sistema en cada función que el usuario solicite será de máximo 8 segundos.	Alta/Esencial
RNF003	La información de los reportes deberá ser clara y precisa de acuerdo al formato que el usuario solicite.	Alta/Esencial
RNF004	El sistema deberá evitar el ingreso de información incorrecta por medio de mensajes que genere el sistema.	Alta/Esencial
RNF005	En caso de que el usuario olvide su usuario/contraseña podrá contactar al administrador quien le proveerá una nueva.	Alta/Esencial
RNF006	El sistema podrá ser usado en toda la jornada laboral del usuario.	Alta/Esencial
RNF007	La información ingresada en el sistema podrá ser visualizada, cambiada y usada para obtener reportes por el usuario registrado	Alta/Esencial
RNF008	El sistema permitirá que el administrador pueda gestionar funcionalidades de acuerdo a los requerimientos futuros.	Alta/Esencial
RNF009	Enterprise Edition 6 y con PostgreSQL como base de datos debido a su licencia GPL para evitar costos y problemas de licenciamiento.	Alta/Esencial



Administrador

Ing. Juan Carlos Vinueza



Secretario

Sr. Yovanni Alvarado

Anexo 11: Constancia de pruebas de usabilidad



**UNIVERSIDAD
NACIONAL
DE LOJA**



*Área de la Energía, las Industrias y los Recursos Naturales No
Renovables*

CARRERA DE INGENIERÍA EN SISTEMAS

De acuerdo a los resultados obtenidos en las pruebas de aceptación tanto el secretario y administrador que laboran en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas, se constató la aceptación con respecto al funcionamiento del sistema, con lo cual se indica que durante el desarrollo de las pruebas realizadas no existió ningún error tanto de ejecución como de validación de datos, por lo tanto se concluye que la aplicación es accesible y eficiente brindando facilidad en la navegación entre las paginas así como un tiempo aceptable de respuesta a los procesos solicitados en el software.

Ing. Juan Carlos Vinueza
Administrador

Sr. Yovanni Alvarado
Secretario

Yesica Norelly Saquicela Celi
Responsable del software

Carla Lisbeth Romero Córdova
Responsable del software

Anexo 12: Encuestas.



**UNIVERSIDAD
NACIONAL
DE LOJA**



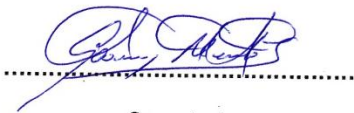
Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

Estimado Sr. Yovanni Alvarado, la finalidad de la presente encuesta es medir la iteración que tiene con el sistema y de esta manera comprobar la usabilidad y funcionalidad del software. Por lo cual, solicito me brinde un poco de su tiempo en responder las siguientes preguntas:

1. **¿Cómo considera usted el nivel de usabilidad del software?**
Fácil (✓)
Media ()
Difícil ()
2. **La iteración que tuvo con la interfaz del software. ¿Fue?**
Agradable (✓)
Poco agradable ()
Nada agradable ()
3. **¿Cómo considera usted el nivel de rendimiento del software?**
Rápido (✓)
Normal ()
Lento ()
4. **¿Las imágenes de los links principales indican claramente la tarea a realizar?**
Sí. (✓)
No. ()
5. **¿Los mensajes de validación e información son cortos y entendibles?**
Sí. (✓)

- No. ()
6. ¿La aplicación contiene colores coherentes y agradables en todas las páginas?
Sí. (✓)
No. ()
7. ¿Cree usted que los criterios de búsqueda de información son los adecuados y devuelven lo que espera?
Sí. (✓)
No. ()
8. ¿Los formatos de los reportes generados por el sistema cumple con sus expectativas?
Sí. (✓)
No. ()
9. ¿Considera usted que los botones de redirección a las páginas se ejecutan de forma rápida?
Sí. (✓)
No. ()
10. ¿La aplicación contiene enlaces de redirección a la página de inicio desde cualquier página del sistema?
Sí. (✓)
No. ()



Secretario

Sr. Yovanni Alvarado

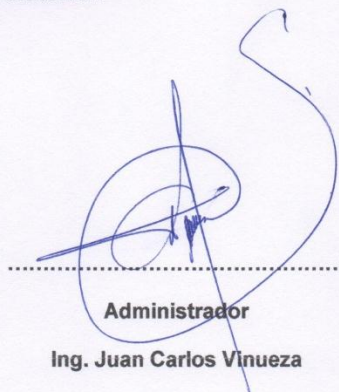


CARRERA DE INGENIERÍA EN SISTEMAS

Estimado Ing. Juan Carlos Vinueza, la finalidad de la presente encuesta es medir la iteración que tiene con el sistema y de esta manera comprobar la usabilidad y funcionalidad del software. Por lo cual, solicito me brinde un poco de su tiempo en responder las siguientes preguntas:


1. **¿Cómo considera usted el nivel de usabilidad del software?**
Fácil (x)
Media ()
Difícil ()
2. **La iteración que tuvo con la interfaz del software. ¿Fue?**
Agradable (x)
Poco agradable ()
Nada agradable ()
3. **¿Cómo considera usted el nivel de rendimiento del software?**
Rápido (x)
Normal ()
Lento ()
4. **¿Las imágenes de los links principales indican claramente la tarea a realizar?**
Sí. (x)
No. ()
5. **¿Los mensajes de validación e información son cortos y entendibles?**
Sí. (x)

- No. ()
6. ¿La aplicación contiene colores coherentes y agradables en todas las páginas?
Sí. (x)
No. ()
7. ¿Cree usted que los criterios de búsqueda de información son los adecuados y devuelven lo que espera?
Sí. (x)
No. ()
8. ¿Los formatos de los reportes generados por el sistema cumple con sus expectativas?
Sí. (x)
No. ()
9. ¿Considera usted que los botones de redirección a las páginas se ejecutan de forma rápida?
Sí. (x)
No. ()
10. ¿La aplicación contiene enlaces de redirección a la página de inicio desde cualquier página del sistema?
Sí. (x)
No. ()



Administrador
Ing. Juan Carlos Vinuesa

Anexo 13: Documento solicitud de reparación

 <p>SOLICITUD DE REPARACIÓN DE EQUIPO</p>		Clave		N°	
		Procedencia		Destino	
<p>Nº 006381</p>		Tipo de Equipo		Marca	
		Fecha		Registro	
Horómetro	Odómetro				
ELEMENTO A REVISAR		DESCRIPCIÓN DE LA FALLA			
Operador o Chofer:		Firma:			
Solicitado por:		Firma:			
Recibido por:		Firma:			
Fecha de Retiro:		De:		20	
OBSERVACIONES					
Serie 5401 - 6400					

Anexo 14: Documento de requisición

Nº		CANT.	UNIDAD MEDIDA	DESCRIPCION		
1		1	M.O	ALQUILER DE EQUIPO DE TOPOGRAFÍA XXXXXXXXXXXXXXXXXXXXXXXXXXXX		
OBSERVACION: NECESARIO PARA EL PROYECTO DE RECONSTRUCCIÓN Y MANTENIMIENTO DE LA VÍA CELICA-CRUZPAMBA-"Y" DEL MUERTO DEL MTOP-LOJA						
SOLICITADO Ing. Luis Ludeña SUPERVISOR FISCALIZADOR DEL PROYECTO			APROBADO Ing. Esner F. Bustamante R. DIRECTOR PROV. MTOP-LOJA (E)		FECHA DE ENTREGA	
Elaborado: YOVANNI ALVARADO			Compra Crédito Bodega Pago		x	

Anexo 15: Partidas de contabilidad

MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS DIRECCIÓN PROVINCIAL DEL MTOP-LOJA

CUADRO DE PARTIDAS EN LOS REQUERIMIENTOS DE GASTOS DE INVERSIÓN

UE	PG	SPG	PROY	ACT	ITEM	UBG	FTE	ORG	PTO	DESCRIPCION E-SIGEF
011	25	00	011	001	730202	1100	001	0000	0000	FLETES Y MANIOBRAS
011	25	00	011	001	730208	1100	001	0000	0000	SERVICIO DE VIGILANCIA
011	25	00	011	001	730404	1100	001	0000	0000	MAQUINARIA Y EQUIPOS (REPARACIONES)
011	25	00	011	001	730405	1100	001	0000	0000	VEHÍCULOS MANTENIMIENTO Y REPARAC.
011	25	00	011	001	730502	1100	001	0000	0000	ARRENDAMIENTO DE LOCALES, RESIDENCIAS, CASILLEROS JUDICIALES
011	25	00	011	001	730803	1100	001	0000	0000	COMBUSTIBLES Y LUBRICANTES
011	25	00	011	001	730811	1100	001	0000	0000	MATR. DE CONSTR. ELECT. PLOMR. Y SEÑALIZACIÓN
011	25	00	011	001	730813	1100	001	0000	0000	REPUESTOS Y ACCESORIOS
011	25	00	149	001	750105	1100	001	0000	0000	FISCALIZACIÓN DE LA AMPLIACIÓN DE LA CARRETERA EL EMPALME - CELICA - ALAMOR Y SU MANTENIMIENTO, INCLUYE LA CONSTRUCCIÓN DEL PUENTE LARAMINE

CUADRO DE PARTIDAS EN LOS REQUERIMIENTOS DE GASTOS DE GASTO CORRIENTE

UE	PG	SPG	PROY	ACT	ITEM	UBG	FTE	ORG	PTO	DESCRIPCION E-SIGEF
011	01	00	000	001	530101	1100	002	0000	0000	AGUA POTABLE
011	01	00	000	001	530104	1100	002	0000	0000	ENERGÍA ELÉCTRICA
011	01	00	000	001	530105	1100	002	0000	0000	TELECOMUNICACIONES
011	01	00	000	001	530106	1100	002	0000	0000	SERVICIOS DE CORREOS
011	01	00	000	001	530203	1100	002	0000	0000	ALMACENAMTO. RECARGA DE EXTINTORES
011	01	00	000	001	530204	1100	002	0000	0000	EDICIÓN, IMPRESIÓN, REPRODUCC. Y PUBLIC.
011	01	00	000	001	530228	1100	002	0000	0000	SERV. DISPOSIT. ELECTRÓNICOS PARA FIRMAS DIGITALES
011	01	00	000	001	530235	1100	002	0000	0000	SERVICIO DE ALIMENTACIÓN
011	01	00	000	001	530301	1100	002	0000	0000	PASAJES AL INTERIOR
		00	000	001	530303	1100	002	0000	0000	VIÁTICOS Y SUBSISTENCIAS AL INTERIOR
011	01	00	000	001	530603	1100	002	0000	0000	SERVICIO DE CAPACITACIÓN
011	01	00	000	001	530802	1100	002	0000	0000	VESTUARIO, LENCERÍA Y P. PROTECCIÓN
011	01	00	000	001	530804	1100	002	0000	0000	MATERIALES DE OFICINA
011	01	00	000	001	530805	1100	002	0000	0000	MATERIALES DE ASEO
011	01	00	000	001	530806	1100	002	0000	0000	HERRAMIENTAS
011	01	00	000	001	530807	1100	002	0000	0000	MATERIALES DE IMPRESIÓN, PUBLIC.
011	01	00	000	001	530809	1100	002	0000	0000	MEDICINAS Y PRODUCT. FARMACÉUTICOS
011	01	00	000	001	530811	1100	002	0000	0000	MATERIALES DE CONSTRUCC. ELECT. PLOMR.
011	01	00	000	001	530824	1100	002	0000	0000	MENAJE DE COCINA, DE HOGAR Y ACC. DESCARTABLES
011	01	00	000	001	570102	1100	002	0000	0000	TASAS GENERALES
		00	000	001	570201	1100	002	0000	0000	SEGUROS

Anexo 16: Entrevista dirigida a Sr. Danny Castillo sobre las partes mecánicas del vehículo.



UNIVERSIDAD
NACIONAL
DE LOJA



*Área de la Energía, las Industrias y los Recursos Naturales No
Renovables*

CARRERA DE INGENIERÍA EN SISTEMAS

RESULTADO DE ENTREVISTA

Martes 01 de mayo del 2014, a partir de 18h00 a 19h00 se realizó una entrevista al especialista en mecánica Dany Castillo, para conocer la importancia de cada parte mecánica de un vehículo. A continuación se lista cada una de las partes mecánicas con su debida importancia.

MOTOR	IMPORTANCIA
Potencia Motor	Muy importante
Sistema de lubricación	Muy importante
Sistema de refrigeración	Muy importante
Sistema de admisión y escape	Muy importante
Instrumentos de control	Muy importante

TRANSMICIÓN	IMPORTANCIA
Embrague	Muy importante
Caja de cambios	Muy importante
Diferenciales	Muy importante

SUSPENSIÓN	IMPORTANCIA
Muelles-Amortiguador	Muy importante
Eje delantero	Muy importante
Eje superior	Muy importante

DIRECCIÓN	IMPORTANCIA
Caja de dirección	Muy importante
Articulaciones y terminales	Muy importante
Pines bocines y rotulas	Muy importante
Bomba de la dirección	Muy importante

FRENOS	IMPORTANCIA
Cilindro principal	Muy importante
Cilindros secundarios	Muy importante
Forros de freno	Importante

SISTEMA ELÉCTRICO	IMPORTANCIA
Baterías	Muy importante
Arranque	Muy importante
Alterador	Muy importante

OTROS	IMPORTANCIA
Llantas	Importante
Carrocería	Importante


 Sr. Dany Castillo.
 Mecánico.

Anexo 17: Artículo sobre el trabajo de titulación

SOFTWARE PARA LA GESTIÓN DE INFORMACIÓN DEL ESTADO MECÁNICO DE LOS VEHÍCULOS DEL MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS

C. Romero, Y. Saquicela, L. Jácome.

Abstract— This research is the result of a dynamic paradigm that has been applied in the development of a web application for vehicle management, requisition, requests for repairs and maintenance. The evaluation of the mechanical condition of the vehicles was based on the assessment of the mechanical spares which allows users to obtain reports from mechanical condition and location of the vehicle components. The end result is a user friendly application that increases vehicle efficiency and user knowledge of city-owned vehicles. This application can be expanded and used in conjunction with other city, province or country facilities in Ecuador.

Keywords—Modelo dinámico, parametrizable, vehículos, IEEE 830, estado mecánico.

Resumen: *El presente artículo es el resultado de haber aplicado un modelo dinámico en el desarrollo de una Aplicación Web para la gestión de vehículos, requisiciones, solicitudes de reparación y mantenimiento, entre otros. Así también se realiza la evaluación del estado mecánico de los vehículos en base a la valoración*

del estado de sus partes mecánicas para obtener reportes del estado mecánico y de ubicación de los vehículos.

I. INTRODUCCIÓN

La información en la actualidad es considerada uno de los principales activos dentro de una empresa, razón por la cual toda organización busca excelencia en cuanto a disponibilidad, confidencialidad e integridad en sus procesos. Por ello se ha considerado realizar un análisis en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas (MTO) de la ciudad de Loja, en el cual se ha encontrado inconsistencia y redundancia de datos por la utilización de paquetes de ofimática (Excel, Word, entre otros) ya que estos programas cubren necesidades generales mas no específicas. De acuerdo al análisis realizado se ha determinado el siguiente problema general de investigación: “¿La implementación de un software estándar parametrizable, permitirá minimizar la desorganización de la información del estado mecánico de los vehículos?”.

Por tal motivo este artículo es para dar a conocer como se desarrolló el sistema con el fin de apoyar

C. Romero, Universidad Nacional de Loja, Loja, Ecuador, chromeroc@unl.edu.ec

Y. Saquicela, Universidad Nacional de Loja, Loja, Ecuador, ynsaquicelac@unl.edu.ec

a los procesos de gestión de información de los vehículos, requisiciones, solicitudes de reparación y mantenimiento entre otros módulos que son utilizados diariamente.

La organización del trabajo es la siguiente: en la Sección II se explican las características de las herramientas utilizadas. La Sección III se describe los materiales y métodos. La Sección IV muestra detalles de los resultados. Finalmente, se pueden encontrar las conclusiones en el Apartado V.

II. CARACTERÍSTICAS DE LAS HERRAMIENTAS UTILIZADAS

Entre las herramientas utilizadas para el desarrollo del presente proyecto tenemos:

- Enterprise Architect (AE): para el análisis del lenguaje de modelado unificado (UML) [1].
- IDE Netbeans: ayuda para la codificación de la aplicación. [2]
- Base de datos PostgreSQL: almacenamiento y consultas de la información ingresada desde la aplicación [3].
- Java Database Connectivity (JDBC): puente de conexión con la base de datos [3].
- Servidor de aplicaciones JBoss: espera y responde peticiones de la aplicación [4]-[5].
- Patrón de diseño MVC: ayuda a separar el problema en Modelo, Vista y Controlador para una mejor solución [6]-[7].
- Java Server Faces (JSF): framework de aplicaciones para manejar las peticiones del cliente y gestionar su ciclo de vida [7].
- Lenguaje de expresiones (EL): para realizar validaciones y según ciertas condiciones renderizar los componentes de la vista [8].
- JBoss Seam: es un framework utilizado para la seguridad de aplicaciones Java EE [9]-[10].
- Maven: Framework de gestión de proyectos de software ayuda a la creación de proyecto Java con el uso de arquetipos predefinidos en el archivo xml (Pom.xml), en el cual se definen las dependencias de las librerías necesarias, generando plugins relacionados

con el jar necesario para un proyecto [11]-[12].

- Bootstrap: permite que la pantalla se adapte automáticamente al tamaño de un ordenador o de cualquier dispositivo móvil, esto se denomina diseño adaptativo o Responsive Design [13].
- JMeter: permite realizar tests de comportamiento funcional y medir el rendimiento de la Aplicación Web [14].

III. MATERIALES Y MÉTODOS.

Para el desarrollo del proyecto se emplearon los métodos de investigación como la observación activa que permitió integrarse al entorno donde se desarrollan las actividades de los usuarios, también el estudio de casos con el cual se logró obtener información necesaria para dar solución al problema a resolver. Entre las técnicas aplicadas para la recolección de información tenemos la entrevista y encuesta que sirvieron para conocer de forma exhaustiva la situación actual dentro del departamento vial del MTOP, de la misma forma se empleó la técnica de investigación bibliográfica para recolectar información necesaria que permita sustentar teóricamente el trabajo de titulación.

Tomando en cuenta que para todo proceso de desarrollo de software es importante aplicar una metodología que ayude a dar homogeneidad y calidad a los resultados, se ha creído conveniente utilizar la metodología Iconix, la cual ha proporcionado métodos para llevar un control del ciclo de vida del presente proyecto. A continuación se menciona como se ha desarrollado cada una de las fases de esta metodología:

1. Análisis de requerimientos: se aplicó el estándar IEEE 830 para una mejor especificación de requerimientos además se construyó el modelo del dominio, modelo de casos de uso, prototipo de interfaz de usuario.
2. Análisis y diseño preliminar: se construyeron los diagramas de casos de uso y robustez además de la descripción de los cursos normales y alternos de los casos de uso.

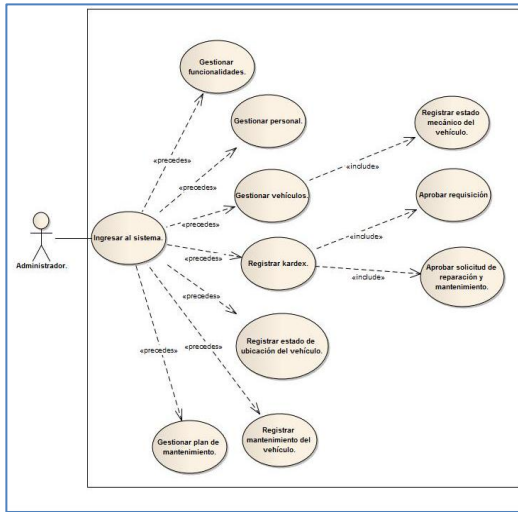


Fig. 4 Diagrama de casos de uso del administrador.

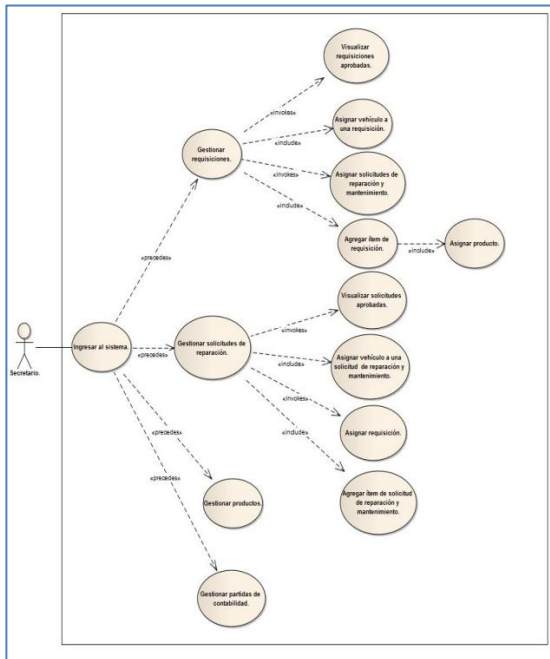


Fig. 5 Diagrama de casos de uso del secretario.

La implementación de la aplicación se realizó siguiendo la lógica que se muestra en el diagrama de componentes. Ver Fig. 6 [17].

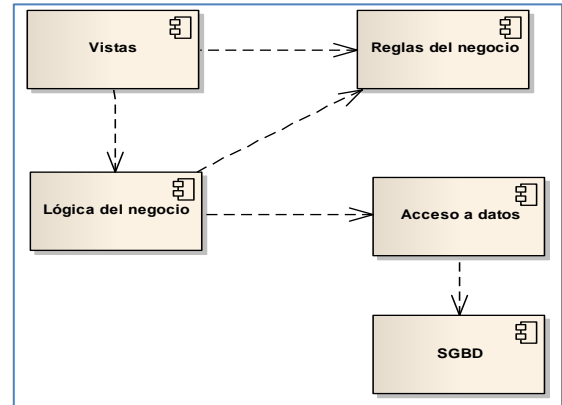


Fig. 6 Diagrama de componentes.

La Arquitectura de la Aplicación Web desarrollada se muestra en la Fig. 7. En la cual se indica que los usuarios pueden ingresar desde un computador o dispositivo conectado a la misma red, desde un navegador Web puede acceder a la aplicación.

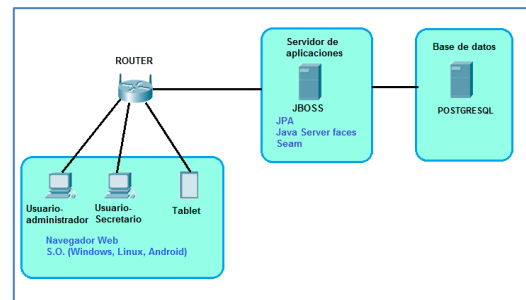


Fig. 7 Arquitectura de la aplicación.

Las restricciones de la tecnología a utilizar se indican en la tabla I.

**TABLA I.
RESTRICCIONES DE LA TECNOLOGÍA DE LA APLICACIÓN**

	Servidor aplicación	Servidor BD	Cliente
Software	JBoss v2	PostgreSQL v9.1	Navegador Web (Windows-Linux-Android)
Hardware	Dell (Optiplex 780)	Dell (Optiplex 780)	Dell (Optiplex 780)

Para la implementación de la Aplicación Web se utilizó como modelo el software empresarial para gestión de proyectos, que se basa en reglas de negocio con configuraciones parametrizables acoplándose a cualquier módulo de una empresa [16], el cual sirvió de base para resolver el problema planteado y de esta forma adaptar la Aplicación Web con dos niveles de personalización permitiendo al usuario agregar en un futuro nuevos cambios en el sistema y así evitar grandes inversiones que demandan la implementación de nuevos requerimientos. A continuación se procede a indicar la creación de entidades y propiedades mediante la interfaz.

En la Fig. 8 [17] se indica la lista de propiedades de la **Entidad** Vehículo que se crearon inicialmente, a partir de esta lista se puede crear una nueva **Propiedad**, **Entidad** y desactivar según se requiera.

Nombre	Etiqueta de vista	Tipo de dato	Valor	Obligatoria	Posición	Crear entidad	Editar	Desactivar
Chasis	Chasis	Estructura		<input checked="" type="checkbox"/>	1	<input type="button" value="C"/>	<input type="button" value="E"/>	<input type="button" value="D"/>
motor2	motor2	Estructura		<input type="checkbox"/>	1	<input type="button" value="C"/>	<input type="button" value="E"/>	<input type="button" value="D"/>
Motor	Motor	Estructura		<input checked="" type="checkbox"/>	2	<input type="button" value="C"/>	<input type="button" value="E"/>	<input type="button" value="D"/>
Historial	Historial	Estructura		<input checked="" type="checkbox"/>	3	<input type="button" value="C"/>	<input type="button" value="E"/>	<input type="button" value="D"/>
SistemaElctrico	Sistema Elctrico	Estructura		<input checked="" type="checkbox"/>	4	<input type="button" value="C"/>	<input type="button" value="E"/>	<input type="button" value="D"/>

Fig. 8. Lista de propiedades de la entidad Vehículo.

Los tipos de datos que se ha considerado para crear una **propiedad** son: **estructura (Structure)**, **fecha (Date)**, **texto (String)**, **real (Double)**, **entero (Integer)**, **entero mayor (Long)**, **Booelano (Bolean)**, **Lista (String [])**, **área de texto (MultiLineString)**, y el tipo "EstadoParteMecánica" se mostrará cuando la **Entidad** a la que se le está agregando la **Propiedad** sea una **Propiedad** de la clase Vehículo.

Para evitar conflictos se ha considerado fijar el nombre del grupo de la **Propiedad** con el mismo nombre de la **Entidad** a la que se está agregando dicha **Propiedad**. En la Fig. 9 [17] se indica lo parámetros para crear una **Propiedad**.

Fig. 9. Crear propiedad.

La **Propiedad** que fue creada como tipo "Estructura" se la podrá crear como **Entidad**, para ello se consideró fijar como nombre de la **Entidad** y etiqueta de vista el nombre y etiqueta respectivamente de la **Propiedad** y de esta forma evitar cualquier inconveniente. En la Fig. 10 [17] se indica los parámetros para crear de una **Entidad**.

Fig. 10. Crear Entidad.

En la Fig. 11 [17] se muestra la **Propiedad** creada anteriormente como **Entidad** y sus **propiedades** dentro de la clase Vehículo.

Fig. 11. Propiedades dentro de la clase Vehículo

Para realizar la evaluación general de un vehículo en base al estado de sus partes mecánicas se lo ha realizado según la importancia que tiene cada parte del vehículo y valorizando en un estado BUENO y MALO a cada una.

Para ello se consideró que cuando se cree una **Propiedad** dentro de una **Entidad** que sea **Propiedad** de la clase Vehículo se mostrar la opción tipo “EstadoParteMecanica” en la cual se debe asignar un porcentaje inicial de 100% muy importante y 80% importante, como se indica en la Fig. 12 [17].

Fig. 12 Crear propiedades tipo “EstadoParteMecanica”.

Para realizar la evaluación en la Fig. 13 [17] se muestra todas las **Propiedad** que son de tipo “EstadoParteMecanica” de la clase Vehículo en el cual se pueda realizar la evaluación de cada parte mecánica y de esta forma obtener la evaluación general del vehículo.

Fig. 13 Propiedades del Vehículo

C. Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la aplicación Web.

Para realizar las pruebas de funcionalidad y rendimiento del sistema se ha creído conveniente hacer uso de la herramienta JMeter, la cual permitió simular la concurrencia de un gran número de usuarios haciendo peticiones al servidor y mediante esto dar a conocer los tiempos exactos de respuesta de la Aplicación Web.

Como se puede observar en la tabla II, los tiempos de respuesta son aceptables; con 50 usuarios 2,712 segundos, con 100 usuarios 1,686 segundos y con 150 usuarios 1,950 segundos; al tratarse de una aplicación que consta de dos tipos de usuarios, los resultados son los esperados.

TABLA II.
RESUMEN DE LOS DATOS OBTENIDOS EN LAS TRES PRUEBAS.

#Usuarios	#Muestras (tareas)	Rendimiento (peticiones)	Tiempo medio de Respuesta	Dispersión
50	650	518,741/min	2712	3963
100	1300	625,923/min	1686	3039
150	1950	652,021/min	1950	3854

Además se realizaron las pruebas de usabilidad las cuales permitieron conocer el impacto que el sistema tuvo frente al usuario, de esta forma se evaluó cada módulo con la finalidad de obtener la respuesta de la Aplicación Web, así como mensajes ya sea de información o de validación, verificando si cada requerimiento se ha cumplido o no.

V. CONCLUSIONES.

La utilización del estándar IEEE 830 ayudó a obtener un documento de requerimientos con el

fin de tener un contrato con el cliente, y así obtener un sistema eficiente acorde a las necesidades del mismo, además es muy importante obtener la documentación suficiente durante el proceso de desarrollo de software, para que en un futuro la misma sirva de guía y se pueda ampliar fácilmente según nuevos requerimientos. De acuerdo al estudio realizado del modelo dinámico se logró implementar un sistema que ayuda al usuario a crear nuevos cambios futuros, de una forma diferente a la parametrización que se realiza comúnmente en el software rígido. En un futuro este modelo puede ampliarse utilizando más niveles de personalización, para la aplicación se ha creído conveniente utilizar dos niveles. Mediante la utilización de pruebas se verificó que los procesos de registro de información en la Aplicación Web se realicen de forma rápida y confiable, permitiendo tener una mayor organización de las actividades que se realizan dentro del departamento de mantenimiento vial del MTOP de Loja. Este sistema al estar implementado con el modelo dinámico puede ser utilizado como base para nuevos proyectos de gestión de información en diferentes instituciones públicas del Ecuador. La licencia de esta Aplicación Web es tipo GPL (Licencia Pública General) es decir que este software pueda ser utilizado de forma libre, redistribuirlo o modificarlo para futuros proyectos ver <http://www.gnu.org/licenses/gpl.html>. Para conocimiento del software realizado puede visitar el siguiente link: <https://www.youtube.com/watch?v=z5pgkIAM0B0>.

REFERENCIAS

- [1]Página oficial de Enterprise, “Enterprise architect”, disponible en: <http://www.sparxsystems.com/products/ea/9/>.
 [2]Página oficial de Netbeans, “Netbeans”, disponible en: <https://netbeans.org>.
 [3]Página oficial de Oracle “Java SE Technologies - Base de Datos”, disponible: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>.
 [4]D. Fernando I. López, “Benchmark para el uso de tecnologías relacionadas a servidores de aplicaciones glassfish y jboss”, Escuela politécnica nacional, Quito del 2009.

- [5]Página oficial de Jboss, “Servidor Jboss”, disponible en: <http://jbossas.jboss.org>.
 [6]Nicolás Tedeschi, “¿Qué es un Patrón de Diseño?”, reporte técnico, publicado el 25 de mayo del 2011.
 [7]M.A. Álvarez (2014, Enero 02), “¿Qué es MVC?”, reporte técnico, publicado el 02 de enero del 2014.
 [8]B. Gutierrez, “Introducción al Lenguaje de expresiones”, reporte técnico, publicado el 2014.
 [9]Página oficial de Seam Security, “Framework Seam”, Disponible en: <http://www.seamframework.org/>.
 [10]Picketlink, “Integración de Picketlink”, disponible en: <http://docs.jboss.org/picketlink/2/latest/reference/html/>.
 [11]M. Grau, “Introducción a maven”, reporte técnico, publicado el 26 de Julio del 2011
 [12]Página oficial de Maven, “Apache Maven Project”, disponible en: <http://maven.apache.org/download.cgi>
 [13]Página oficial de bootstrap, “Bootstrap”, disponible en: <http://getbootstrap.com/>
 [14]J. Díaz, C. Tzancoff, A. Rodríguez, V. Soria, “Usando Jmeter para pruebas de rendimiento”, Universidad Nacional de la Plata
 [15]UTPL, “¿Qué es IEEE?”, reporte técnico, publicado el 16 de abril del 2011.
 [16]J. Granda, “Software empresarial para gestión de proyectos”, Disponible en: <https://github.com/jlgranda/gluebrms>
 [17]C. Romero, Y. Saquicela, "Diagramas del software del estado mecánico de los vehículos", disponible en: <http://diagramasestmecvehiculos.blogspot.com/>



Carla Romero, egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja periodo 2009-2014, conocedora en análisis y desarrollo de software con Java, mantenimiento y reparación de computadoras. Áreas de interés: redes, desarrollo de software. Provincia de Loja, Ciudad Loja, Ecuador, 2014.



Yesica Saquicela, egresada de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja periodo 2009-2014, conocedora en análisis y desarrollo de software con Java. Áreas de interés desarrollo e Ingeniería de Software. Provincia de Loja, Ciudad Loja, Ecuador, 2014.



Luis Jácome, Ingeniero en Sistemas Informáticos y Computación en la Universidad Técnica Particular de Loja, Magíster en Telemática en la Universidad de Cuenca, estudiante de Maestría en Ingeniería Computacional y Sistemas Inteligentes de la Universidad del País Vasco, Área de interés: Desarrollo de Software e Inteligencia Artificial. Provincia de Loja, Ciudad Loja, Ecuador, 2014.

Anexo 18: Certificado de traducción del resumen de memoria y artículo.

LOJA
CIUDAD.

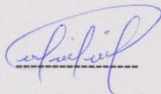
Loja, 27 de Octubre del 2014

CERTIFICADO

Yo **VÍCTOR EFRÉN LIMA POMA** Licenciado en Ciencias De la Educación Mención Idioma Inglés con cédula nro. **1104906290**, certifico que se ha realizado la traducción del resumen del Trabajo de Titulación cuya problemática versa sobre el tema: "**Software para la gestión de información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas.**", realizado por las egresadas **CARLA LISBETH ROMERO CÓRDOVA, YESICA NORELLY SAQUICELA CELI**, previa a la obtención del título de Ingeniero en Sistemas.

Es todo cuanto puedo informar sobre el particular y autorizo a las interesadas hacer uso del presente certificado de la manera que estime conveniente.

Atentamente,



Víctor Efrén Lima Poma

Anexo 19: Licencia



Software para la gestión de información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas por Carla Lisbeth Romero Córdova, Yesica Norelly Saquicela Celi se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](#).

Anexo 20: Anteproyecto del trabajo de titulación.



**UNIVERSIDAD
NACIONAL
DE LOJA**

TT-CIS-XA-008



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Software para la gestión de información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas”

ANTEPROYECTO DE TRABAJO DE TITULACIÓN

Autores:

- Carla Lisbeth Romero Córdova
- Yesica Norelly Saquicela Celi

Asesor:

- Ing. Luis Antonio Chamba Eras, Mg. Sc.

LOJA-ECUADOR

2014

A. Tema

“Software para la gestión de la información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas”

B. Problemática

1. Situación problemática

En la actualidad uno de los activos más importante de toda entidad es la información, considerada base fundamental para obtener un alto nivel de competitividad y posibilidades de desarrollo, inicialmente las empresas tanto públicas como privadas realizaban la gestión de la información de forma manual produciendo desorganización en las actividades, hoy en día con el uso de herramientas tecnológicas los procesos de las organizaciones se consideran más eficientes [1]-[2].

A nivel de Ecuador e Internacional, la innovación de la tecnología en las empresas ha contribuido para mantener una visión global de la situación actual de sus procesos. Por esta razón la utilización de un software para la gestión de la información permite que las empresas realicen sus procesos de una forma rápida y sencilla [3]-[4]. Entre algunos casos de éxito relacionados se puede mencionar la implementación de un Software para la gestión de la información en la empresa Kauferstein, el cual permitió estar al día en sus actividades empresariales [5].

En la mayoría de las empresas los software más comunes que se utilizan para la gestión de información, son desarrollados a medida, es decir, diseñados de acuerdo a las necesidades específicas de la empresa, por esta razón en algunos casos se convierte en una herramienta poco eficiente, evitando ser adaptable a nuevos requerimientos de la empresa [6]-[7].

Como una nueva alternativa al desarrollo de software a medida, se encuentra el software estándar, que se basa en cubrir necesidades que surjan en entidades de la misma naturaleza [8]. Entre los tipos de software estándar tenemos los rígidos en los cuales la organización debe adaptarse totalmente al estándar desarrollado, mientras que los productos software estándar parametrizables permiten ajustarse a nuevos cambios futuros requeridos por la empresa [9]. Un ejemplo de un software estándar parametrizable es el que se desarrolló en la ciudad de México para la emisión de facturas electrónicas [10].

En base a la búsqueda de información bibliográfica no se ha podido encontrar un software dentro de la ciudad de Loja que permita ajustar cambios futuros requeridos por una empresa, es decir que el software se vaya adaptando a las necesidades de la entidad y que el mismo tengan la facilidad de gestionar (crear, modificar, eliminar) funcionalidades de acuerdo a sus requerimientos.

Por tal motivo se ha creído conveniente desarrollar un software estándar parametrizable que permita adaptar nuevos requerimientos solicitados en el Departamento de Mantenimiento Vial del Ministerio de Transporte y Obras Públicas de la ciudad de Loja, y de esta manera brindar la posibilidad al administrador gestionar las funcionalidades de acuerdo a sus necesidades. De acuerdo a una encuesta estructura (ver anexo 20.1, 20.2) al encargado de llevar la información del estado mecánico de los vehículos, el cual manifestó que existen más Departamentos de Mantenimiento Vial en otras ciudades que realizan esta gestión de información de manera manual o utilizando paquetes de Microsoft office (Excel, Word, entre otros) para la administración de los datos, generando que la información no sea procesada de manera eficiente, de esta forma se ha podido detectar la siguiente situación problemática:

- Inconsistencia de los datos y pérdida de información.
- Información duplicada en los registros del estado de los vehículos debido a la redundancia de datos.
- El formato para registrar los datos de los vehículos no se encuentra estandarizado.
- Los registros de los datos son obsoletos debido a que la información no se encuentra actualizada.

5. Problema de investigación

¿La implementación de un software estándar parametrizable, permitirá minimizar la desorganización de la información del estado mecánico de los vehículos?

C. Justificación

El presente Trabajo de Titulación (TT), ayudará reforzar y aplicar los conocimientos adquiridos en el transcurso de la formación como Ingeniero en Sistemas, para ello se desarrollará un software de gestión de la información de los vehículos, que permita crear, modificar y eliminar funcionalidades de acuerdo a sus necesidades futuras que surjan en del Departamento de Mantenimiento Vial del Ministerio de Transporte y Obras Públicas en la ciudad de Loja. Por ello se ha creído conveniente desarrollar este tipo de software

estándar parametrizable ya que en nuestra localidad no existen productos de software con características adaptables a las necesidades de la empresa.

En el desarrollo de este trabajo se utilizará diferentes tecnologías, estándares y herramientas necesarias que permitan la construcción de una aplicación Web adaptable a nuevos requerimientos del usuario. Además al gestionar la información de forma electrónica permitirá minimizar gastos innecesarios de recursos materiales derivados del medio ambiente como papel, entre otros. Los gastos que demande este trabajo serán asumidos por las investigadoras, ya que se trata de una investigación académica. De esta manera se justifica el Trabajo de Titulación en el ámbito académico, técnico-tecnológico, ambiental y económico.

D. Objetivos

1. Objetivo General

Desarrollar un software estándar parametrizable que permita adaptarse a requerimientos futuros que surjan en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas para la gestión de información de los vehículos.

2. Objetivos Específicos

- Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar.
- Implementar la metodología seleccionada que permita guiar la gestión del proyecto.
- Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la aplicación Web.

E. Alcance

El presente trabajo de titulación se centra especialmente en el desarrollo de un software parametrizable que permita adaptar nuevos requerimientos solicitados por el Departamento de Mantenimiento Vial del Ministerio de Transporte y Obras Públicas para la gestión de la información de los vehículos y de esta forma tengan la facilidad de gestionar (crear, modificar, eliminar) funcionalidades de acuerdo a sus necesidades futuras. Además se hará uso de la tecnología Responsive web design el cual permitirá que la estructura sea cambiada de acuerdo a la resolución de la pantalla del dispositivo (Computadora, Tablet) desde el que se esté visualizando y de esta manera la aplicación pueda ser utilizada sin ninguna restricción en diferentes entornos de acceso [11].

El software permitirá al administrador del sistema conocer por medio de alertas las fechas tentativas en que los vehículos deberán salir de talleres según la información que proporcione el mecánico, de esta manera el administrador podrá registrar la fecha exacta en que salieron los vehículos.

También el software permitirá la administración de usuarios, gestión (crear, buscar, modificar, eliminar) de vehículo y además asignar conductores y mecánicos a cada vehículo. La secretaria podrá realizar solicitudes de reparación de vehículos, requeridas por el conductor y requisiciones presentadas por el mecánico para corrección de las fallas encontradas en los vehículos.

El administrador podrá llevar una Kardex para el registro de las solicitudes y requisiciones emitidas por la secretaria, además en una ficha técnica registrará los estados (bueno, malo, regular) de las partes mecánicas del vehículo y mediante la utilización de un algoritmo que le ayude al administrador a determinar el estado (Bueno o Malo) general del vehículo en base a cada uno de los estados de las partes mecánicas de los vehículos (ver anexo 20.5).

El software permitirá generar reportes mensuales del control de estados (en reparación, para baja o remate, trabajando con fallas, equipo bueno inactivo, trabajando normalmente) del vehículos (ver anexo 20.6) y su ubicación, así también resumen del estado general por vehículo y un resumen de estado de la maquinaria por número de días, para ser enviados por correo electrónico a las personas que soliciten dicha información.

Como un aporte al software se ha creído conveniente implementar un módulo para la gestión de planes de mantenimiento de los vehículos.

Para ello se ha considerado conveniente un tiempo de 400 horas en el cual se ha planteado las siguientes fases y actividades estimadas:

1. Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar.

- Búsqueda de información para el desarrollo del software empresarial.
- Comparativas de las metodologías de desarrollo.
- Selección de la metodología a utilizar para el desarrollo de la aplicación Web.

2. Implementar la metodología seleccionada que permita guiar la gestión del proyecto.

- Analizar los procesos actuales sobre el registro de la información de los vehículos.

- Diseñar un modelo datos que permita ajustar la aplicación Web a los requerimientos del usuario.
 - Implementar un software que permita gestionar la información del estado mecánico de los vehículos.
3. Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la aplicación Web.
- Analizar los métodos y herramientas necesarias para realizar pruebas de rendimiento y usabilidad.
 - Validar el funcionamiento del software en base a los resultados obtenidos en las pruebas realizadas.

F. Metodología

Para el presente Trabajo de Titulación (TT) los métodos y técnicas que se utilizará son las siguientes:

Métodos

Observación activa: Este método se utilizará con el fin de obtener una mayor comprensión sobre los procesos que se realiza en el departamento de mantenimiento vial [12].

Estudio de casos: Se utilizará este método para llevar a cabo entrevistas, encuestas, observaciones entre otras, que permita obtener mayor información [12].

Técnicas

Entrevista: Esta técnica será utilizada para recopilar la información necesaria de la situación actual del departamento de mantenimiento vial.

Investigación bibliográfica: Con esta fuente se sustentará la investigación, ya que las consultas serán realizadas de fuentes confiables como: artículos, libros, tesis de grado, casos de éxito entre otras.

G. Cronograma

El presente Trabajo de Titulación (TT) se desarrollará en el Ministerios de Transportes y Obras Públicas de la ciudad de Loja, en el Departamento de Mantenimiento Vial, para ello se utilizará un tiempo de 400 horas en el cual se ha planteado las siguientes fases y actividades estimadas:(ver anexo 20.4)

R. Presupuesto y Financiamiento

1.1. Talento Humano

TABLA XLI.
TALENTO HUMANOS ESTIMADO A DISPONER PARA EL DESARROLLO DEL
TT.

Rol	Número de horas	Costo de horas	Costo total
Analista- Investigador- Diseñador- Desarrollador	200	\$ 10.00	\$ 2000.00
Analista- Investigador- Diseñador- Desarrollador	200	\$ 10.00	\$ 2000.00
Subtotal			\$ 4000.00

1.2. Recursos técnicos

TABLA XLII.
RECURSOSTÉCNICOS ESTIMADOS A DISPONER PARA EL DESARROLLO DEL
TT

Descripción	Cantidad	Valor unitario	Valor Total
Hardware			
Computadora	4	\$ 250.00	\$ 1000.00
Impresora	4	\$ 60.00	\$ 120.00
Disco USB	1	\$ 50.00	\$ 50.00
Software			
Netbeans		Gratis	\$ 0.00
Latex		Gratis	\$ 0.00
Subtotal			\$ 1170.00

1.3. Recursos materiales

TABLA XLIII.
RECURSOS MATERIALES ESTIMADOS A DISPONER PARA EL DESARROLLO
DEL TT

Descripción	Cantidad	Valor unitario	Valor Total
Cartucho de tinta	4	\$ 25.00	\$ 100.00
Resma de papel	2	\$ 4.00	\$ 8.00
Transporte	2	\$ 0.25	\$ 200.00
Internet	300 h	\$ 0.80	\$ 240.00
Copias	200	\$ 0.02	\$ 4.00
Varios		\$ 400.00	\$ 400.00
Subtotal			\$ 952.00

1.4. Total de Recursos

TABLA XLIV.
PRESUPUESTO GENERAL ESTIMADO A DISPONER PARA EL DESARROLLO DEL
TT

Recursos	Subtotal
Recursos humanos	\$ 4000.00
Recursos materiales	\$ 1170.00
Recursos técnicos y tecnológicos	\$ 925.00
Total	\$ 6652.00

2. Financiamiento

El presupuesto estimado para el desarrollo del Trabajo de Titulación, será asumido por las investigadoras, por tratarse de una investigación de carácter académico.

S. Bibliografía

- [1] J. Medina, P. Aguilar, "Administración y calidad de la información de los sistemas de información contable de las PYMES", presentado en Abril del 2013.
- [2] D. Pérez and M. Dressler, "Tecnologías de la información para la gestión del conocimiento," vol. 3, pp. 31–59, 2007.
- [3] J. Martínez, J. Majó, M. Casadesús, "El uso de las tecnologías de la información en el sector hotelero" VI Congreso "Turismo y Tecnologías de la Información y las Comunicaciones", 2006.
- [4] J. Mota, L. Zavaleta, L. Luque, "Procesos de innovación y competencias de los recursos humanos en la industria del software en Argentina Revista Iberoamericana de Ciencia, Tecnología y Sociedad. 24.8 (Sept. 2013): p145.," vol. 8, 2013.
- [5] F. Kaufenstein, "Un cambio a mejor", Kaufenstein, reporte técnico, Alemania-Haiger, Julio 2010.
- [6] J. Blanco, A. Rodríguez, "Revisión, verificación y validación en un proceso de software.", publicado en Marzo del 2011.
- [7] C. Jiménez, J. Erasmo, "Generación ágil de software orientado a objetos a gestión de información.", Colombia, Junio 2010.
- [8] U. Ruiz, "Fundamentos de Software", reporte técnico, publicado el 23 de noviembre del 2012.
- [9] J. Cestero, "Software a medida o estándar en la PYME: estándar sin duda", reporte técnico, publicado el 02 de febrero del 2009.
- [10] Alejandro H. Angulo, Agustín V. Moya, Raúl M. Anwandter, "Herramienta de software parametrizable, para la emisión de facturas electrónicas según la legislación de México.", Universidad de Chile, 2012.
- [11] E. Marcotte. , "Diseño Web Responsivo", reporte técnico, publicado el 25 de Mayo del 2010.
- [12] C. Dawson, "El proyecto fin de carrera en ingeniería informática: una guía para el estudiante", Prentice Hallducación, 2002, pp. 21-41.

T. Anexos

Anexo 20.1: Entrevista



UNIVERSIDAD NACIONAL DE LOJA
Área de la Energía, las Industrias y los Recursos
Naturales No Renovables



CARRERA DE INGENIERÍA EN SISTEMAS

Estimado Ing. Juan Carlos Vinueza, la finalidad de la presente encuesta es recopilar la información necesaria para la realizar un software, con la finalidad de gestionar la información del estado mecánico de los vehículos de una manera más rápida y confiable. De esta manera, solicito me brinde un poco de su tiempo en responder las siguientes preguntas:

1. ¿En qué consiste el trabajo que realiza para llevar el registro del estado mecánico de los vehículos?

.....
.....

2. ¿Considera Ud. que el trabajo que realiza para llevar el control del estado mecánico de los vehículos es el apropiado?

.....
.....

3. Cree que los procesos que realiza en su trabajo son muy tediosos.

.....
.....

4. Que le gustaría obtener al optimizar los procesos de su trabajo.

.....
.....

5. Cuantas personas se encargan de manipular esta información.

.....
.....

6. ¿Qué rol cumplen las personas que manejan esta información?

.....
.....

7. ¿Cree que es conveniente que el software permita gestionar (crear, modificar, agregar) los datos de manera dinámica?

.....
.....

Anexo 20.2: Resultado de la entrevista



**UNIVERSIDAD
NACIONAL
DE LOJA**



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS


ENTREVISTA

Entrevistado: Ing. Juan Carlos Vinueza.

Resultados de la entrevista:

En el Ministerio de Transporte y Obras Públicas dentro del departamento de Mantenimiento vial, realizan el registro del estado mecánico de los vehículos se de la siguiente manera: la secretaria se encarga de registrar, choferes , conductores y solicitudes de orden de reparación de los vehículos, esta solicitud luego es entregada a mi persona y a partir de esta orden se registra los cambio del estado de las partes mecánicas del vehículo y cada falla es registrada en un Kardex, en base a estos que tienen los vehículos diariamente se da un estado general del vehículo como por ejemplo, si esta en reparación, a mantenimiento, trabajando normalmente, etc. Estos registros son presentados en un reporte que se realiza cada mes del estado actual del vehículo para ser luego enviado a la matriz que se encuentra en Quito.

Según el criterio del ing. Juan Carlos Vinueza, manifestó que existe más departamento del mantenimiento vial del control del estado mecánico de los vehículos, que realizan el mismo proceso para el registro de la información.



Ing. Juan Carlos Vinueza.
Firma

Anexo 20.3: Carta de compromiso



Ing. Hernán Torres Carrión

Coordinador de la Carrera de Ingeniería en Sistemas

Universidad Nacional de Loja

Ciudad.

Yo, Juan Carlos Vinueza, administrador del control de los vehículos en el departamento de mantenimiento vial del Ministerio de Transporte y Obras Públicas, con número de cedula 1103187769, acepto y me comprometo a ofrecer la información necesaria para que las estudiantes de la carrera de Ingeniería en Sistemas: Carla Lisbeth Romero Córdova y Jesica Norelly Saquicela Celi, del Decimo módulo puedan realizar su Trabajo de Titulación, durante el periodo académico marzo-julio del 2014.

De acuerdo a lo anterior, autorizo a las interesadas hacer uso del presente certificado de la manera que estimen conveniente.

Atentamente

Ing. Juan Carlos Vinueza.

Anexo 20.4: Cronograma estimado de actividades

Nombre de tarea	Comienzo	Fin	Horas
Fase1: Analizar y seleccionar la metodología para el desarrollo de software según el tipo de proyecto a implementar	lun 28/04/14	vie 09/05/14	50 Horas
Búsqueda de información para el desarrollo del software empresarial.			15 horas
Comparativas de las metodologías de desarrollo.			15 Horas
Selección de la metodología a utilizar para el desarrollo de la aplicación Web			5 Horas
Elaboración de la documentación			15 Horas
Fase2: Implementar la metodología seleccionada que permita guiar la gestión del proyecto.	lun 12/05/14	jue 31/07/14	260 Horas
Analizar los procesos en el registro de la información de los vehículos.			25 Horas
Diseñar un modelo datos que permita ajustar la aplicación Web a los requerimientos del usuario.			25 Horas
Implementar un software que permita gestionar la información del estado mecánico de los vehículos.			180 Horas
Elaboración de documentación			20 Horas
Fase3: Utilizar pruebas de rendimiento y usabilidad para verificar la calidad de la aplicación Web.	lun 04/08/14	vie 22/08/14	75 Horas
Analizar los métodos y herramientas necesarias para realizar pruebas de rendimiento y usabilidad.			15 Horas
Realizar pruebas de rendimiento.			15 Horas
Realizar pruebas de usabilidad.			15 Horas
Elaboración de la documentación.			10 Horas
Fase 4: Finalización del proyecto	lun 25/08/14	vie 05/09/14	15 Horas
Elaborar la memoria final de (TT)			15 Horas

Figura 51. Cronograma estimado de actividades del Trabajo de Titulación.

Anexo 20.5: Ficha técnica del vehículo



FICHA TECNICA DE VEHICULOS Y ESTADO MECANICO

ESPECIFICACIONES TECNICAS							
Registro	Tipo	Marca	Modelo	Cilindraje	Año	Nro. de Chasis	Nro. de Motor
01-344	CAMIONETA	CHEVROLET	DIMAX	3000 CC	2009	8LBETF3FO90001081	4JH1-697084
Placa	Color	Combustible	Tracción	Cabina	Llantas/Aros	Nro. Cilindros	Avalúo
	AZUL	DIESEL	4X2	DOBLE	R16	4	12000

ESTADO MECANICO							
MOTOR	ESTADO	TRANSMISIÓN	ESTADO	DIRECCIÓN	ESTADO	FRENOS	ESTADO
Potencia Motor	<i>BUENO</i>	Embrague	<i>BUENO</i>	Caja de dirección	<i>BUENO</i>	Cilindro principal	<i>BUENO</i>
Sistema de lubricación	<i>BUENO</i>	Caja de cambios	<i>BUENO</i>	Articulaciones y terminales	<i>BUENO</i>	Cilindros secundarios	<i>BUENO</i>
Sistema de refrigeración	<i>BUENO</i>	Diferenciales	<i>BUENO</i>	Pines bocines y rotulas	<i>BUENO</i>	Forros de freno	<i>REGULAR</i>
Sistema de admisión y escape	<i>BUENO</i>	SUSPENSIÓN	ESTADO	Bombas de la dirección	<i>BUENO</i>	SISTEMA ELECT.	ESTADO
Sistema de combustible	<i>BUENO</i>	Muelles/Amort	<i>BUENO</i>	OTROS	ESTADO	Baterías	<i>REGULAR</i>
Soportes	<i>BUENO</i>	Eje delantero	<i>BUENO</i>	Llantas	<i>REGULAR</i>	Arranque	<i>BUENO</i>
Instrumentos de control	<i>BUENO</i>	Eje posterior	<i>BUENO</i>	Carrocería	<i>MALO</i>	Alternador	<i>BUENO</i>



RESUMEN	ESTADO	UBICACION ACTUAL
	<i>BUENO</i>	PROYECTO SARAGURO-LOJA

MINISTERIO DE TRANSPORTE Y OBRAS PÚBLICAS

DIRECCIÓN PROVINCIAL DE

CONTROL DE ESTADO DE EQUIPO CAMINERO

CLAVE DE NÚMERO Y COLOR	1		2		3			4			5			EQUIPO PARA BAJA O REMATE																										
	EQUIPO BUENO INACTIVO	EQUIPO TRABAJANDO NORMAL	EQUIPO TRABAJANDO CCN FALLAS	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN	EQUIPO EN REPARACIÓN																											
CLASE	MARCA	MODELO	Nº REGISTRO	LOCALIZACIÓN	MES DE ENERO 2014: AREA DE TALLERES												ELABORADO: ING. J. CARLOS VINUEZA																							
E	Q	U	I	P	O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
Bachadora	Super Cifa		30-007	El Tingo																																				
Bachadora	Super Cifa		30-001	El Tingo																																				
Barredora	Rosco		RB38	Talleres																																				
Barredora	Rosco		RB38	Talleres																																				
Barredora	BROCEB		RCT 350	Talleres																																				
Bus	Hino		KB 212	Talleres																																				
Camión Baja	Volkswa		BR051657	Talleres																																				
Carro Taller	Volk. 9-150		BR049664	Talleres																																				
Carro Taller	Nissan		CMB	Talleres																																				
Carro Taller	M-Benz		15118	Talleres																																				
Carro	Nissan		CMB	Cariamanga																																				
Camioneta	Chevrolet		87	Talleres																																				
Camioneta	Chevrolet		LUV	Talleres																																				
Camioneta	Toyota		STUD	Cariamanga																																				
Camioneta	Toyota		STUD	Talleres																																				
Camioneta	Chevrolet		LUV	Talleres																																				
Camioneta	Chevrolet		LUV	Talleres																																				
Camioneta	Chevrolet		LUV	Talleres																																				
Camioneta	Chevrolet		LUV	Loja-Vicabamb																																				
Camioneta	Chevrolet		LUV	El Tingo																																				
Camioneta	Chevrolet		Dimax	Talleres																																				
Camioneta	Chev.		Dimax	Alamor																																				
Camioneta	Mazda		2007	Loia																																				

Anexo 20.7: Licencia



Anteproyecto del Software para la gestión de información del estado mecánico de los vehículos del Ministerio de Transporte y Obras Públicas por Carla Lisbeth Romero Córdova, Yesica Norelly Saquicela Celi se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](#).