

**UNIVERSIDAD RICARDO PALMA**  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**“DESARROLLO DE UNA APLICACIÓN  
WEB PARA EL MONITOREO DE  
VEHÍCULOS CON DISPOSITIVOS GPS QUE  
COMERCIALIZA UNA EMPRESA DE  
TELECOMUNICACIONES”.**

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO INFORMÁTICO

PRESENTADO POR  
**Bocanegra Ureta, Rubén Gabriel**

**LIMA - PERÚ 2012**

<b>AGRADECIMIENTOS</b> .....	4
<b>RESUMEN</b> .....	5
<b>ABSTRACT</b> .....	6
<b>INTRODUCCIÓN</b> .....	7
<b>CAPÍTULO I - VISIÓN DEL PROYECTO</b> .....	9
Introducción: .....	9
<b>CAPÍTULO II - MODELADO DEL NEGOCIO</b> .....	35
Introducción: .....	35
<b>CAPÍTULO III - REQUERIMIENTOS DEL PROYECTO</b> .....	42
Introducción: .....	42
<b>CAPÍTULO IV - ARQUITECTURA</b> .....	105
Introducción: .....	105
<b>CAPÍTULO V - DESARROLLO Y PRUEBAS</b> .....	291
Introducción .....	291
<b>CAPÍTULO VI - GESTIÓN DEL PROYECTO</b> .....	313
Introducción .....	313
<b>CAPÍTULO VII - CONCLUSIONES Y RECOMENDACIONES</b> .....	326
Introducción .....	326
Conclusiones .....	326
Recomendaciones .....	327
<b>BIBLIOGRAFÍA</b> .....	328
<b>GLOSARIO</b> .....	330
<b>SIGLARIO</b> .....	332

*DEDICATORIA Dedicado a mi único Señor y Salvador Jesucristo, digno de toda la adoración, gloria, honra y alabanza por siempre. “El muda los tiempos y las edades; quita reyes y pone reyes; da la sabiduría a los sabios y ciencia a los entendidos. El revela lo profundo y lo escondido; conoce lo que está en tinieblas y con Él mora la luz”. Daniel 2, 21-22*

## **AGRADECIMIENTOS**

*A mis padres que con su esfuerzo y dedicación lograron formarme en cada etapa de mi vida.*

*A mis abuelos que con su cariño y comprensión me enseñaron el significado de la dedicación.*

*A mis profesores de la facultad que con sus consejos y enseñanzas se convirtieron en maestros y amigos.*

*A mi asesor de Tesis, que con su paciencia y dedicación me dio un valiosísimo apoyo para lograr este sueño.*

*Y en especial a Dios que con su gracia y misericordia permite que alcance cada meta a fin de que se cumpla su propósito en mi vida.*

## RESUMEN

Este proyecto de investigación aplicada presenta la metodología de desarrollo comprendida en la realización de una solución de monitoreo para vehículos con dispositivos GPS, el software propuesto forma parte del servicio de monitoreo vehicular que ofrece una empresa de telecomunicaciones. Se describe y especifica la arquitectura del sistema a través de casos de uso en las diversas disciplinas del proceso de desarrollo de software.

Si bien existen diversas soluciones de monitoreo vehicular que son usadas en distintos centros de control, la solución propuesta agrupa las funcionalidades representativas de las mejores soluciones de monitoreo del mercado y las presenta de manera objetiva a través de una propuesta completa y puntual donde el usuario final pueda dar uso de la solución a través de un navegador web, así mismo se muestra el diseño de la solución utilizando las disciplinas y patrones que exige la ingeniería de software.

Palabras Clave: Monitoreo Vehicular, Google Maps, GPS, dispositivos de localización, geocercas, aplicaciones de monitoreo.

## **ABSTRACT**

This applied research Project presents the methodology of development understood in the achievement of a tracking solution for vehicles with GPS devices, the proposed software is part of a service of vehicle tracking offered by a telecommunications company. The system's architecture is described and specified around the use cases in the multiples disciplines of the software process development.

Although, there are multiples vehicle tracking solutions which are used in differents control centers, the offered solution gather the most representative functionalities from the best tracking solutions in the market and present it on objective way through in a complete and punctual proposal where the final user can use the solution using a web browser, likewise is shown the design of the solution using the disciplines and patterns requested by the software engineering.

Keywords: Vehicle Tracking, Google Maps, GPS, localization devices, geofences, tracking applications.

# INTRODUCCIÓN

En los últimos años han ocurrido cambios radicales en el modo de cómo las personas establecen comunicación, hace apenas unos 20 años en Perú era común esperar al cartero visitando nuestras casas trayendo noticias acerca de familiares lejanos y si queríamos enterarnos de algo importante veíamos el noticiero por la noche a fin de estar actualizados. Sin embargo en los últimos diez años ha habido cambios radicales con respecto a cómo las personas se ponen en contacto, desde la aparición de la internet, la reducción de costos en el uso de celulares y la competencia entre empresas de telecomunicaciones que cada año lanzan nuevas constelaciones de satélites para reducir costos de transmisión y ganar más cliente.

Este nuevo esquema genera que las personas no consideren al uso de servicios de telecomunicaciones como una opción, ahora consideran dichos servicios como una necesidad importantísima, haciéndolas fundamentales para su vida diaria y casi todos sus procesos de negocio. Y es en esa convergencia entre la vida diaria de la persona y sus procesos de negocio donde los servicios de telecomunicaciones hacen mayor eco, ya que la mayoría de usuarios consideran que es mejor estar al tanto de todo en todo lugar que solo estar al tanto de todo en la oficina o en donde se labore. Esta nueva necesidad trae consigo nuevos servicios y demanda de mejores aplicaciones que exploten de manera optimizada estos servicios, si antes el gerente de una compañía quería estar al tanto de las facturas canceladas debía de ir a su oficina, prender su ordenador, acceder a una aplicación comúnmente llamada “Integrada” e imprimir un reporte a fin de saber cómo se va desenvolviendo la labor de cobranzas. Ahora ese mismo gerente lleva su laptop de 14 pulgadas en su maletín y se conecta en una cafetería con internet inalámbrico, revisa su correo electrónico y recibe el reporte de facturas canceladas en las últimas 4 horas como archivo adjunto, junto con una serie de indicadores que le ayudan a tomar decisiones.

Pero estas “facilidades” o nuevos “servicios” no están dirigidos a exclusivos sectores de personas como gerentes de bancos que pueden acceder a su información desde cualquier parte, ahora cualquier persona puede pagar unos cuantos dólares al año y ser usuario de un sistema CRM tipo SaaS “Software as a Service” y gestionar la información de sus clientes potenciales desde su casa y enviar cotizaciones sin mayor apuro. Como podemos apreciar el uso de la información se ha liberado de tal forma que cualquiera con acceso a una PC y a internet puede gozar los beneficios que las empresas de telecomunicaciones y servicios en internet ofrecen, y esto nos lleva al título de la presente investigación aplicada “Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de telecomunicaciones”.

Hace una década en nuestro país, las instituciones que poseían una flota de vehículos que deseaban realizar el monitoreo de estos, ya sea para estar al tanto de los incidentes ocurridos o hacerle seguimientos a los recorridos, debían de hacer grandes inversiones en infraestructura para implementar dichos servicios, desde la compra de servidores para la gestión de datos y de mapas (en caso de ser gráfica la visualización), contratos de una línea dedicada con el proveedor de telecomunicaciones, los dispositivos GPS, el software para el monitoreo de los vehículos y la asignación de personal dedicado para el monitoreo de la información y que tome la acción adecuada para cada tipo de evento entre otros. El hecho de monitorear vehículos estaba limitado a solo aquellos que podían realizar ese tipo de inversión, es decir a grandes empresas y a empresas públicas, que tengan necesidad de monitorear vehículos que lleven mercancía de mucho valor o por algún otro motivo de fuerza mayor estén obligadas a realizar dicha actividad.

En estos últimos 10 años las tarifas de conexión celular y satelital que son la “materia prima” para los dispositivos GPS se han reducido de manera significativa, sumado a la mejora continua de los dispositivos GPS han hecho que los costos de comunicación se hayan reducido aún más, por ejemplo, en la actualidad existen equipos híbridos que aprovechan la cobertura celular para transmitir información cuando detectan redes celular y solo usan la cobertura satelital cuando salen de la cobertura celular, ahorrando costos de manera drástica a los usuarios de estos equipos. Todo esto ha impulsado que el monitoreo de vehículos terrestres tenga un florecimiento sorprendente

con la reducción de costos de equipo y la transmisión de la información, generando la demanda de un nuevo servicio que es “Ofrecer el servicio de monitoreo vehicular a cualquier persona que pueda adquirir un dispositivo GPS y pagar por la transmisión” tomando en cuenta que ahora en muchos casos la transmisión viene a ser un pago fijo mensual y el equipo GPS esta ya pagado dentro de los pagos fijos que realiza el usuario a las empresas de telecomunicaciones.



# **CAPÍTULO I - VISIÓN DEL PROYECTO**

## **1.1 Introducción:**

Generalmente cuando las organizaciones deciden implementar un proyecto de desarrollo de software, existen varias posibilidades que el proyecto fracase por varios factores, uno de ellos es el exceso de entusiasmo por parte de los stakeholders, que con poco fundamento, asignan recursos técnicos y presupuestos, haciendo que los proyectos se cancelen ni bien se dan por iniciados. La visión del proyecto es un conglomerado de metas y objetivos claros que el equipo de desarrollo y los demás participantes del proyecto pueden cumplir aportando beneficios a la compañía. Si bien existen diversos temas de gestión que serán tratados en el Capítulo 6, la visión del proyecto explica claramente que es lo que se pretende realizar en función a una problemática puntual y que beneficios generará el presente proyecto de investigación tomando en cuenta todos los factores de la compañía que envuelven al proyecto de desarrollo.

### **1.1.1 Convenciones:**

La organización donde se realizará la investigación aplicada es una empresa privada de telecomunicaciones satelitales que posee una importante participación en el mercado sudamericano, por términos de confidencialidad el nombre de la empresa no será mencionado en la presente investigación y la llamaremos con el nombre de TELCO (Abreviatura de empresa de telecomunicaciones) al referirnos acerca de dicha empresa.

### **1.1.2 Tematización:**

Debido a que el estándar tecnológico de los productos software para el monitoreo de vehículos y las tecnologías más recientes están basados en web, el estudio trata acerca de llevar las funcionalidades características de un sistema de monitoreo vehicular con dispositivos GPS a una aplicación basada en la plataforma web. El estudio tendrá como fuente la información que arrojan los dispositivos GPS suministrados por la TELCO junto con el sistema de monitoreo vehicular que usan actualmente.

## 1.2 El Problema:

### 1.2.1 El Negocio

La TELCO (Ver Sección 1.1.1) es la empresa donde realizaremos la presente investigación aplicada, la TELCO es una de las empresas líderes de telefonía satelital en el Perú, sus clientes son principalmente empresas que necesitan de conexión donde las redes de comunicación convencionales no cubren sus expectativas.

### La Organización



Figura 1.1: Organización de la TELCO

## **La Misión**

Proporcionar un alto valor al cliente en servicios de telecomunicaciones, excediendo sus expectativas.

## **La Visión**

Consolidarse como una empresa innovadora cercana al cliente y liderando con sus líneas de negocio a los segmentos en los que participa en la industria de telecomunicaciones.

### **1.2.2 Los procesos del Negocio**

Debido a que la TELCO es una empresa con una gama de procesos en diversas áreas, nos centraremos específicamente en los procesos relacionados al monitoreo vehicular, no obstante describiremos una breve introducción de lo que comprende el negocio.

#### **1.2.2.1 Los servicios**

Los principales servicios que ofrece donde provee de conexión satelital son los siguientes:

1. Telefonía Satelital en tierra y mar.
2. Transmisión de Datos en modo simplex, dúplex y alta velocidad.
3. Monitoreo de vehículos.

Es en este último punto donde centraremos la investigación, ya que la TELCO ofrece el servicio de monitoreo de vehículos que utilizan sus equipos GPS.



Imágenes obtenidas de [www.globalstar.com](http://www.globalstar.com)

**Figura 1.2: "MOVIL GSP 1700" Equipos de Transmisión de Voz**



Imágenes obtenidas de [www.globalstar.com](http://www.globalstar.com)

Figura 1.3: "BGAN EPLORER 111" Equipo de Transmisión de Voz y Datos



Imagen obtenida de [www.skypatrol.com](http://www.skypatrol.com)

Figura 1.4: "SKYPATROL EVOLUTION" Equipo utilizado para monitoreo vehicular

### 1.2.2.2 El Monitoreo de Vehículos

Los clientes de la TELCO pertenecen a diferentes sectores, como minería, pesca, transporte de valores, transporte interprovincial, transporte aéreo, fuerzas armadas, etc. Todos estos clientes tienen vehículos de diferentes tipos que recorren el territorio nacional en toda su extensión y requieren de la información necesaria para determinar el estado de sus vehículos, la ruta que siguen estos y estar informados de cualquier eventualidad.

La TELCO les ofrece a estos clientes equipos GPS (Sistema de Geoposicionamiento) y el servicio de transmisión de datos que envían estos equipos, es conocido para los clientes que la cantidad y detalle de la información varía en función al modelo del equipo GPS, así mismo el costo de la transmisión varía en función a la cantidad de información que envía el equipo en un tiempo determinado.

Los diferentes modelos de equipos GPS pueden enviar información de todo tipo en función a los sensores que este posea, por ejemplo entre los más comunes tenemos:

- ✓ Latitud
- ✓ Longitud
- ✓ Altitud
- ✓ Velocidad
- ✓ Rumbo
- ✓ Entre otros

Los sensores de estos equipos GPS junto al software que ellos posean para gestionar sus sensores, pueden así mismo enviar información acerca de eventos, por ejemplo:

- ✓ Encendido del Vehículo
- ✓ Trama Normal
- ✓ Ingreso a Geocercas (En caso de que estos equipos puedan registrar geocercas)
- ✓ Alarma de exceso de velocidad
- ✓ Botón de pánico
- ✓ Entre otros

Cada vez que el equipo GPS envía información, está enviando información de un evento sucedido, generalmente son eventos que indican sólo la posición o “tramas normales”, pero existen ocasiones que los sensores de estos equipos en función al software que poseen pueden detectar eventos como el ingreso a una geocerca o la presión del botón de pánico por más de 3 segundos y envían la información de siempre añadiendo la información relevante del evento ocurrido.

Así mismo existen modelos que envían su información por redes celulares cuando detectan una y por la red satelital cuando salen de la cobertura celular, a estos modelos se le llaman híbridos y ahorran costos de transmisión, debido a que el uso de la red celular es menos costoso que la red satelital. Algunos equipos pueden ser configurados mediante el envío de comandos cuando están transmitiendo y así cambiar el comportamiento de transmisión o configurar nuevos eventos en función a sus sensores. Como podemos apreciar, estos equipos no sólo transmiten la posición geográfica, además de esto, entregan información relevante del comportamiento del vehículo que es de suma importancia para sus usuarios y las personas que están pendientes de estos.

A fin de que los clientes estén en pleno conocimiento de la información que transmiten los vehículos, la TELCO contacta con un proveedor de software que instala una aplicación donde podrán realizar el monitoreo de sus vehículos, esta aplicación se conecta con la base de datos de la TELCO y extrae la información relevante al vehículo a monitorear. Queda claro para los clientes que el servicio de instalación del software y la licencia del mismo es un precio fuera del servicio de transmisión de datos que se acuerda directamente con el proveedor de software, en caso de que el cliente desee utilizar otra instancia del software, deberá adquirir otra licencia del mismo.

Existen clientes que no pueden costear la instalación y la licencia del software de monitoreo de vehículos, para eso la TELCO posee el servicio de atención al cliente donde ante cualquier evento de emergencia o consulta de reporte, los operadores del servicio de atención al cliente hacen las gestiones necesarias y contactan con el cliente.

Las nuevos avances en la tecnología de los dispositivos GPS y la reducción en los costos de transmisión de datos vía satelital y celular, han hecho que la información que transmiten estos dispositivos sea cada vez mayor y con más detalle, lo cual ha hecho que el software que actualmente usa la TELCO y sus clientes para el monitoreo de vehículos sea obsoleto, así mismo el incremento de la demanda del servicio monitoreo de vehículos de otros sectores de mercado que no pueden costear la instalación del software actual ni del pago de la licencia, han hecho que la

TELCO proponga una solución económica a fin de que los nuevos clientes y los actuales aprovechen los beneficios de sus dispositivos GPS, utilicen el servicio de transmisión de datos y que el costo del servicio monitoreo vehicular quede inmerso dentro del precio de la transmisión sin afectar de manera significativa el precio del servicio en general.

### **1.2.2.3 Descripción de los procesos del negocio**

Los procesos de negocio principales en el servicio de monitoreo de vehículos son los siguientes:

#### **1.2.2.3.1 Comercialización**

Un asesor comercial de la TELCO presenta al cliente los diferentes modelos de equipos GPS junto con la tarifa de la transmisión de la información. El cliente selecciona los equipos que usará su flota de vehículos y firma un contrato por la prestación del servicio de transmisión de datos

#### **1.2.2.3.2 Servicio Técnico**

El asesor comercial entrega las especificaciones técnicas requeridas por el cliente y el equipo técnico de la TELCO instala los equipos GPS en los vehículos asignados. Al instalar un dispositivo GPS, registran el número de serie del equipo y crean un identificador lógico del mismo, que servirá más adelante para identificar el origen de la información que emitirá el equipo, así mismo configuran los sensores y la frecuencia de transmisión que ejecutará el equipo en función a las especificaciones del cliente.

#### **1.2.2.3.3 Instalación del Software de Monitoreo**

Si el cliente desea tener la aplicación software dentro de sus instalaciones, el asesor comercial contacta con la empresa proveedora del software y esta pacta con el cliente por el servicio de instalación y licencias.

Actualmente, la empresa cuenta con 100 clientes que utilizan el servicio de monitoreo vehicular.

#### **1.2.2.3.4 Atención al Cliente**

Si el cliente no posee una instalación del software de monitoreo o por alguna razón esta es inaccesible al cliente. El cliente contacta con el servicio de atención al cliente de la TELCO, el personal del servicio de atención al cliente utiliza el software de monitoreo e informan al cliente el estado del vehículo y en algunas ocasiones envían comandos al equipo para re-configurarlo y este pueda desempeñarse en función a los nuevos requerimientos del cliente. En caso de que el equipo presente desperfectos, el equipo es enviado al área de Servicio Técnico quien es responsable de verificar las causas del desperfecto y repararlo si es posible en función a la garantía del equipo y a los términos contractuales previos.

### **1.2.3 Descripción del Problema:**

La TELCO como empresa de telecomunicaciones satelitales, ofrece actualmente a sus clientes dispositivos GPS y transmisión satelital para sus vehículos, los clientes de la TELCO instalan estos dispositivos en sus vehículos con la finalidad de monitorear la posición geográfica de estos, este monitoreo es realizado a través de una aplicación de monitoreo vehicular de escritorio que utiliza una plataforma GIS instalada en las computadoras de los clientes. El problema radica que la TELCO necesita instalar la aplicación de monitoreo vehicular en las computadoras de los centros de control de cada cliente, este proceso demanda tiempo a la TELCO ya que necesita asignar dichas actividades a personal técnico capacitado y exige al cliente tener una infraestructura de hardware y software específica para utilizar la aplicación de monitoreo vehicular, por consecuencia el proceso de comercialización de los dispositivos GPS y transmisión satelital se ve afectado.

## **1.3 Fundamentación del Problema**

### **1.3.1 Marco Teórico**

El monitoreo de vehículos con dispositivos o receptores GPS, abarca en la actualidad un abanico extenso de tecnologías, desde el concepto de sistemas de posicionamiento, sistemas de información geográfica y aplicaciones software. Como es de suponer todas estas tecnologías tienen una profunda base de conocimientos y distintas aplicaciones, debido a esto, nos enfocaremos en explicar de manera objetiva estas y su influencia en el servicio de monitoreo de vehículos.

#### **1.3.1.1 Sistema de Posicionamiento Global**

Según [RAMOS-BOSCH 08] el Sistema de Posicionamiento Global (GPS) es el más conocido y usado sistema de navegación por satélite. Que consiste en por lo menos una red de 24 satélites orbitando a 20200 km alrededor de la superficie terrestre y que continuamente transmiten señales de frecuencias duales que son procesadas por los receptores GPS. En un principio el GPS fue diseñado para uso militar ya que el 22 de Febrero de 1978 fue lanzado el primer satélite y fueron 2 eventos clave que hicieron que el GPS fuera de uso completamente civil. El primero en 1983 cuando fue abierto al público y el 2000 cuando se desactivó la accesibilidad selectiva, que consistía en una degradación intencionada de la transmisión.

[BAO-YEN 00] menciona que la navegación por satélite comenzó a principios de 1970, ya que tres sistemas de satélites fueron explorados antes que se diera inicio al programa GPS: El Sistema de Navegación por Satélite de la marina de Estados Unidos, conocido como TRANSIT, el sistema TIMATION de la marina de Estados Unidos, su nombre viene del agrupamiento de las palabras (Time Navigation), y el proyecto de la fuerza de Estados Unidos 621B. Estos 3 proyectos proporcionaron diversos alcances técnicos que enriquecieron los inicios del programa GPS. El programa GPS fue aprobado en Diciembre de 1973. El primer satélite del programa fue lanzado en 1978. En Agosto de 1993, el sistema GPS ya tenía alrededor de 24 satélites en órbita y en Diciembre del mismo año la capacidad operativa inicial fue

establecida. En Febrero de 1994, la Agencia de Aviación Federal (FAA) declaró al GPS listo para el uso de la aviación.

Según [POZO-RUZ 00] indica que el principio de un sistema GPS es calcular la posición de un punto en el espacio de coordenadas  $(x,y,z)$ , basándose en las distancias del punto a un mínimo de tres satélites cuya ubicación es conocida. La distancia del equipo GPS y el satélite se mide calculando el tiempo que demora llegar la señal al punto de referencia. Para poder calcular este tiempo, los relojes de los satélites deberán estar sincronizados. Si bien los relojes de los satélites son muy precisos por lo contrario los relojes de los equipos GPS o receptores no lo son tanto debido a que son fabricados a bajo coste, a fin de corregir cualquier inconveniente, es usado un cuarto satélite para la corrección de los cálculos y así encontrar la posición exacta del receptor o equipo GPS.

### **1.3.1.2 Receptor GPS**

[BAO-YEN 00] describe a un receptor GPS como un dispositivo electrónico compuesto por una antena, una cadena de radio frecuencia y convertidor análogo digital, que se comportan de la siguiente manera: La señal transmitida del satélite GPS son recibidas por la antena. A través de la cadena de radio frecuencia (RF) la señal de entrada es amplificada a una adecuada amplitud y la frecuencia es convertida a la frecuencia de salida deseada. Luego la señal de salida es digitalizada por un convertidor análogo digital (ADC). La antena, la cadena RF y el ADC son dispositivos usados en el receptor GPS. Luego que la señal es digitalizada, se puede utilizar un software para localizar las coordenadas del receptor GPS, a la cual se llama fase de transición de la información de navegación. En un receptor convencional esta fase es desempeñada por el hardware. Los pseudo-rangos y la información de efemérides pueden ser obtenidas de la información de la navegación. Con la información de efemérides se puede obtener la posición de los satélites. Por último la posición del receptor puede ser calculado por la posición de los satélites y el pseudo-rango.

[RODRIGUEZ-SOLANO 99] hace una descripción muy acertada de las ventajas que ofrece las un receptor GPS y sus diversos usos, debido a su tamaño reducido es fácil de transportar en diversos tipos de vehículos, como bicicletas, autos, helicópteros, y así cartografiar pistas y carreteras, describiendo a un receptor GPS como una gran herramienta para un sistema de información geográfica.

### **1.3.1.3 Sistema de Información Geográfica**

[SANTIAGO QUINTANA 08] Menciona que un Sistema de Información Geográfica puede tener 3 conceptos, el primero, como una aplicación software que permite manejar y visualizar mapas, el segundo concepto indica que una aplicación SIG es aquella aplicación software que permite manejar mapas y además realizar operaciones de análisis como superposiciones vectoriales. Por último indica que un SIG no sólo es la aplicación software que permite manejar y visualizar mapas junto con operaciones vectoriales, sino que además engloba todo el hardware involucrado y software anexo para procesar toda esa información.



Para [LONGLEY 05] GIS o un Sistema de Información Geográfica es una ciencia que involucra la descripción, explicación y predicción de patrones y procesos a escalas geográficas, así mismo es una disciplina que se utiliza para resolver problemas mediante una metodología. Siendo un GIS ya una tecnología probada que hoy en día provee fundamentos sólidos al análisis de datos basados en mapas del mundo real. Así mismo con el avance de la tecnología, hace que los GIS puedan superar sus expectativas actuales, y más aún con el uso del internet, nuevas investigaciones impulsan el mejoramiento día a día de los Sistemas de Información Geográfica.

Para [BONHAM-CARTER 06] el término de GIS es mucho más que un simple programa computacional, es toda una ciencia que tiene adeptos alrededor del mundo, tanto universidades como institutos técnicos dictan cursos basados en el manejo de un GIS. Sin un GIS sería tremendamente difícil tratar de procesar información geográfica basada en capas, ya que un GIS se especializa en analizar y manejar información geográfica basada en capas.

#### **1.3.1.4 Sistema de Monitoreo Vehicular**

[TAT-JEN 07] define que un Sistema de Monitoreo Vehicular como aquel sistema que se encarga de administrar y analizar la información de la posición de un vehículo proveniente de un dispositivo inalámbrico en tiempo real, así mismo, un Sistema de Monitoreo Vehicular es el centro de cualquier Sistema de Información de Transporte. Este dispositivo inalámbrico puede ser un dispositivo GPS o un transceptor de radio frecuencia o RF instalado en el vehículo, en caso de que el vehículo transite en la cobertura de una red inalámbrica puede usar un transceptor RF a fin de ahorrar costos, debido a que la conexión GPS suele ser más costosa.

Así mismo el autor, detalla la estructura básica de una tabla que maneja información de un Sistema de Monitoreo Vehicular, esta tabla por lo menos deberá poseer 3 campos, los cuales son:

Grupo: en caso de que la red de sensores o transmisores se dividan en grupos, se puede prevenir transmisión innecesaria.

Fuente: el número del nodo que directamente transmite el mensaje.

Tiempo: fecha de la recepción del mensaje.

[KAPLAN 06] afirma que la función de una aplicación de monitoreo vehicular, es aquel sistema donde la posición del vehículo es determinada y luego enviada de manera inalámbrica a una instalación de control centralizado. Así mismo describe la arquitectura de un Sistema de Monitoreo Vehicular, como la composición de un receptor GPS, sensores auxiliares conectados al receptor GPS y un procesador de computadora para controlar los componentes y calcular la solución de la posición optimizada.

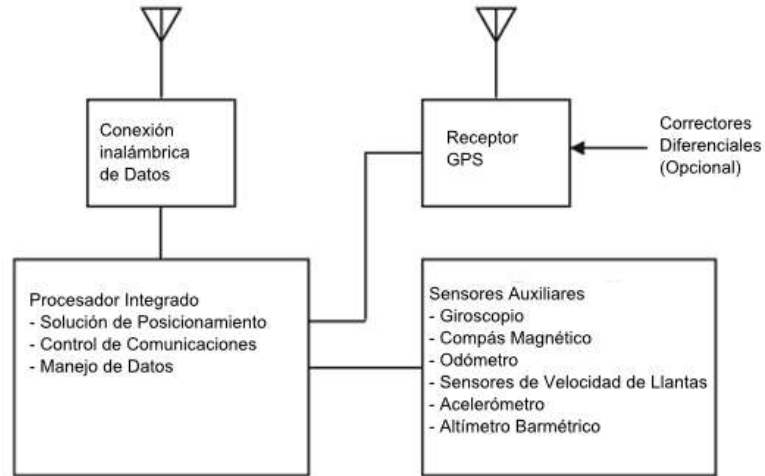


Imagen obtenida de [KAPLAN 06]

Figura 1.5: Arquitectura de un Sistema de Monitoreo Vehicular

[THANGAVELU 07] afirma que los sistemas de monitoreo vehicular son dispositivos electrónicos en vehículos para permitir a los dueños de los vehículos o a terceros, rastrear la ubicación del vehículo. Los más modernos sistemas de monitoreo vehicular usan equipos GPS para asegurarse la ubicación exacta del vehículo. Así mismo muchos sistemas de monitoreo vehicular combinan características celulares y satelitales para transmitir la posición del vehículo al usuario.

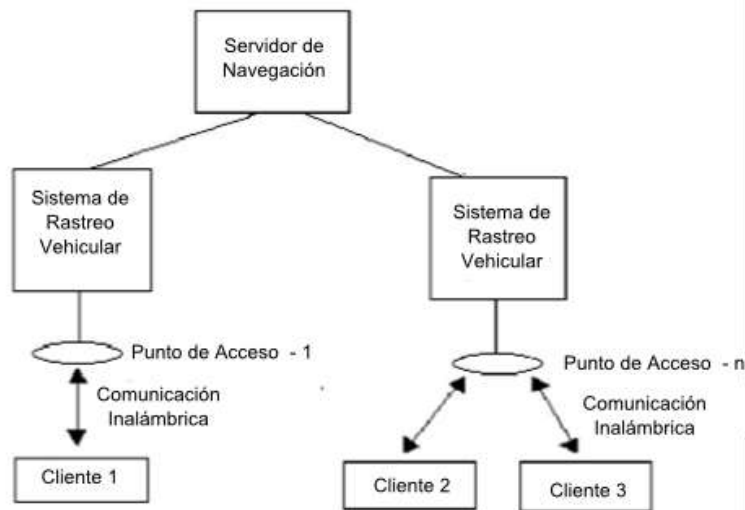


Imagen obtenida de [THANGAVELU 07]

Figura 1.6: Ilustra la arquitectura entre un servidor de navegación y el sistema de monitoreo vehicular.

### 1.3.1.5 Empresa de Telecomunicaciones

[BELL 05] establece que una empresa de telecomunicaciones es aquella que ofrece utilizar enteramente la tecnología telefónica como medio de comunicación.

[GLOBALSTAR 11] describe a Globalstar como una compañía que ofrece servicios de alta calidad en voz y datos vía satélite a través de Norte América y alrededor de 120 países. Los servicios de voz y datos son ofrecidos a través de unidades móviles para satisfacer las necesidades de negocio y recreacionales de sus usuarios. Así mismo afirma que los servicios de Globalstar ayudan a entregar información en áreas remotas donde información clave operacional o de negociaciones es recolectada, ahorrando tiempo y dinero a una infinidad de empresas.

[IRIDIUM 11] afirma que Iridium Communications Inc. es la única empresa de telecomunicaciones satelitales que ofrece realmente una cobertura global del servicio de comunicación de voz y datos, siendo esta una empresa que innova en tecnología y un líder en el mercado de empresas de telecomunicaciones satelitales. Así mismo indica que los servicios de Iridium es la forma más avanzada de conducir actividades diarias de misión crítica a través de servicios de comunicaciones confiables y casi en tiempo real.

#### **1.3.1.6 Comercialización de servicios de telecomunicaciones**

[LOUIS 02] establece que una empresa de telecomunicaciones está comprendida de distribuidores y vendedores de hardware y software, la cual se caracteriza por un servicio casi instantáneo prestado entre distribuidores y clientes, y entre vendedores. Donde los proveedores de servicios tienen acuerdos con los vendedores, y poseen representaciones locales en cada mercado. Los distribuidores nacionales, poseen múltiples oficinas en un único mercado para satisfacer las necesidades de los clientes, también poseen fuerzas de campo a fin de reponer diversos equipos de telecomunicaciones en cada mercado local para su venta.

[REIF 97] ofrece una amplia definición de los servicios que ofrece una empresa de telecomunicaciones, hace una división entre proveedores de telefonía local, larga distancia, proveedores de acceso competitivo (CAPs), servicios de comunicación inalámbrica, operadores de televisión por cable, y servicios de valor añadido incluyendo una variedad de ofertas especializadas por la red regular de telefonía o vía redes especiales de comunicación.

El término de valor añadido, si bien no está muy bien claro en la industria actualmente, pero agrupan los servicios de transmisión de paquete y protocolos de conversión, base de datos en línea y páginas amarillas electrónicas, servicios de mensajería y conferencia, como mensajes de voz, correo electrónico, servicios especializados de fax, audio conferencia y servicios especializados de datos como frame relay, servicios de procesamiento de transacciones como intercambio electrónico de datos EDI. Así mismo incluye sistemas avanzados de PBX para manejar comunicaciones de fax y voz, y en algunos casos transferencias de voz entre computadoras, la mensajería de texto y la telefonía celular, entre otros.

### 1.3.1.7 Aplicación Web

[JABLONSKI 04] afirma que una aplicación web no tiene nada de extraordinario comparado con una aplicación de computadora, ya que no deja de tomar en cuenta las reglas tradicionales del diseño de una aplicación. Una aplicación Web es una aplicación que muestra páginas HTML generadas dinámicamente y sus componentes son similares a una aplicación computacional.

Si se analiza la arquitectura conceptual, en una aplicación web se puede reconocer ciertos componentes como el paquete de presentación, lógica del negocio y manejo de datos, en otras palabras la interfaz de usuario, la funcionalidad de la aplicación y la manipulación de la información.

Si bien una aplicación web es una aplicación computacional estándar, tiene características específicas, por ejemplo una aplicación web no necesita ser ejecutable, se basa principalmente en el contenido, la interfaz de usuario está basada en páginas HTML generadas dinámicamente y la navegación está basada en hipervínculos.

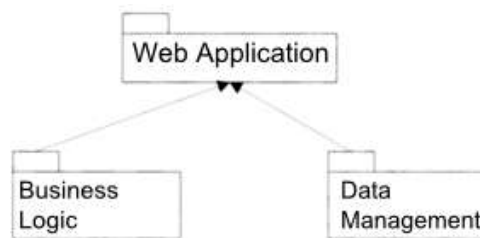


Imagen obtenida de [JABLONSKI 04]

Figura 1.7: La Arquitectura de una Aplicación Web

Según [SHKLAR 03] una aplicación web es una agrupación de protocolos y lenguajes asociados, que incluye al protocolo HTTP (HyperText Transfer Protocol) y al lenguaje HTML (HyperText Markup Language) que son fundamentales al momento de la creación y de la transmisión de una página web. Así mismo hace mención de otros protocolos de internet que intervienen en una aplicación web, por ejemplo: TELNET, FTP, SMTP e IMAP, así mismo lenguajes avanzados como XML. A esto le añade el conocimiento de bases de datos y presentación multimedia.

El autor afirma que la idea de una aplicación web es casi la suma de todos los conceptos, ya que las personas que diseñan aplicaciones web, no sólo deberían conocer de protocolos como HTTP y HTML, ya deberán conocer Javascript, XML, bases de datos relacionales, diseño gráfico y multimedia entre otros conceptos.

### 1.3.1.8 Proceso de Desarrollo de Software

[DERNIEAME 99] define el proceso de desarrollo de software como la manera en que el desarrollo del software es organizado, manejado, medido, soportado y mejorado (independientemente del tipo del tipo de soporte tecnológico utilizado en el desarrollo). Si bien el proceso de desarrollo de software tiene diferentes niveles y modelos, según como sea presentado, todas las organizaciones

envueltas en el proceso de desarrollo de software siguen un proceso de algún tipo, implícita o explícitamente, reproducible, instrumentado, adaptable o de otra manera. Las empresas dedicadas a desarrollar software y los negocios en general que consideran al software como un factor clave en su negocio, tiene muy claro que la clave para entregar un software exitoso tiene mucho que ver con el efectivo manejo del proceso de software.

[LARMAN 04] indica puntualmente que un proceso o método de desarrollo de software, es la organización de actividades relacionadas con la creación, presentación y mantenimiento de sistemas de software.

[JACOBSON 00] afirma que el proceso de desarrollo de software deberá ser automatizado donde el proceso es crítico y deberá ser formado por herramientas para implementar y automatizar el proceso. Así mismo declara que sin un propio soporte tecnológico el proceso es muy difícil de implementar. La automatización de un proceso de software es indispensable para hacer iteraciones posibles y manejables.

#### **1.3.1.9 Patrón de Software**

[ZAMAN 02] define un patrón de software como un diseño reusable que ha sido capturado a través de la experiencia, donde las mejores procesos han sido definidos puntualmente, y resuelven problemas específicos de manera exitosa.

[CRAWFORD 03] describe un patrón como una solución que siempre se aplica a un mismo problema. Desde el punto de vista de la programación un patrón es una solución encapsulada en un objeto para resolver un problema específico.

### **1.3.2 Estado del Arte**

#### **1.3.2.1 GIS/GPS/GPRS and Web based Framework for Vehicle Fleet Tracking**

Según Michael Medagama autor de la presente investigación aplicada [MEDAGAMA 08], presenta la plataforma web de un sistema de monitoreo de flotas, este sistema está diseñado para administrar y rastrear vehículos que son usados por empresas.

Este sistema comprende el dispositivo montada en el vehículo, un servidor central del sistema y una aplicación web. A través de este sistema los usuarios tienen la facilidad de monitorear la ubicación geográfica (a través de un mapa) y otra información relevante del vehículo.

La investigación resalta el diseño y el desarrollo relacionado a los aspectos del monitoreo vehicular del sistema, siendo este sistema diseñado para servir a empresas con grandes flotas vehiculares con complejos requerimientos de uso. Los empleados quizá necesiten usar frecuentemente para viajes cortos, y otros para distancias largas, algunos viajes necesiten ser programados de manera inmediata, y otros basados en programas diarios. Los empleados pueden ser recogidos y dejados

en algún punto, y los vehículos que estén en reparación deberán ser considerados inaccesibles. Esas son algunas funcionalidades que deberá atender el sistema. La investigación ilustra como diversas tecnologías se integran para una meta común y realizan sinergias para dar solución a una temática que se aplica en el mundo real.

### **Objetivos**

El objetivo principal de la investigación es dar solución a los siguientes problemas:

Carencia de un sistema apropiado para guardar un registro del recorrido de las flotas vehiculares.

Pérdida de productividad y de tiempo en el transporte de empleados.

Pérdida de productividad e ineficiencia al utilizar vehículos sin un programa o una ruta determinada.

El sistema pretende resolver estos problemas con:

- ✓ Una intuitiva interface web.
- ✓ Una infraestructura de comunicación disponible y eficiente.
- ✓ Herramientas de geo-información que ayuden al fácil entendimiento de la información.
- ✓ Herramientas Open-Source

### **Arquitectura del Sistema**

El sistema de monitoreo de flotas apunta a superar las limitaciones de una aplicación web. La arquitectura es cliente servidor donde el cliente es el navegador web y el servidor comparte funciones como web server, servidor de comunicaciones, servidor de base de datos y servidor de mapas. GPRS (General Packet Radio Service) es seleccionada como medio principal de comunicación entre la unidad vehicular y el servidor, y la información recolecta del GPS es almacenada en el servidor de base de datos.

### **Diseño del Software**

Los componentes principales del sistema son el sistema de comunicación vía sockets, el servidor web y el servidor de mapas. El servidor de comunicación vía sockets es el componente central que establece conexión con las unidades rastreadas, establece una conexión TCP/IP y es capaz de comunicarse con múltiples unidades utilizando múltiples hilos. Cuando una unidad vehicular se conecta, el servidor la autentica y reconoce al cliente. Luego, el servidor procederá a recibir la información del cliente y almacenarlo en la base de datos. La aplicación web tomará la data de la base de datos y la procesará cuando el usuario haga futuras peticiones. Así mismo la aplicación web es una herramienta de mantenimiento del sistema, ya que un usuario se puede ingresar al sistema y en función a su nivel de permisos puede acceder a diversos módulos funcionales de la aplicación.

## **Componentes de Geo-Información**

El módulo de geo-información maneja toda la información espacial del sistema. Los principales objetivos de estos módulos son manejar las posiciones, geocercas, rutas y tours. Las posiciones son ciudades, calles, u otros sitios identificables por la organización, las geocercas son áreas circulares definidas alrededor de una locación, el usuario puede definir la posición y el radio de esta área, en el sistema si un vehículo se encuentra en una determinada posición es porque está dentro de una geocerca. Luego se entiende por ruta al conjunto de geocercas conectadas entre sí, luego estas rutas le permiten definir al usuario un tour que es el recorrer una ruta en un horario determinado.

## **Herramientas Gis – Open Source**

Para el desarrollo del componente de mapeo del sistema, se utilizó MapServer, esta herramienta GIS es utilizada para presentar información geográfica en el mapa a través de una aplicación web. Los mapas son digitalizados en archivos de tipo shape o shp y la información relacionada estos, están almacenados en archivos de tipo database file o dbf. Los mapas son presentados como un conjunto de capas superpuestas, la información de estas capas están contenidas en los archivos shape y database file. Cuando el usuario desea ver un área geográfica específica a través del área del mapa, el sistema hace una petición al componente de mapas y este devuelve un archivo gráfico Joint Photographic Experts Group (JPEG) como producto de la combinación de las capas geográficas que maneja MapServer.

## **Utilidad para el proyecto de Tesis**

La presente investigación brinda mucha información relacionada a la arquitectura de una aplicación web de monitoreo vehicular, ya que hace una clara distinción de los componentes que intervienen en ella y como estos interactúan para entregar la información pertinente al usuario. Es rescatable de esta investigación la definición objetiva de componentes para el manejo de comunicación, manejo de la base de datos, de la aplicación web a nivel funcional y el servidor de mapas, los cuales serán definidos posteriormente en la presente investigación aplicada.

### **1.3.2.2 A Beast in the Field: The Google Maps Mashup as GIS**

Christopher Miller ha realizado una investigación sobre los sistemas de información geográfica (GIS) y Google Maps [MILLER 06], donde indica que en la última década que los GIS se han probado así mismos como herramientas ágiles y potentes en el campo académico, cívico y otras disciplinas. No obstante si bien las funcionalidades de los GIS son impresionantes, el mejor avance en sistemas de información geográfica es una aplicación web híbrida llamada Google Maps.

## **GIS y el debate de la sociedad**

El avance de los computadores personales en los ochentas y la intervención de la internet en los noventas trajeron los sistemas de información geográfica al público y al sector privado les dio un nuevo mercado para proveer información de análisis, presentación, representación y organización para suplir las nuevas necesidades del mercado, estas circunstancias permitieron la democratización del GIS. No obstante un GIS no era del todo abierto al público, ya que para tener un GIS era necesario contar con la infraestructura de hardware y software que generalmente requería una cuantiosa inversión, haciendo el uso de los GIS limitado. Era necesario que este tipo de aplicaciones sean más abiertas a la sociedad debido al tipo de información que manejan ya que la información relacionada puede ser enriquecida por los integrantes de la sociedad, esto se dio cuando apareció Google Maps.

## **Google Maps como GIS**

Las aplicaciones híbridas son nuevos servicios construidos en base al código de una o dos aplicaciones que se combinan, el producto de esta sinergia resultan proyectos interesantes, uno de estos proyectos interesantes es Google Maps, que desde que salió a la luz en el 2005 superó con creces a los actuales sistemas de información geográfica.

Google Maps es un servicio web como MapQuest, Yahoo Maps, no obstante la calidad de la presentación y el detalle de las capas que ofrece Google Maps es mucho más superior que otros incontables servicios que están en la web.

## **Google Maps Mashups**

Google Maps ofrece un soporte de comunicación vía ajax mediante archivos XML, que permiten que el uso del mapa sea agradable y la presentación de las imágenes sean casi automáticas a medida que el usuario navega a través del mapa. Además Google Maps ofrece toda una plataforma para implementar aplicaciones híbridas, las cuales son llamadas Google Maps Mashups que son la combinación de la plataforma GIS de Google Maps y la data geo-espacial que proveen los programadores, estas aplicaciones pueden ser de todo tipo y a la vez muy útiles [chicagocrime.org](http://chicagocrime.org), [housingmaps.com](http://housingmaps.com), entre otras.





Figura 1.8 CHICAGOCRIME.ORG, aplicación web basada en Google Mas donde es posible visualizar la incidencia criminal por ubicación geográfica

Este tipo de aplicaciones pueden desarrollarse en base al API que ofrece Google Maps, este API supe todos los requerimientos necesarios de un GIS, como el ubicar marcadores en el mapa, definir poli líneas, polígonos, definir capas de presentación por defecto y superponer imágenes. Generalmente los GIS comunes necesitan un soporte de base de datos para manejar esta información vectorial y así presentar todo este abanico de superposiciones, una aplicación basada en Google Maps, sólo necesita declarar estas superposiciones en simples instrucciones Javascript, que es el lenguaje de programación de Google Maps y el más usado por las interfaces clientes en la mayoría de los navegadores web.

### Utilidad para el proyecto de Tesis

La investigación realizada por [MILLER 06] brinda una afirmación sobre las ventajas de usar Google Maps Api como sistema de información geográfica para una aplicación web frente a otros sistemas, dada las funcionalidades como marcadores, polilíneas, polígonos y otros relacionados, invita a escoger a Google Maps como plataforma GIS para el presente proyecto de investigación aplicada. Así mismo expone aplicaciones web interesantes que han tenido como base Google Maps API que son usadas actualmente por cientos de personas en la red, lo que ilustra la robustez de este servicio que ofrece Google Inc. de manera gratuita.

### 1.3.2.3 Apparatus and method of map engine clustering automation

Li Liu registró una patente en el 2008 sobre un método para realizar agrupamiento de superposiciones en un mapa [LIU 08], en la descripción de la patente expone que diversas aplicaciones basadas en mashups en línea como Google Maps, Microsoft Virtual Earth, Yahoo! Earth y fuera de línea como Microsoft MapPoint, Google Earth y Microsoft Street Maps, presenta información en

posiciones geográficas como marcadores, lo cual permite a los usuarios visualizar información relevante o de interés en el mapa.

### **El problema**

El problema existe cuando en algunas aplicaciones basadas en mapas, es necesario lista una cantidad cuantiosa de marcadores en el mapa, y a raíz de eso produce un problema en el desempeño de la aplicación, afectando la usabilidad del mapa, agotando los recursos del computador cliente y degradando la experiencia del usuario. Esto se debe a que los GIS generalmente se ejecutan en el computador del cliente, por ejemplo Google Maps es una aplicación que si bien el servidor de mapas lo proporciona la infraestructura de Google, la aplicación en se desempeña en una página HTML usando a Javascript como lenguaje de programación y usa los recursos de memoria y procesamiento del computador del cliente. Consecuentemente es necesario incrementar los recursos de hardware del computador para poder presentar la gran cantidad de marcadores en el mapa.

### **La solución**

En general, es necesaria una disminución de marcadores basados en el nivel de zoom y usando técnicas apropiadas para agrupar marcadores automáticamente y así incrementar el desempeño, usabilidad y experiencia del usuario. Una personificación de esta solución es un método de agrupamiento automático de marcadores de un mapa, el método está comprendido de lo siguiente:

1. Un motor de agrupamiento de marcadores automático (MECA), que es un módulo de comunicación con el motor del mapa para recibir el último contenido del mapa del motor del mapa. El módulo de agrupamiento MECA, obtiene el último contenido del motor del mapa y reemplaza el contenido del recibido por un contenido que presenta agrupamiento de marcadores basados en su proximidad y su nivel de zoom.
2. Un módulo de comunicación cliente que se conecta con el módulo MECA para recibir el último contenido agrupado, el módulo de comunicación actualiza el mapa con los marcadores agrupados.

Sobre el método de agrupamiento de marcadores en el mapa, el método comprende los siguientes pasos:

1. Recibir los marcadores del motor de mapas.
2. Recuperar una lista de publicaciones para procesar en relación a la información del mapa.
3. Filtrar la lista de publicaciones basados en un criterio, de modo que las publicaciones que coinciden con el criterio son filtrados de acuerdo con la operación de filtrado, resultando un mapa filtrado de marcadores correspondientes a las publicaciones cuales fueron filtrados y una primera lista de marcadores restantes que no fueron filtrados.
4. Agrupando la primera lista de publicaciones restantes de modo que las publicaciones que están substancialmente próximas son agrupadas en función a una operación de agrupamiento resultando un mapa de marcadores agrupados correspondiente a las publicaciones donde fueron agrupadas y una segunda lista de agrupaciones que no fueron agrupadas.

- Proporcionar una lista de marcadores y publicar su información relacionada para ser mostrada en el mapa.

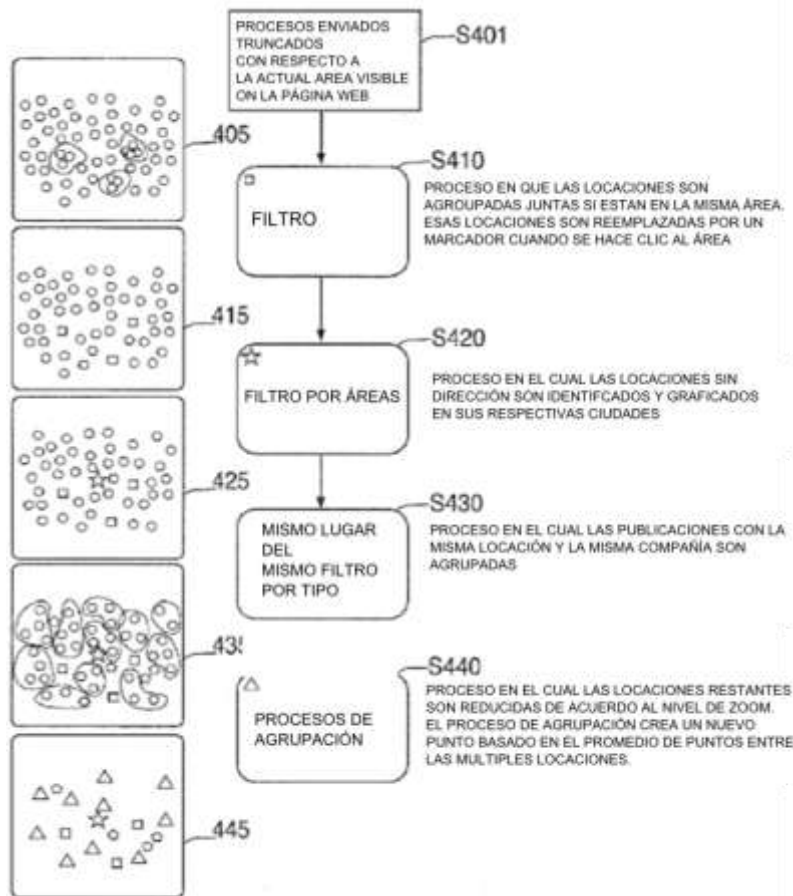


Imagen obtenida de [MILLER 06]

Figura 1.9 Proceso de agrupamiento de marcadores

### Utilidad para el proyecto de Tesis

La solución propuesta por Li Liu es un aporte muy importante para el presente proyecto de investigación, debido que a que los mapas de monitoreo vehicular generalmente presentan cientos de unidades vehiculares en el mapa, tomando similitud con Li Liu serían marcadores, de esta forma en un área del mapa donde existan muchas unidades próximas con un nivel de zoom se aplicaría la solución propuesta.

En la actualidad existe una solución llamada Marker Clusterer que es una implementación en javascript del aporte hecho por Li Liu para la versión 3 del API de Google Maps, esta herramienta será de suma utilidad para la presente investigación aplicada.

### 1.3.2.4 Location identification and vehicle tracking using VANET (VETRAC)

Arunkumar Thangavelú [THANGAVELU 07] expone que las ciudades que poseen vehículos requieren un sistema para determinar la ubicación y el movimiento de los vehículos a un momento dado. El sistema de monitoreo vehicular VETRAC permite a los conductores de los vehículos y a terceras personas rastrear la ubicación de cualquier vehículo en movimiento. La mayoría de los sistemas de monitoreo vehicular utilizan receptores GPS para transmitir su posición, otros sistemas de monitoreo vehicular utilizan dispositivos receptores híbridos que combinan la tecnología celular y satelital a fin de ahorrar costos. VETRAC utiliza dispositivos WiFi IEEE 802.11g (WIFI) para transmitir la ubicación del vehículo y la información puede ser vista en mapas electrónicos usados en internet o software especializado.

La ventaja de un dispositivo WIFI frente a un dispositivo GPS, es que un dispositivo GPS no puede transmitir en un túnel o en grandes áreas cerradas, perdiéndose la información transmitida. La meta del proyecto VETRAC es cualquier vehículo pueda ser ubicado en una ciudad pueda transmitir aprovechando redes WIFI pre-configuradas en los principales lugares públicos y usar estas redes para enviar transmisiones a bajo costo, así mismo serviría como sistema de navegación a fin de informar la situación del tráfico de manera que los conductores se informen de manera correcta.

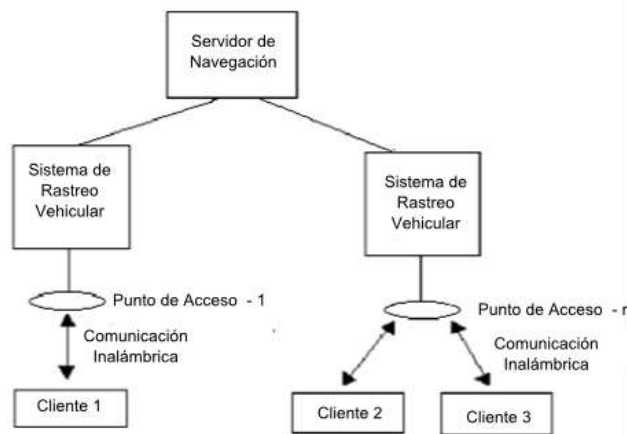


Imagen obtenida de [THANGAVELU 07]

**Figura 1.10 Sistema de monitoreo vehicular VETRAC**

### Motivación del Proyecto

La motivación del proyecto VETRAC es:

- ✓ Minimizar la congestión vehicular entregando información oportuna para evitar el tráfico a través de un sistema de navegación basado en rutas múltiples.
- ✓ Proveer una solución inteligente para identificar rutas óptimas para el conductor.

- ✓ Proveer información basado en video sobre el comportamiento del tráfico, siendo la implementación viable dado que la transmisión de la información mediante redes WIFI es económica.
- ✓ Identificar la posición del vehículo e identificar la entrada o salida a nuevas zonas.
- ✓ Proporcionar comunicación por conferencia a través de las redes WIFI para los conductores.

El sistema VETRAC trabaja bajo el principio de identificar un vehículo, al cual se le asigna la una dirección IP móvil. El sistema rastrea la posición del vehículo a través de las redes WIFI que se conectan con el servidor del sistema que se comunica con los dispositivos instalados en los vehículos. El sistema provee un panel de control donde el usuario puede identificar la posición actual del vehículo, la dirección del destino y la ruta recorrida y la distancia por recorrer.

### Arquitectura del Sistema

El diseño del sistema está organizado en distintos módulos que poseen funcionalidades específicas, el módulo principal de VETRAC es el sistema de monitoreo y el manejo de rutas.

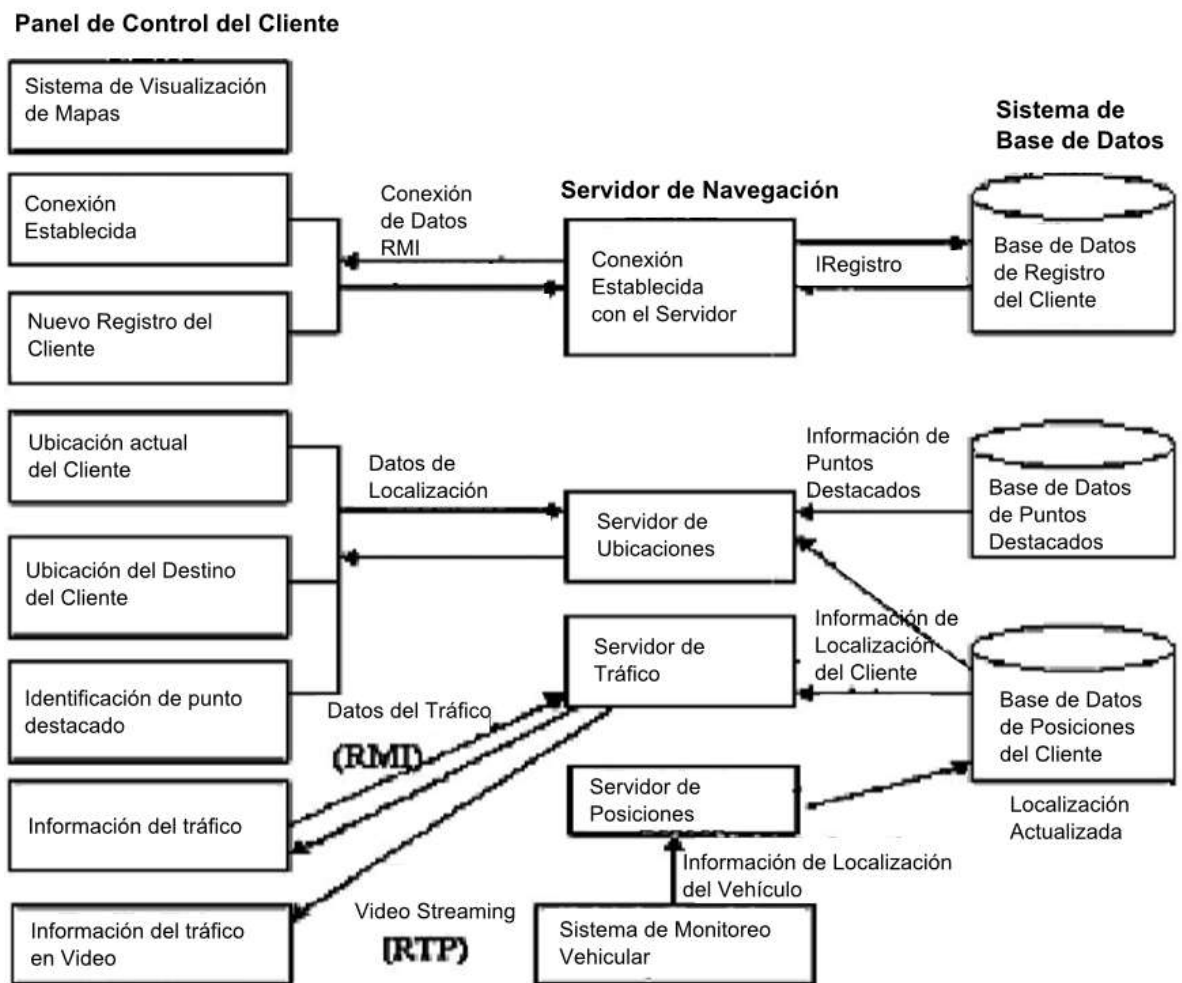


Imagen obtenida de [THANGAVELU 07]

**Figura 1.11 Arquitectura del sistema VETRAC, las funcionalidades están agrupadas por módulos**

- a) El servidor de navegación es la parte principal del sistema que permite manejar los requerimientos de los clientes junto con el sistema de monitoreo, el sistema establece conexión con el cliente, gestiona la información del tráfico y la transmisión de las conferencias.
- b) El sistema de monitoreo vehicular se conecta con redes WIFI que cubren un área determina y establecen conexión con el receptor WIFI, este sistema se conecta con servidor y actualiza la posición del vehículo en la base de datos.
- c) El panel de control del cliente que permite al usuario conocer la posición actual del vehículo, información del tráfico, visualización de video conferencias.
- d) El GIS donde el usuario puede visualizar la posición del vehículo en un mapa junto con las rutas recorridas.

### **Utilidad para el proyecto de Tesis**

El proyecto VETRAC es un ejemplo de sistema de monitoreo vehicular que pretende dar solución a diversos problemas vehiculares en una ciudad, así mismo la investigación hecha por Arunkumar Thangavelú, proporciona claramente la arquitectura de un sistema de monitoreo vehicular definiendo puntualmente las funcionalidades de cada módulo. Esta arquitectura será de inspiración para la implementación del presente proyecto de investigación aplicada, ya que módulos como la base de datos de posiciones, la base de datos de clientes, el servidor de posiciones, el panel de control junto con el GIS, serán considerados a fin de implementar el aporte del presente proyecto de investigación aplicada.

## 1.4 Objetivos del Proyecto

### 1.4.1 Marco Lógico

#### Árbol de Problemas

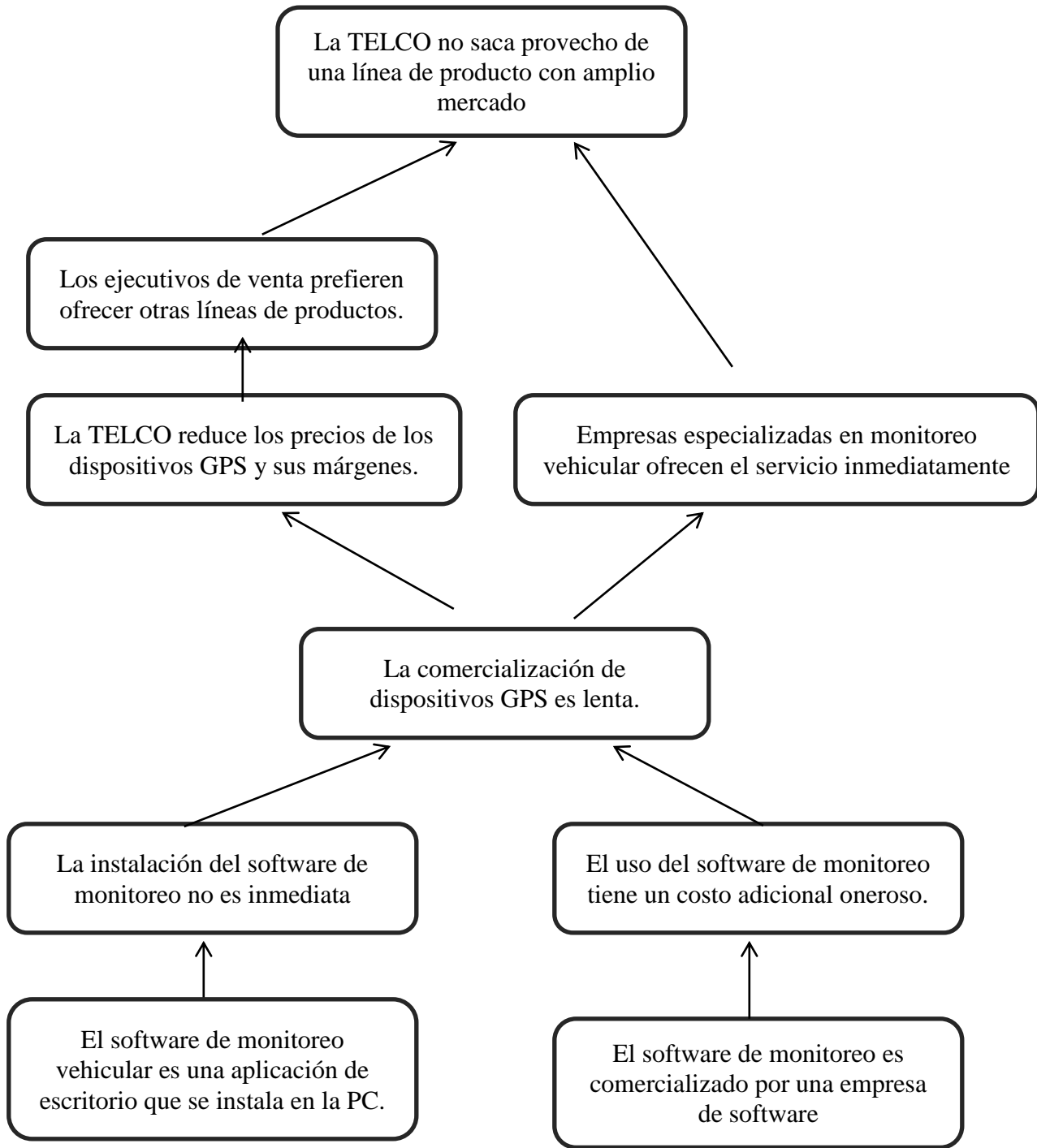


Figura 1.12 Árbol de Problemas

## Árbol de Objetivos

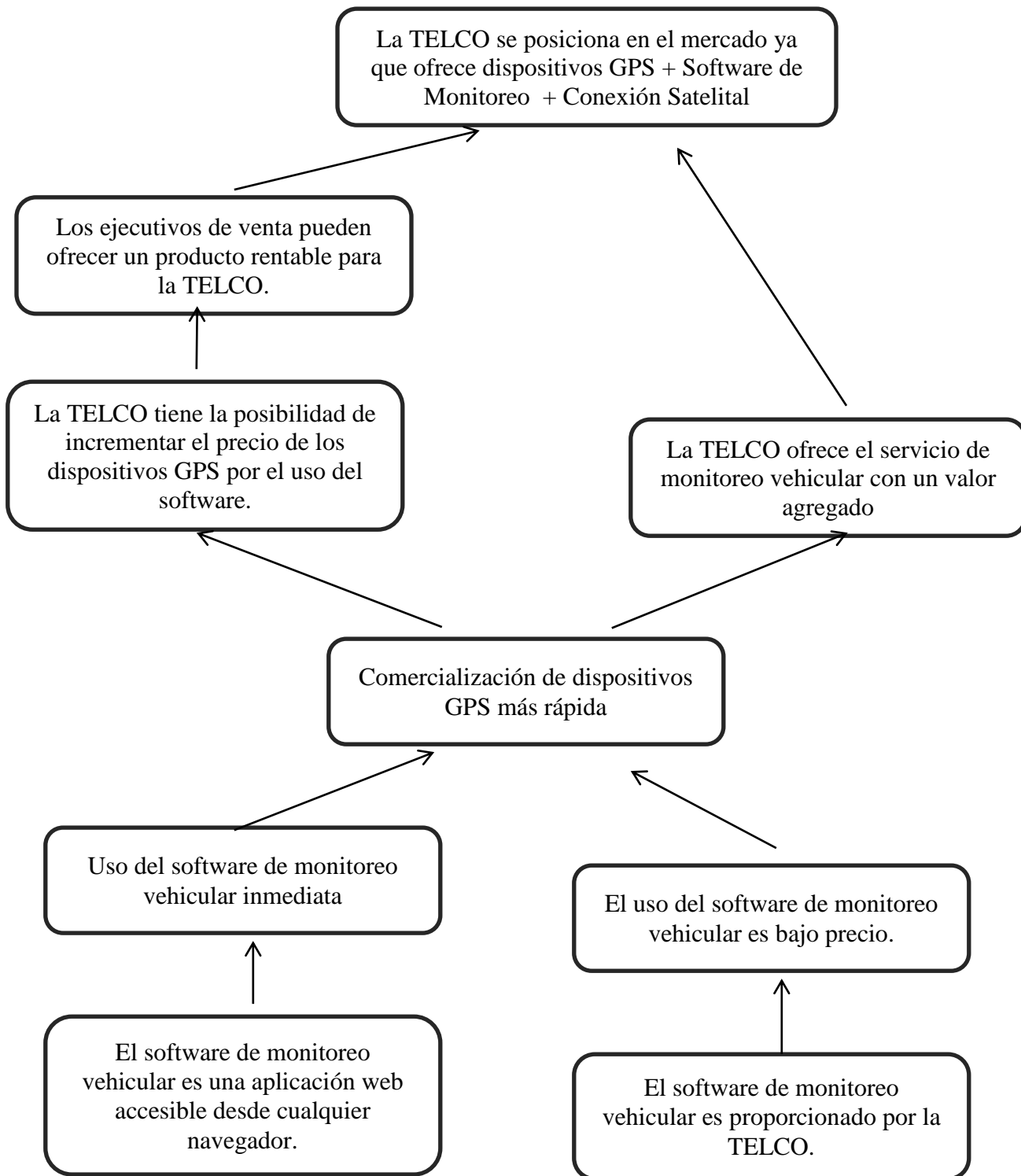


Figura 1.13 Árbol de Objetivos



### **1.4.2 Objetivo General**

El objetivo principal es desarrollar una aplicación web que realice el monitoreo de vehículos con dispositivos GPS y agilice la implantación del servicio en los clientes.

### **1.4.3 Objetivos Específicos**

- 1 Conocer los conceptos más importantes del proceso de monitoreo vehicular a fin de plasmarlos en una solución de software.
- 2 Conocer las funcionalidades más representativas de las aplicaciones de monitoreo en diversas industrias a fin de tener una fuente rica en requerimientos.
- 3 Modelar una solución web que cumpla con las exigencias más importantes del servicio de monitoreo vehicular.

## **1.5 Beneficios del Proyecto**

### **1.5.1 Beneficios Tangibles**

Permitirá la reducción de costos por parte de los usuarios en el monitoreo de sus vehículos ya que podrán hacer uso de esta nueva aplicación como un servicio adicional sin mayores complicaciones.

Así mismo permitirá atender la necesidad del mercado que requiere dispositivos GPS y un servicio de control sobre estos, por ejemplo según el Reglamento Nacional de Administración del Transporte aprobado por D.S. 017-2009 MTC exige que las unidades vehiculares de transportes interprovinciales utilicen dispositivos GPS. Según la SUTRAN (Superintendencia de Transporte Terrestre de Personas, Cargas y Mercancías) existen 6 632 buses en el Perú (año 2010).

### **1.5.2 Beneficios Intangibles**

La importancia de esta investigación radica en que implementando una aplicación web que monitoree vehículos con dispositivos GPS de manera satisfactoria, se entregaría una solución para diversos tipos de industria que requieran monitorear sus vehículos de forma eficiente y gozando de todos los beneficios que ofrece una aplicación basada en plataforma web (por ejemplo: uso de las funcionalidades del software desde distintos equipos conectados a la red, visualización de la información a través de un navegador web estándar, actualización del software en un solo equipo servidor que es replicado instantáneamente por todos los equipos clientes, etc.).

## **1.6 Alcance del Proyecto**

La finalidad de la presente investigación es desarrollar una aplicación web que sirva de herramienta de monitoreo de vehículos, el presente aporte se centrará en dicha premisa la cual enmarcará el presente aporte. En el monitoreo vehicular intervienen diversos procesos, desde la activación del dispositivo, la transmisión satelital, el envío de la ubicación a un almacén de datos y estos transportados a un aplicativo que utilice una plataforma GIS que permita analizar la información y monitorear el vehículo al usuario final, los principales procesos identificables son los siguientes:

- 1. Registro y activación del dispositivo GPS:** Este proceso se da cuando es instalado un dispositivo de rastreo en el vehículo, este dispositivo es seleccionado

acorde a las necesidades del cliente, así mismo este dispositivo es activado para que empiece a transmitir.

**2. Transmisión de la posición del vehículo:** Cuando el dispositivo está activado, este empieza a transmitir, puede transmitir usando la red satelital, la red celular o escoger la red según convenga, el tiempo de transmisión es determinado mediante la configuración del dispositivo.

**3. Estandarización de la información recibida y almacenamiento de una base de datos:** La TELCO recibe la información que envían los dispositivos GPS, sin embargo estos dispositivos generalmente no son los mismos, son de diferentes fabricantes, envían distintos datos, algunos datos puntuales por la capacidad limitada del hardware y otros envían gran cantidad de información por los sensores que poseen, esta información es recibida en diferentes formatos y tiene que ser estandarizada para su correcto entendimiento y posterior procesamiento.

**4. Recepción de la información estandarizada para el análisis del aplicativo de monitoreo vehicular:** Una vez que la información es estandarizada, a cada unidad de transmisión se le conoce con el nombre de “trama”, cada trama está comprendida básicamente por el identificador del dispositivo GPS, la fecha y hora de transmisión, la latitud y longitud, esta información es suficiente para que cualquier solución de monitoreo vehicular pueda analizar la información.

Esos 4 procesos básicos son los que engloban el servicio de monitoreo vehicular, no obstante la presente investigación aplicada se centrará en el último proceso, el cual es la “Recepción de la información estandarizada para el análisis del aplicativo de monitoreo vehicular”. Ya que en los 3 primeros procesos están más inclinados en atender asuntos relacionados a la transmisión de la información.

## **1.7 Conclusiones**

El conocimiento de todos los temas pertinentes al presente proyecto de investigación, desde el planteamiento de los objetivos, el marco teórico, estado del arte y el conocimiento de los beneficios, da un panorama claro de que el presente proyecto tendrá una utilidad inmediata para los usuarios que se beneficien con el producto software y abrirá nuevos nichos de mercado en comercialización de dispositivos GPS a la TELCO que es uno de sus principales intereses, sumando a esto la visión circunscribe el presente proyecto de investigación en el negocio de la TELCO de manera clara, presentando procesos clave donde se irán desarrollando los siguientes temas de la presente investigación.

## **CAPÍTULO II - MODELADO DEL NEGOCIO**

### **2.1 Introducción:**

El Modelado del Negocio es el punto de partida del proceso de desarrollo, a lo largo de la presente investigación aplicada se ha dado a entender que el fin es implementar una aplicación web de monitoreo vehicular, tomando eso en cuenta el Modelado del Negocio permitirá identificar los principales actores del negocio que hacen uso del servicio de monitoreo vehicular, así mismo podremos describir los principales casos de uso del negocio que le dan valor al servicio de monitoreo vehicular junto con las principales entidades tratadas en los casos de uso. Si bien es claro que las principales entidades son el dispositivo GPS, la unidad vehicular y la trama que representa la transmisión, al hacer un análisis más detallado de los casos de uso del negocio identificaremos más entidades que son de suma importancia junto con sus atributos. Producto del Modelo del Negocio se identificarán que actividades dentro de los casos de uso serán candidatas para ser requisitos de la nueva solución software a fin de dar paso al análisis de requisitos.

### **2.2 Reglas del Negocio:**

Regla 1: El Cliente deberá saber inmediatamente la posición y el estado de su unidad vehicular una vez solicitada.

Regla 2: Si el cliente desea tener un software de monitoreo, este deberá ser el software que la TELCO le proporcione, ya sea directamente o por medio de terceros.

Regla 3: La TELCO deberá registrar toda la información que el dispositivo transmita desde el inicio de su funcionamiento.

### **2.3 Actores y Trabajadores del Negocio**

#### **2.3.1 Actor del Negocio – El Cliente**

El cliente es el actor principal del negocio, ya que da inicio a los principales procesos como la solicitud del servicio de transmisión de datos, solicitud de atención del servicio de postventa y solicitar el software de monitoreo vehicular.

### 2.3.2 Trabajadores del Negocio

**Asesor Comercial:** Es el responsable de ofrecer los productos y servicios al Cliente, firmar acuerdos y contratos con el Cliente y gestionar todos los procesos donde puedan ocurrir una posible venta.

**Ingeniero de Soporte:** Es el responsable de instalar los equipos de localización en las unidades vehiculares que el Cliente haya seleccionado.

**Operario de Soporte:** Es el responsable de absolver todas las dudas o consultas que tenga el cliente posterior a la venta del servicio o producto, interviene principalmente en el servicio de post-venta.

**Empresa de Software:** En una empresa de implementación de soluciones informática que es contratada por la TELCO para representarla e instalar el software de monitoreo si el Cliente los solicita.

## 2.4 Casos de Uso del Negocio

### 2.4.1 Diagrama de Casos de Uso del negocio

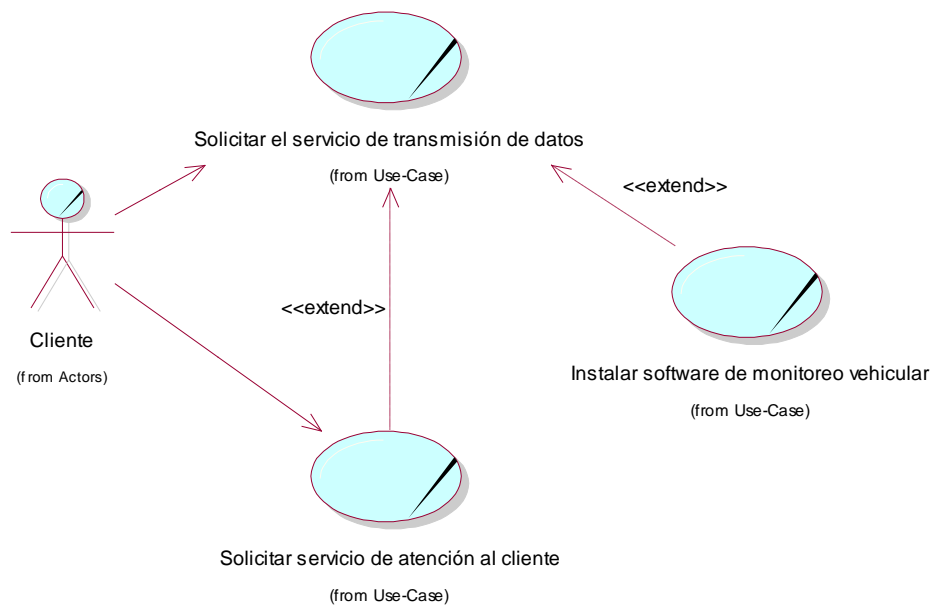


Figura 2.1 Diagrama de Casos de Uso del Negocio

### 2.4.2 Solicitar el servicio de transmisión

El Caso de Uso del Negocio “Solicitar el servicio de transmisión” describe el proceso inicial del negocio, ya que se enfoca en el negocio principal de la TELCO que es ofrecer el servicio de transmisión de datos, en el presente caso de uso el Cliente solicita el servicio de transmisión de datos y así mismo un equipo de localización que sirva para poder ubicar su unidad vehicular.

## Flujo Básico

El Caso de Uso del Negocio se da inicio cuando el Cliente solicita el servicio de transmisión de datos a fin de poder monitorear su unidad vehicular, seguidamente el Asesor Comercial le presenta los diversos modelos de equipos de localización, una vez que el Cliente haya seleccionada el equipo, el Asesor Comercial le presenta los planes tarifarios asociados al equipo, ya que el equipo puede ofrecer distintas prestaciones, una vez que el Cliente haya seleccionado el plan tarifario el Ingeniero de Soporte instala el equipo en la unidad vehicular del Cliente, activa el dispositivo y finalmente el Cliente da su aprobación del servicio.

## Diagrama de Actividades

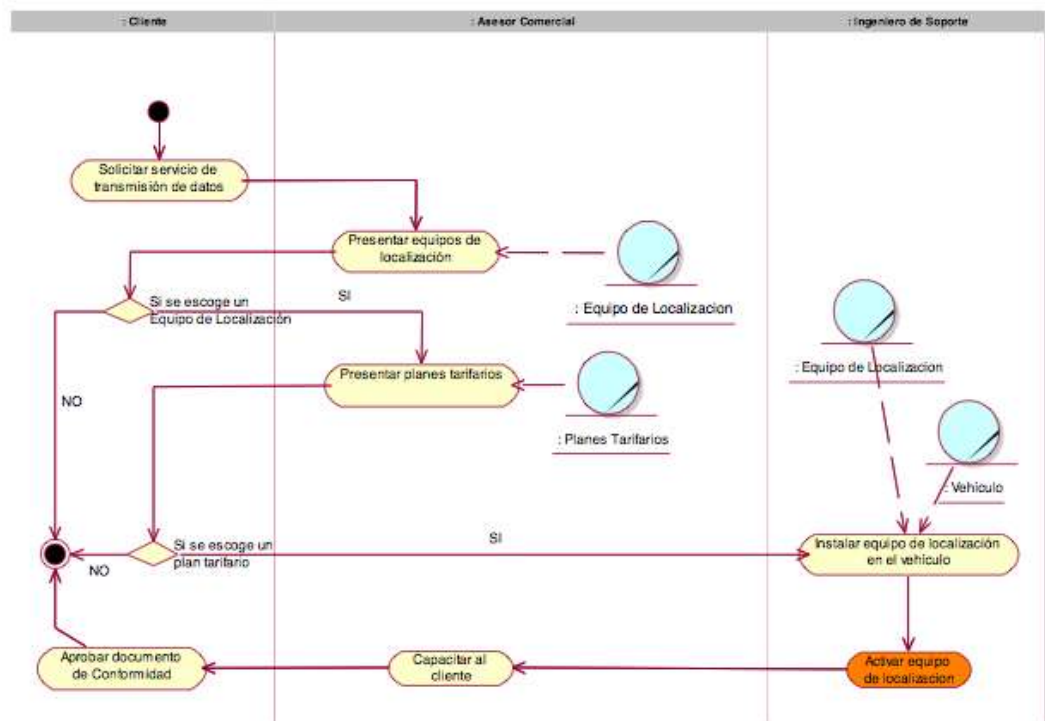


Figura 2.2 CUN – Solicitar el servicio de transmisión de datos – Diagrama de Actividades - Flujo Básico

### 2.4.3 Instalar el software de monitoreo vehicular

Generalmente el Cliente desea poseer un equipo de localización en sus unidades vehiculares para cuando suceda una emergencia, en ese caso llama a la TELCO para saber la posición de su unidad vehicular, estas situaciones pueden ser muy escasas, pero se da el caso en que el Cliente desea poseer una herramienta que le permita visualizar donde está su unidad vehicular sin tener que consultar a la TELCO, bajo esa premisa el Cliente solicita a un Asesor Comercial la instalación del software de monitoreo en sus

instalaciones, para eso el Asesor Comercial contacta con una Empresa de Software que en representación de la TELCO instala el software de monitoreo en la PC del Cliente.

### Flujo Básico

El Caso de Uso del Negocio se inicia cuando el Cliente solicita al Asesor Comercial la instalación del software de monitoreo vehicular en sus instalaciones, el Asesor Comercial contacta con la Empresa de Software, la Empresa de Software en representación de la TELCO instala el software en la PC del Cliente, posterior a esto el Cliente firma un documento de conformidad por el servicio.

### Diagrama de Actividades

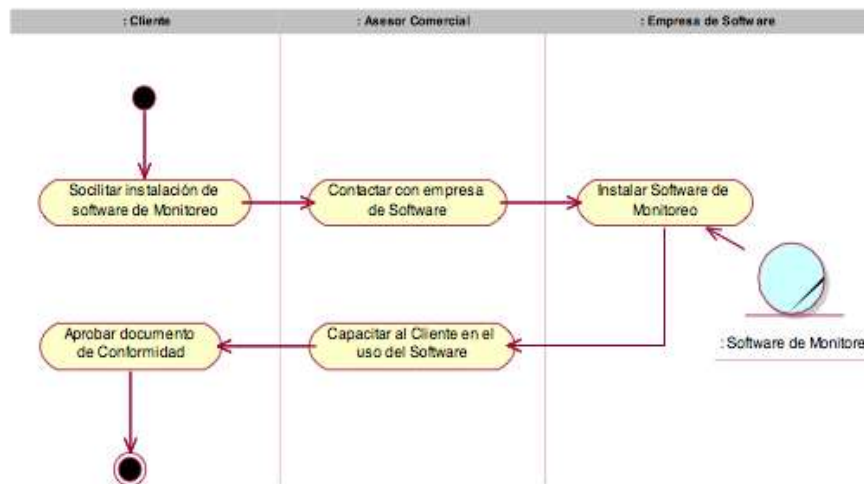


Figura 2.3 CUN – Instalar software de monitoreo vehicular – Diagrama de Actividades - Flujo Básico

#### 2.4.4 Solicitar el servicio de atención al cliente

Como mencionamos anteriormente el Cliente solicita el servicio de postventa a fin de saber dónde se encuentra su unidad vehicular, el Operario de Soporte que tiene una instalación del software de monitoreo le comunica donde se encuentra su vehículo y en el detalle de la información que transmite el equipo de localización, también se puede dar el caso de que el Cliente solicite enviar un comando al equipo para que interactúe directamente con el vehículo, por ejemplo: El apagado del motor o el abrir un compartimiento especial de la unidad vehicular, este caso se da en unidades vehiculares que transportan valores muy importantes.

#### Flujo Básico – Solicitar posición

El Cliente solicita el servicio de atención al cliente y un Operario de Soporte se contacta inmediatamente con el Cliente, este le solicita la posición de su unidad vehicular y el Operario de Soporte consulta el software de monitoreo vehicular y le comunica telefónicamente donde se encuentra su unidad vehicular y el estado de esta.

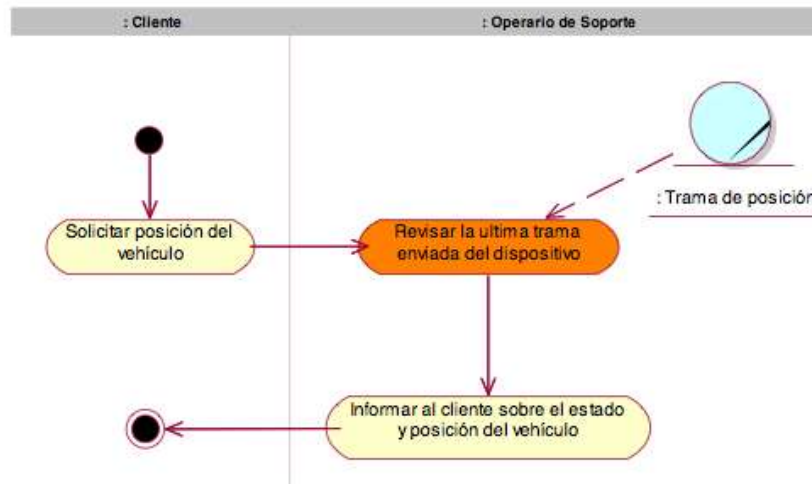


Figura 2.4 CUN – Solicitar el servicio de atención al cliente – Diagrama de Actividades - Flujo Básico

### Flujo Alternativo – Enviar Comando

El Cliente solicita el servicio de atención al cliente y un Operario de Soporte se contacta inmediatamente con el Cliente, este le solicita enviar un comando al equipo de localización, el Operario de Soporte revisa la última trama emitida del equipo de localización y solicita la confirmación del Cliente, este confirma el envío del comando y el Operario de Soporte envía el comando al equipo de localización.

### Diagrama de Actividades

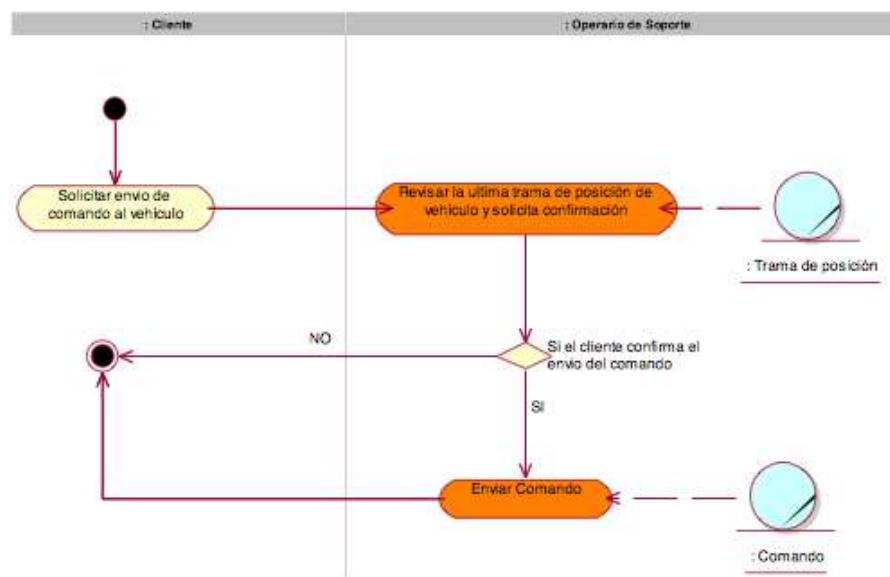


Figura 2.5 CUN – Solicitar el servicio de atención al cliente – Diagrama de Actividades - Flujo Alternativo

## 2.4.5 Actividades candidatas a automatizar

En los diagramas de actividades presentados se han apreciado actividades con sombra naranja, estas actividades son candidatas como requerimientos del sistema que tendrá la solución software a desarrollar en la presente investigación. Estas actividades son:

- 1 Revisar la última trama enviada del dispositivo.
- 2 Solicitar confirmación de enviar comando.
- 3 Enviar Comando

## 2.4.6 Diagrama de Entidades del Negocio

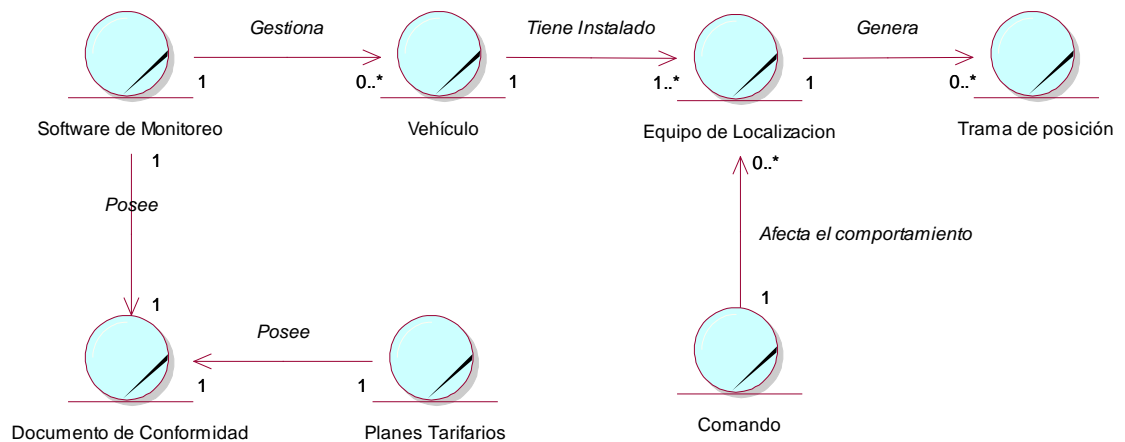


Figura 2.6 Diagrama de Entidades del Negocio

A continuación, una descripción de cada entidad:

**Software de Monitoreo:** Es la solución software que ofrece la Empresa de Software que representa a la TELCO, esta herramienta permite localizar y enviar comandos a los equipos de localización.

**Vehículo:** Es la unidad vehicular que tiene instalado el equipo de localización y es rastreada por este. Posteriormente esta entidad será conocida por “unidad” o “unidad vehicular”.

**Equipo de localización:** Es el equipo que transmite datos de la ubicación de la unidad vehicular, este equipo puede interactuar directamente con la unidad a través del envío de comandos. Posteriormente esta entidad será conocida por “Card”.

**Trama de posición:** Es el conjunto de datos que son enviados por una transmisión del equipo, este consta de la latitud, longitud, velocidad, fecha, entre otros datos. Esta entidad será conocida posteriormente por el nombre de “CardData” o “Trama”.



**Comando:** Es la instrucción que influye el comportamiento del equipo de localización, este Comando está compuesto por parámetros y necesita una confirmación por parte del Cliente para ser enviado.

**Planes Tarifarios:** Esta entidad representa el conjunto de condiciones y normas a la que estarán sujetos los servicios que utilizará el Cliente con el equipo de localización y su transmisión de datos.

**Documento de Conformidad:** Es el documento que certifica que el Cliente tiene los servicios y productos adecuadamente instalados según fue solicitado.

## 2.5 Conclusiones

Del Modelado del Negocio se ha obtenido los procesos en los cuales el producto software aportará en gran manera, se identifican actividades clave en el negocio que involucran el servicio de monitoreo vehicular, los principales trabajadores del negocio y las entidades que son la materia prima para el resto del modelado del sistema. Si bien el monitoreo vehicular como servicio tiene varias décadas en las industrias de minería, aviación y entre otros, los procesos del negocio son los mismos y los requerimientos son bastante puntuales, no obstante estos deben de responder de manera efectiva a las reglas del negocio que se presenta, en este caso son 3 reglas del negocio puntuales que permitirán a la TELCO mantener una relación directa con sus clientes de la mano de una herramienta que es de gran utilidad para estos ya que optimizará grandemente el servicio que ahora reciben.

## **CAPÍTULO III - REQUERIMIENTOS DEL PROYECTO**

### **3.1 Introducción:**

Los requerimientos del proyectos estarán comprendidos en primer lugar por el modelado de casos de uso del sistema junto con la identificación y descripción de los actores del sistema, este modelo de casos de uso de sistema en primer lugar será producto del modelado del negocio, así mismo se incluirán requisitos que se identificaron como aporte a la presente investigación en el estado del arte.

En el Análisis de los Requisitos definiremos que es lo que se propone como solución software desde el punto de vista funcional, también mencionaremos características no funcionales de la solución, como el tiempo de respuesta y las características de comportamiento de la interfaz de usuario. Producto del análisis de los requerimientos tendremos la especificación de los casos de uso del sistema junto con los prototipos del sistema, los cuales servirán de fundamento para el Análisis del Sistema.

### **3.2 Requerimientos del Software**

#### **3.2.1 Requerimientos Funcionales**

1. Desde una sola consola, el cliente podrá ver sus unidades, visualizar su recorrido, enviar comandos y atender alertas.
2. Seleccionando la unidad vehicular el cliente podrá ver el estado y la posición de la unidad.
3. El cliente podrá centrar el mapa solamente introduciendo la placa del vehículo en el panel de búsqueda. Así mismo podrá centrar el mapa en función a un punto de interés.
4. El cliente podrá ver todos los puntos de interés categorizados en el mapa seleccionándolos en el panel de búsqueda.
5. El Administrador del sistema podrá visualizar rápidamente las unidades de cualquier cliente, solo deberá ingresar la Razón Social del cliente, y el mapa inmediatamente actualizará las unidades vehiculares de este.

6. El Administrador del sistema podrá registrar, actualizar y eliminar la información de los modelos de los equipos de localización así mismo podrá asociar comandos a los modelos de los equipos
7. El Administrador del sistema podrá ingresar nuevos comandos al sistema indicando sus argumentos
8. El Administrador del sistema podrá ingresar y actualizar nuevos eventos al sistema
9. El Administrador del sistema podrá ingresar nuevas categorías o tipos de puntos de interés al sistema
10. El Administrador del sistema podrá registrar nuevos clientes al sistema y dar mantenimiento a su información.
11. El Administrador del sistema podrá registrar nuevos equipos de localización y dar mantenimiento a la información de los equipos de localización ya existentes.
12. El Administrador del sistema y el cliente podrá registrar nuevas unidades vehiculares y asociarles un equipo de localización, así mismo podrá ingresar información relacionada a la unidad.
13. El Administrador del sistema y el cliente podrá crear grupos de unidades o flotas.
14. El Administrador del sistema y el cliente podrá ingresar conductores y asociarlos a unidades vehiculares.
15. El Administrador del sistema y el cliente podrá registrar geocercas o zonas áreas de tránsito, estas podrán ser de control de velocidad, de exclusividad de movimiento o prohibidas, así mismo, estas geocercas deberán ser asociadas a las flotas a fin de tener efecto sobre las unidades vehiculares.
16. El Administrador del sistema y el cliente podrán ingresar nuevos puntos de interés al sistema, indicando información relevante al punto de interés y su posición geográfica.
17. El operario de soporte y el cliente podrán generar un reporte tabular de recorrido de una unidad vehicular en un periodo de tiempo dado.
18. El operario de soporte y el cliente podrán generar un reporte tabular de alertas de la unidad vehicular en un periodo de tiempo dado.
19. El operario de soporte y el cliente podrán generar un reporte tabular de comandos enviados a la unidad vehicular en un periodo de tiempo dado.
20. El usuario del sistema podrá actualizar su información personal, información de acceso y de contacto.
21. El usuario deberá ingresar al sistema a través de un formulario de autenticación, donde ingresará su nombre de usuario y contraseña de acceso.
22. El usuario podrá recuperar su contraseña a través de un formulario de recuperación, ingresando su correo electrónico previamente registrado.
23. El usuario podrá eliminar la sesión activa dentro del sistema.
24. El administrador del sistema podrá ingresar nuevos usuarios del sistema e ingresar su información relevante.
25. El administrador del sistema podrá ingresar nuevos roles al sistema, y asociarle permisos de funcionalidades a cada rol del sistema.

### 3.2.2 Requerimientos no Funcionales

#### Desempeño:

1. El tiempo de la presentación de la consola web para el monitoreo vehicular no deberá de exceder los 6 segundos.
2. El tiempo para la presentación del detalle del estado de unidad vehicular no deberá exceder los 4 segundos.
3. La presentación del recorrido (30 últimas posiciones) de una unidad vehicular no deberá exceder los 4 segundos.
4. La búsqueda de una unidad vehicular dentro de la consola web, deberá ser inmediata.
5. El centrar una unidad vehicular después de una búsqueda no deberá exceder los 2 segundos.
6. El listar registros de cualquier tipo de mantenimiento no deberá exceder los 4 segundos.
7. El registrar o actualizar cualquier registro del sistema no deberá exceder los 6 segundos.

#### Usabilidad:

1. El sistema en general deberá ser accesible desde los siguientes navegadores: Firefox 6.\* +, Google Chrome 11 +, Safari 5 +.
2. La consola web de monitoreo deberá ser comprendida en una sola pantalla, donde el usuario podrá tener acceso a las funcionalidades más importantes de monitoreo de manera inmediata.
3. El diseño web del sistema deberá ser minimalista y ocupar al máximo las dimensiones del monitor.

#### Disponibilidad:

1. El sistema podrá ser accedido las 24 horas del día, los 365 días del año, desde cualquier navegador web anteriormente mencionado.
2. La información de recorrido de las unidades vehiculares tendrá como máximo 1 mes de antigüedad a fin de garantizar un correcto desempeño de la Base de Datos del sistema.

#### Seguridad:

1. Cualquier usuario que desee tener uso del sistema de monitoreo, deberá identificarse mediante su nombre de usuario y contraseña, su nombre de usuario deberá ser un correo electrónico válido.
2. No podrá existir otro medio de acceder a la información que no sea mediante el formulario de autenticación y/o el manejador de la base de datos.

### 3.3 Casos de Uso del Sistema

#### 3.3.1 Diagrama de Actores del Sistema

**Asesor Comercial:** Este actor del sistema representa al trabajador del negocio “Asesor Comercial” cuyo papel en los casos de uso del sistema es acceder al mapa web y demostrar las unidades vehiculares siendo monitoreadas por la solución software.

**Operario de Soporte:** Este actor del sistema representa al trabajador del negocio “Operario de Soporte”, cuyo papel en los casos de uso del sistema es monitorear y interactuar con las unidades vehiculares a pedido del cliente.

**Cliente:** El actor Cliente representa al actor del negocio “Cliente”, cuyo papel en el sistema es de monitorear sus propias unidades vehiculares.

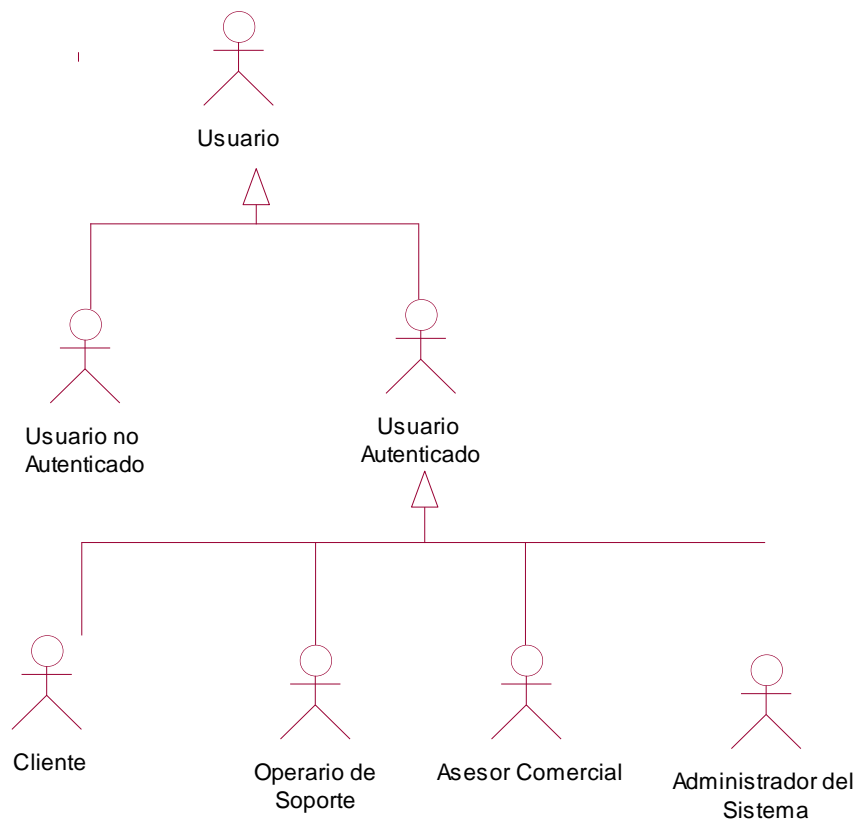
**Administrador del Sistema:** El administrador del Sistema puede acceder a toda la información generada en el sistema y ejecutar tareas de mantenimiento y configuración en la solución software.

**Usuario:** Representa a los actores del sistema en general que utilizan alguna funcionalidad del sistema.

**Usuario Autenticado:** Representa a los usuarios del sistema que han sido identificados y validados por un método de autenticación y pueden acceder a las funcionalidades que requieren identificación confirmada del usuario.

**Usuario No Autenticado:** Representa a los usuarios del sistema que no han sido autenticados por el sistema y solo pueden acceder a funcionalidades que no requieren identificación confirmada.

A continuación en la siguiente figura se presenta el diagrama de actores del sistema, donde se puede apreciar que los actores principales del sistema heredan funcionalidades del actor “Usuario Autenticado” dando entender que la mayoría de las funcionalidades del sistema necesitan una identificación confirmada, apoyando la afirmación inicial del diagrama de paquetes del sistema.



**Figura 3.1 Diagrama de actores del Sistema**

### 3.3.2 Diagrama de Paquetes

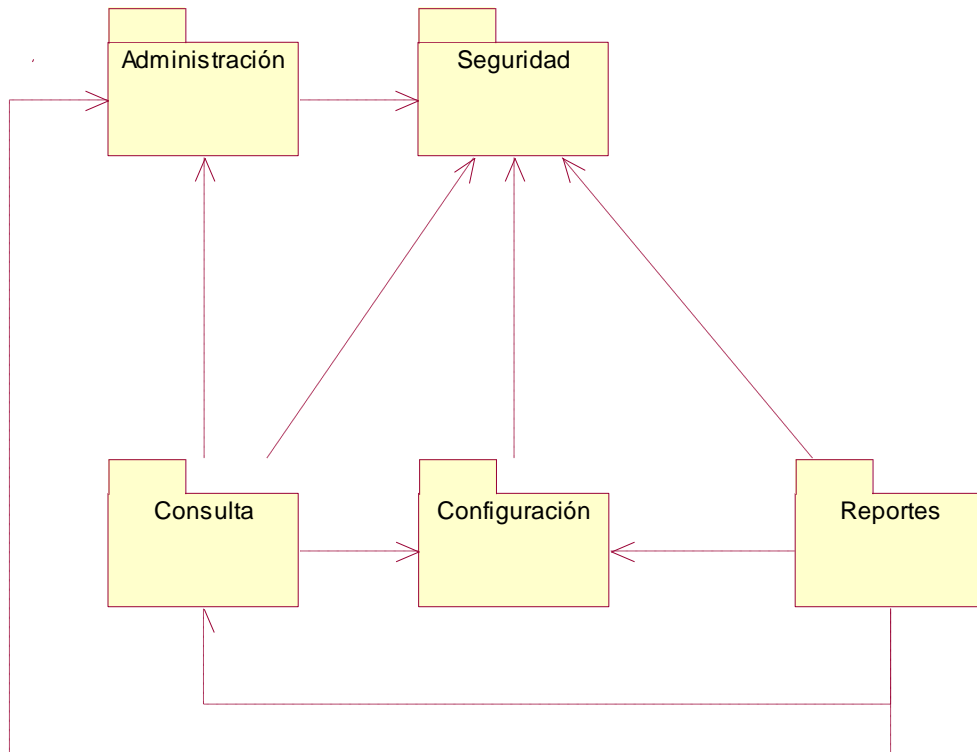


Figura 3.2 Diagrama de Paquetes del Sistema

Como podemos apreciar existe una relación evidente entre los paquetes del sistema, por ejemplo en el diagrama de paquetes se observa que los paquetes de Administración, Consulta, Configuración y Reportes dependen del paquete de Seguridad, dando a entender que la mayoría de las funcionalidades pasan por un proceso de autenticación.

## Diagrama de Casos de Uso del paquete Consulta

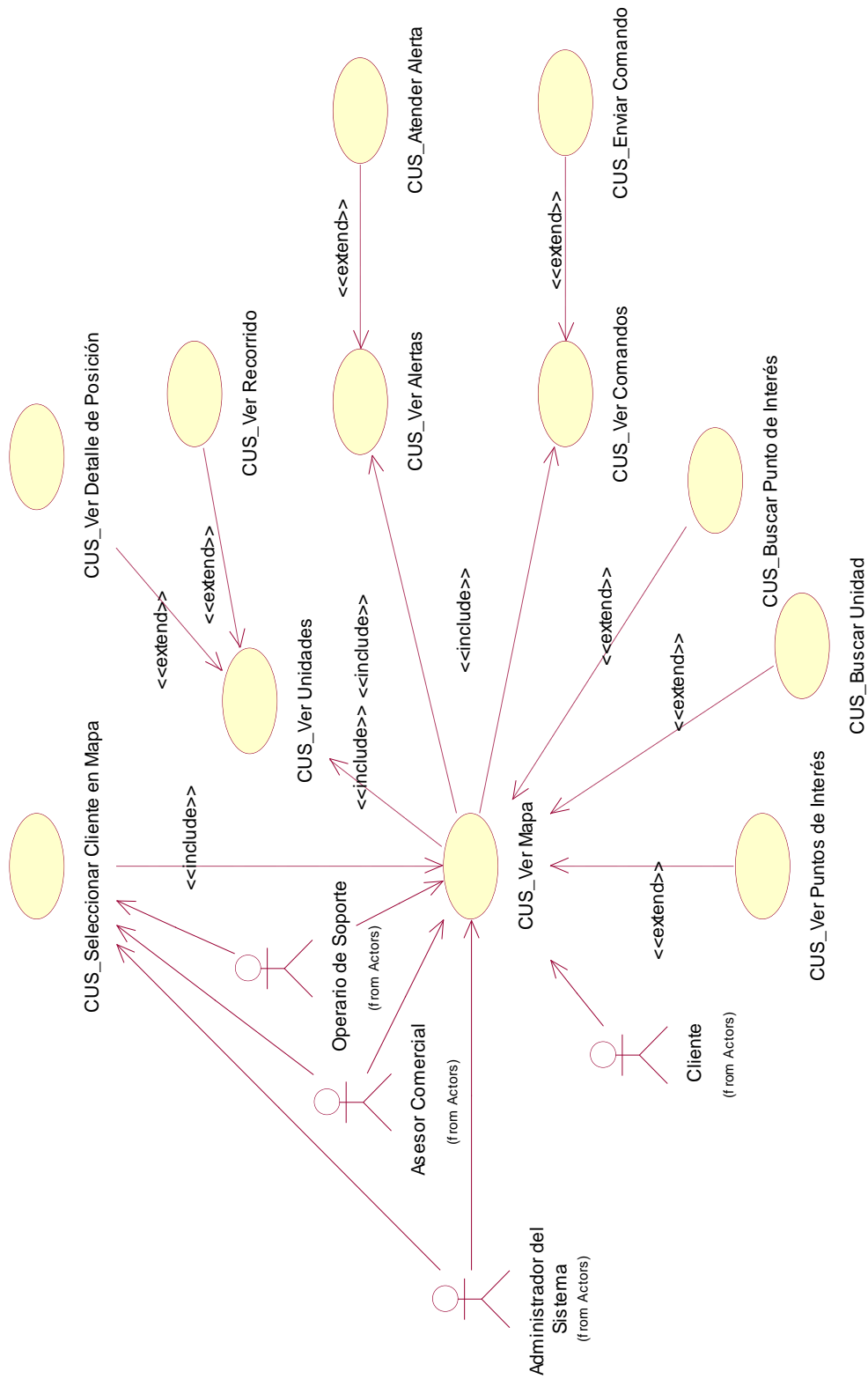


Figura 3.3 Casos de Uso del Paquete “Consulta”

## Diagrama de Casos de Uso del paquete Configuración

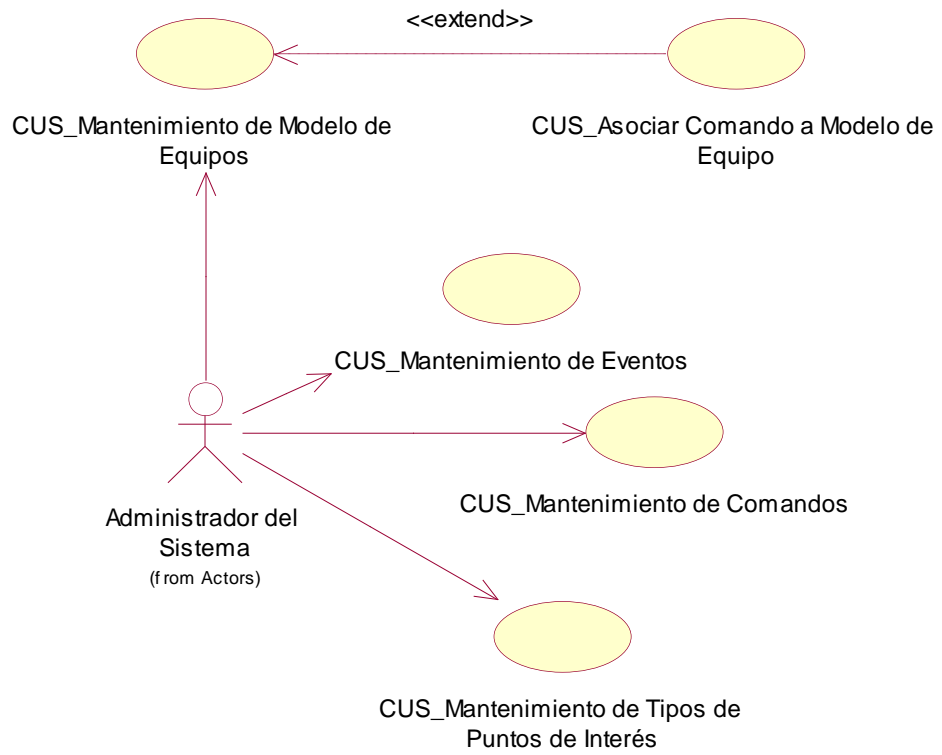


Figura 3.4 Casos de Uso del Paquete “Configuración”



### Diagrama de Casos de Uso del paquete Administración

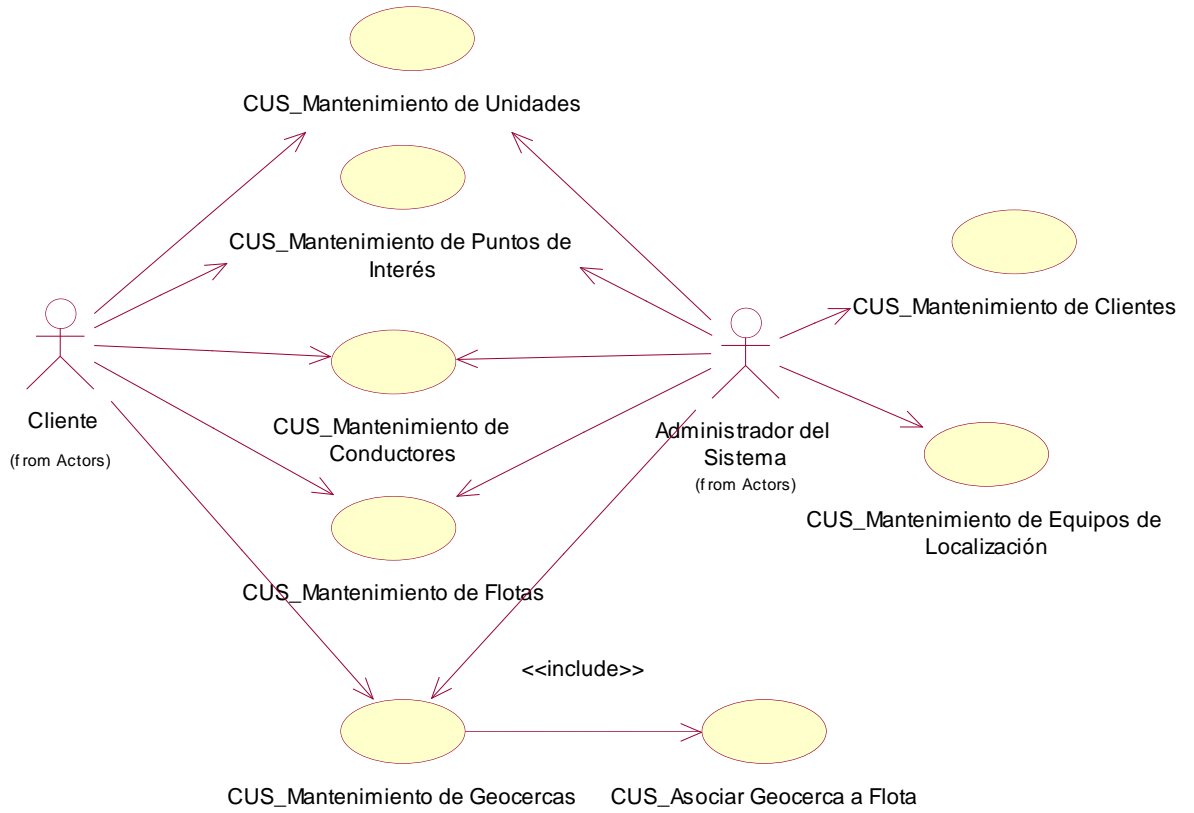


Figura 3.5 Casos de Uso del Paquete “Administración”

### Diagrama de Casos de Uso del paquete Reportes

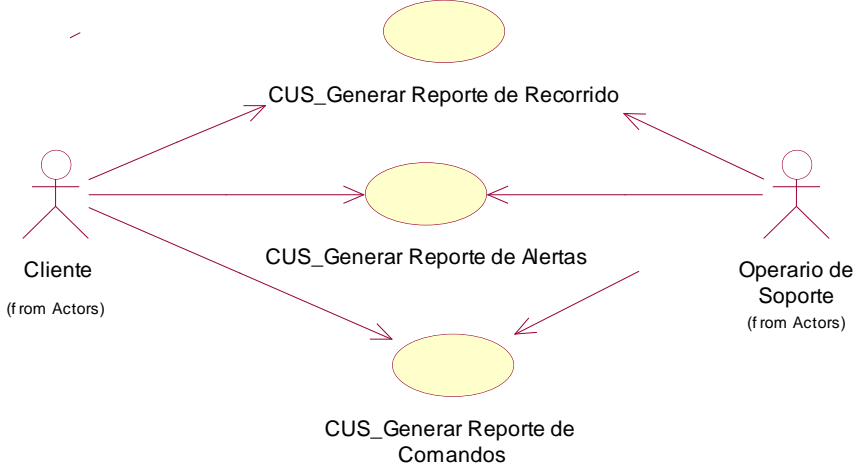


Figura 3.6 Casos de Uso del Paquete “Reportes”

## Diagrama de Casos de Uso del paquete Seguridad

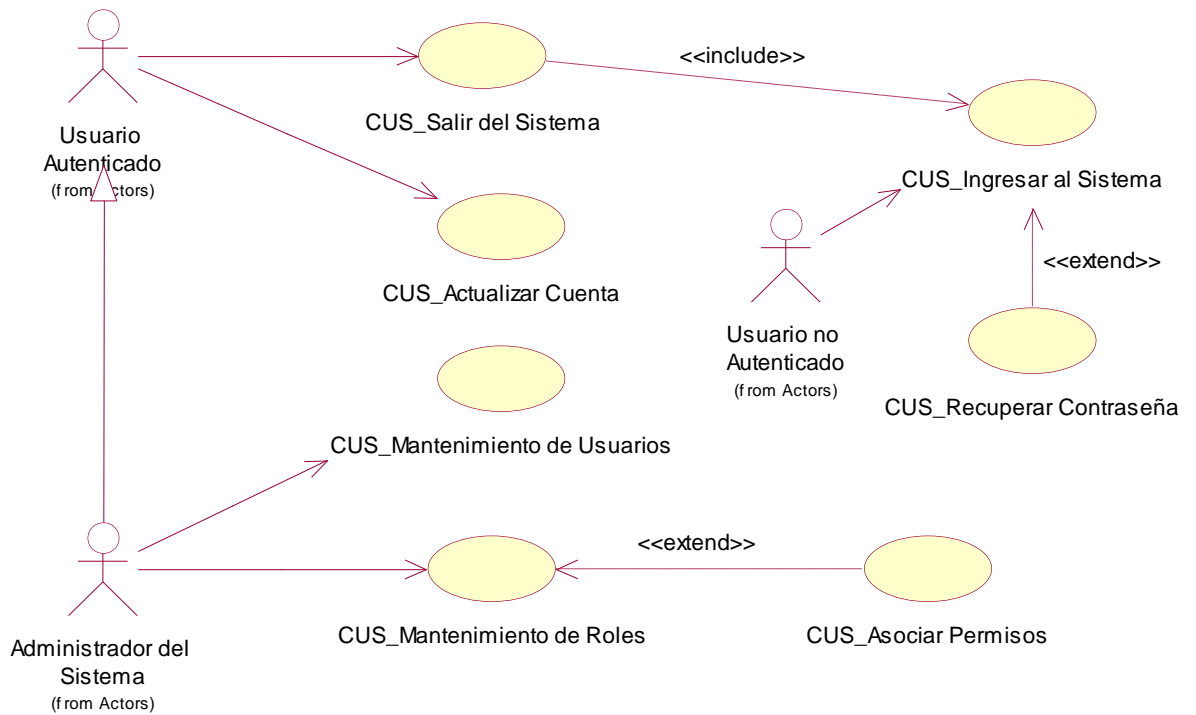


Figura 3.7 Casos de Uso del Paquete “Seguridad”

### Arquitectura de los Casos de Uso del Sistema

El fin de la Arquitectura de los Casos de Uso del Sistema es identificar los casos de uso que son más significantes para la solución software, como podemos apreciar todos los casos de uso del sistema excepto el caso de uso “CUS\_Mantenimiento de Equipos de Localización” y “CUS\_Mantenimiento de Geocercas” pertenecen al paquete de “Consulta”, ya que este paquete agrupa las funcionalidades más representativas que tendrá el sistema. En el siguiente punto describiremos a los casos de uso del sistema por paquete.

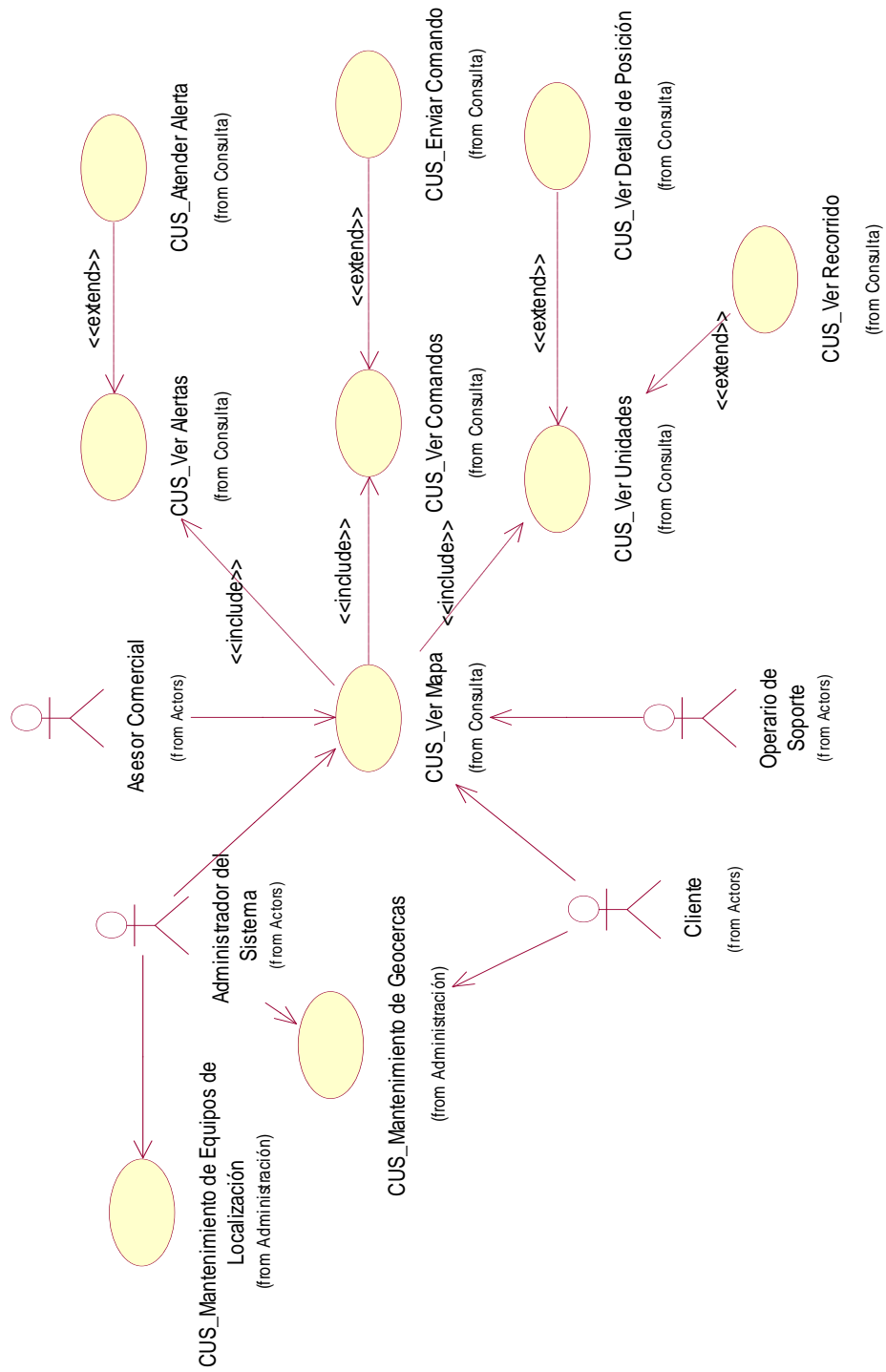


Figura 3.8 Arquitectura de Casos de Uso del Sistema

### 3.4 Descripción de Casos de Uso del Sistema

#### 3.4.1 Descripción de Casos de Uso del Sistema – Paquete Consulta

##### 3.4.1.1 CUS - Ver Mapa

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte

**Descripción:** El presente caso de uso permite a los actores del sistema visualizar un mapa de Google Maps y obtener toda la información relacionada a sus unidades vehiculares, como su última posición, el recorrido, alertas, y comandos enviados.

**Descripción de la secuencia principal – Cargar Mapa:** El Caso de Uso se inicia cuando el actor del sistema selecciona la opción “ver mapa” en el menú de opciones y el sistema presenta un mapa web junto con un panel de control y una barra de herramientas para buscar unidades y encontrar puntos de interés. El panel de control le permite, listar las unidades junto con su última transmisión, visualizar el recorrido de las unidades, visualizar las alertas y visualizar los últimos comandos enviados.

**Post-Condición:** El mapa web es cargado, junto con el panel y la barra de herramientas, las unidades son visualizadas en el mapa, y las unidades que están más próximas se agrupan por proximidad y nivel de zoom. El mapa podrá maximizarse ocupando toda la ventana del navegador web.

**Excepción:** En caso de que el actor del sistema no esté autenticado, el sistema deberá redirigirlo a la pantalla de inicio del sistema.

**Requerimientos no Funcionales:** El mapa web deberá funcionar con el navegador Mozilla Firefox versión 4.0.

##### 3.4.1.2 CUS - Ver Unidades

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte

**Descripción:** El presente caso de uso tiene como función listar las unidades en el panel de control del mapa web y actualizar la posición de las unidades en el mapa.

**Descripción de la secuencia principal:** El usuario del sistema selecciona la opción refrescar mapa y el sistema obtiene las últimas posiciones de las unidades asociadas a la cuenta cliente, el sistema muestra las unidades vehiculares en el panel y actualiza sus posiciones en el mapa web.

**Post-Condición:** Las posiciones de las unidades vehiculares son actualizadas, junto con la información en el panel, cada unidad en el mapa y en panel deberán mostrar opciones para ver el detalle de la trama, visualizar el recorrido de la unidad, enviar un comando o visualizar sus geocercas asociadas. Si el equipo de localización provee información del rumbo, la unidad deberá mostrar el rumbo como un marcador del mapa.

**Excepción:** En caso de que se pierda la conexión a internet, se deberá presentar un mensaje de no conexión.

**Requerimientos no Funcionales:** La visualización de unidades en el mapa no deberá exceder de los 4 segundos.

### 3.4.1.3 CUS - Ver Recorrido

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte

**Descripción:** El presente caso de uso tiene como función listar en el panel y mostrar en el mapa las últimas 30 transmisiones o tramas que generó el equipo de localización.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en un marcador que representa a una unidad vehicular, selecciona la opción ruta, el sistema presenta las últimas 30 posiciones en el mapa en formato de poli líneas y lista las transmisiones en el mapa.

**Post-Condición:** El recorrido de la unidad con sus últimas 30 posiciones es mostrado en el mapa y listados en el panel. Si el usuario del sistema desea puede visualizar el recorrido de otra unidad de manera simultánea.

**Excepción:** En caso de que la PC del usuario del sistema no tenga los recursos necesarios para mostrar los recorridos simultáneos el navegador web deberá preguntar al usuario si desea continuar con la funcionalidad.

**Requerimientos no Funcionales:** La presentación del recorrido no deberá exceder los 6 segundos.

### 3.4.1.4 CUS - Ver Alertas

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte

**Descripción:** El presente caso de uso tiene como función listar en el panel las alertas ordenadas por fecha ocurridas con las unidades vehiculares.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en la opción Ver Mapa del menú principal, el sistema muestra las alertas de sus unidades vehiculares ordenadas por fecha en el panel de la derecha.

**Post-Condición:** Las alertas de las unidades son mostradas en el panel de la izquierda, el usuario puede atender la alerta instanciando al caso de uso CUS\_Ver\_Detalle\_Posición.

**Excepción:** No presenta excepciones.

**Requerimientos no Funcionales:** La presentación de las alertas no deberá exceder los 4 segundos.

### 3.4.1.5 CUS - Ver Comandos

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte

**Descripción:** El presente caso de uso tiene como función listar en el panel los comandos enviados al equipo de localización y describir el estado del envío.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en la opción Ver Mapa del menú principal, el sistema muestra los comandos enviados a los equipos de localización de sus unidades ordenadas por fecha, e indicando el estado de cada envío.

**Post-Condición:** Los comandos enviados son mostrados en el panel de la izquierda, el usuario puede ver un pequeño resumen haciendo clic en un comando enviado.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** La presentación de los comandos no deberá exceder los 4 segundos.

### 3.4.1.6 CUS – Enviar Comando

**Actores:** Cliente, Administrador del Sistema, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función enviar un comando al equipo de localización.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en la ventana de información de una unidad, selecciona la opción “Comando”, el sistema le muestra una ventana de advertencia indicando que los comandos a enviar pueden modificar el comportamiento del equipo de localización, el usuario deberá aceptar el mensaje de advertencia, el sistema le muestra una lista de comandos a seleccionar, el usuario selecciona el comando y hace clic en la opción siguiente, el sistema le muestra el argumento que compone el comando, el usuario llena los argumentos y selecciona la opción enviar, el sistema le exige confirmar el envío del comando, el usuario confirma el envío y el comando es enviado al equipo, el sistema lista el comando enviado en el panel de comandos indicando si el comando llegó correctamente al equipo o aún no ha llegado al equipo .

**Post-Condición:** Los comandos enviados son mostrados en el panel de la izquierda, el usuario puede ver un pequeño resumen haciendo clic en un comando enviado.

**Excepción:** El sistema deberá enviar los comandos a través de un servicio proxy, en caso de que el servicio no se encuentre activo, el sistema deberá mostrar un mensaje indicando la ausencia del servicio de envío de mensajes.

**Requerimiento no Funcional:** El envío del comando no deberá exceder los 6 segundos.

## Prototipo del Caso de Uso del Sistema

### Enviar Comando (Paso 2)

UNIDAD: 1

Equipo: SKYPATROL

Servicio: CELULAR

COMANDO CONF. TIEMPO TRANSMISION  
ENVIAR UN REQUERIMIENTO DE POSICION ACTUAL

ARGUMENTOS:

HORA  EJEMPLO: 1 HORAS

MINUTOS  EJEMPLO: 30 MINUTOS

SEGUNDOS  EJEMPLO: 30 SEGUNDOS

[Cancelar](#)

Figura 3.15 Formulario de envío de comando

### 3.4.1.7 CUS – Ver Detalle de la Posición

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función ver todo el detalle de la posición de la unidad vehicular, ya sea como detalle de su recorrido o en la última trama emitida.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en la ventana de información de una unidad, selecciona la opción “Detalle”, el sistema le muestra una ventana donde muestra un mapa donde se indica la posición exacta de la unidad, y un grupo de pestañas que muestra el detalle de la trama, información de contacto, información del equipo y detalle de la alerta en caso de haberla, el usuario puede visualizar la información navegando por las pestañas.

**Post-Condición:** El usuario puede cerrar la ventana de información haciendo clic en cualquier parte del navegador (excepto la ventana) o haciendo clic en el ícono de cerrar de la ventana de información.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** La visualización de la ventana de información no deberá exceder los 6 segundos.

## Prototipo del Caso de Uso del Sistema



Figura 3.16 Ventana de información del Detalle de la Posición

### 3.4.1.8 CUS – Atender Alerta

**Actores:** Cliente, Administrador del Sistema, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función atender una alerta visualizada en la lista de alertas del panel de control del mapa.

**Descripción de la secuencia principal:** El usuario del sistema habiendo hecho clic en la lista de alertas del mapa, hace clic en una alerta, el sistema le muestra una ventana de información donde le da la opción de visualizar el “Detalle” de la posición (Ver punto 3.4.1.7), el sistema en la pestaña de alerta de la venta de información muestra un formulario con el campo de “Descripción de la Atención”, el usuario escribe la atención de la alerta y selecciona la opción “Atender Alerta”, el sistema registra la atención junto con la descripción indicada, así como al usuario que hizo la atención, junto con la fecha y hora.

**Post-Condición:** Una vez que la alerta es atendida, solo se podrá visualizar la descripción de la alerta.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El registro de la atención de la alerta no deberá exceder los 6 segundos.

## Prototipo del Caso de Uso del Sistema



Figura 3.17 Atención de la Alerta



### **3.4.1.9 CUS – Buscar Punto de Interés**

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función centrar el mapa web en función a un punto de interés.

**Descripción de la secuencia principal:** El usuario del sistema escribe el nombre de un punto de interés en la caja de texto de puntos de interés, el sistema muestra una lista de puntos de interés cuyos nombres coinciden con el texto ingresado, el usuario selecciona un registro de la lista y el sistema centra el mapa en función a la latitud y longitud del punto de interés seleccionado.

**Post-Condición:** Cuando el punto de interés es seleccionado, el centro del mapa es actualizado.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** Centrar el mapa deberá tomar un segundo.

### **3.4.1.10 CUS – Buscar Unidad**

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función ubicar una unidad vehicular y centrar el mapa web en función a las coordenadas de la unidad.

**Descripción de la secuencia principal:** El usuario del sistema escribe la placa de la unidad en la caja de texto de unidad, el sistema muestra una lista de unidades cuyas placas coincidan con el texto introducido, el usuario del sistema selecciona un registro de la lista presentada, el sistema centra el mapa web en función de la latitud y longitud de la unidad vehicular seleccionada presentándola en el mapa web.

**Post-Condición:** Cuando la unidad vehicular es seleccionada, el centro del mapa web es actualizado mostrando la unidad vehicular en el mapa.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El centrar el mapa en función a la unidad vehicular deberá ser inmediato.

### 3.4.1.11 CUS – Ver Puntos de Interés

**Actores:** Cliente, Administrador del Sistema, Asesor Comercial, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función visualizar los puntos de interés en el mapa web.

**Descripción de la secuencia principal:** El usuario del sistema selecciona la opción Puntos de Interés en el Mapa web, el sistema le muestra una ventana donde puede seleccionar los tipos de puntos de interés registrados a nombre de su cuenta cliente, el usuario selecciona los tipos de puntos de interés y selecciona la opción “Ver en Mapa”, el sistema muestra los puntos de interés asociados a los tipos de puntos de interés seleccionados, estos puntos de interés mostrados están conformados por un icono característico al tipo de punto de interés, el nombre, la descripción, página web, y teléfonos de contacto.

**Post-Condición:** Los puntos de interés permanecerán en el mapa web hasta que el usuario no refresque el mapa web.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El mostrar los puntos de interés en el mapa web no deberá de exceder los 6 segundos.

#### Prototipo del Caso de Uso del Sistema

Mostrar en el Mapa:

<input type="checkbox"/> AEROPUERTOS	<input type="checkbox"/> ALOJAMIENTO	<input type="checkbox"/> ALQUILER DE VEHICULOS	<input type="checkbox"/> BANCOS Y CAJEROS
<input type="checkbox"/> BARES Y CLUB	<input type="checkbox"/> BOMBEROS	<input type="checkbox"/> CEMENTERIOS	<input type="checkbox"/> CENTROS COMERCIALES
<input type="checkbox"/> COMIDA RAPIDA	<input type="checkbox"/> COMISARIAS	<input type="checkbox"/> ESCUELAS	<input type="checkbox"/> FARMACIAS
<input type="checkbox"/> FERRETERIAS	<input type="checkbox"/> GASOLINERIAS	<input type="checkbox"/> HOSPITALES	<input type="checkbox"/> IGLESIAS
<input type="checkbox"/> PEAJES	<input type="checkbox"/> POLICE	<input type="checkbox"/> RESTAURANTES	<input type="checkbox"/> TALLERES DE MECANICA
<input type="checkbox"/> TELEFONOS	<input type="checkbox"/> TIENDAS DE ALIMENTOS		

Ver en Mapa

Figura 3.20 Selección de Tipos de Puntos de Interés

### 3.4.1.12 CUS – Seleccionar Cliente en el Mapa

**Actores:** Administrador del Sistema, Operario de Soporte.

**Descripción:** El presente caso de uso tiene como función cambiar rápidamente de cuenta cliente en el mapa web visualizando sus unidades, sin necesidad de salir del sistema e ingresar con otro nombre de usuario asociado a esa cuenta cliente.

**Descripción de la secuencia principal:** El usuario del sistema en la caja de texto de la cuenta cliente, escribe el nombre de la cuenta cliente cuyas unidades vehiculares desea monitorear, el sistema le indica una lista de cuentas clientes cuyos nombres coinciden con el texto ingresado, el usuario selecciona un registro de la lista, el sistema elimina todos los objetos del mapa, del panel y actualiza la información con las unidades vehiculares de la cuenta cliente seleccionada.

**Post-Condición:** Todas las funcionalidades relacionadas deberán estar asociadas a la nueva cuenta cliente seleccionada.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El seleccionar una nueva cuenta cliente no deberá tomar más de 6 segundos.

### **3.4.2 Descripción de Casos de Uso del Sistema – Paquete Configuración**

#### **3.4.2.1 CUS – Mantenimiento de Modelos de Equipos**

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar, eliminar y listar un modelo de equipo.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Modelos” del menú de Configuración del sistema, el sistema muestra una lista paginada de 10 registros por página donde lista los modelos de los equipos, junto con las opciones de registrar, ver, eliminar, actualizar y comandos.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el nombre, la marca, el modelo, el servicio, la tecnología y una matriz donde podrá asociar los sensores registrados en el sistema a los puertos de entrada del modelo del equipo, el usuario ingresa la información solicitada y selecciona la opción registrar, el sistema le muestra un mensaje de éxito indicando que el modelo del equipo fue registrado satisfactoriamente.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema le muestra el detalle del modelo del equipo junto con la opción de eliminar, el usuario confirma eliminar el modelo del equipo, el sistema verifica que el modelo del equipo no tenga registros asociados y el usuario es dirigido a la lista de modelos indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un modelo con registros asociados.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Nombre:  \*

Marca:  \*

Modelo:

Servicio:

Tecnología:

#### Información de Sensores

Código	Nombre del Sensor
IO_01	<input type="text" value="..Sin Configurar.."/>
IO_02	<input type="text" value="..Sin Configurar.."/>
IO_03	<input type="text" value="..Sin Configurar.."/>
IO_04	<input type="text" value="..Sin Configurar.."/>
IO_05	<input type="text" value="..Sin Configurar.."/>
IO_06	<input type="text" value="..Sin Configurar.."/>
IO_07	<input type="text" value="..Sin Configurar.."/>
IO_08	<input type="text" value="..Sin Configurar.."/>
IO_09	<input type="text" value="..Sin Configurar.."/>
IO_10	<input type="text" value="..Sin Configurar.."/>
IO_11	<input type="text" value="..Sin Configurar.."/>
IO_12	<input type="text" value="..Sin Configurar.."/>

Registrar

[Cancelar](#)

Figura 3.22 Prototipo de formulario de registro de Modelo de Equipo

### 3.4.2.2 CUS – Asociar Comando a Modelo de Equipo

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad asociar los comandos registrados en el sistema a un modelo de equipo.

**Descripción de la secuencia principal – Mostrar Comandos:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Modelos” del menú de Configuración del sistema, el sistema muestra una lista paginada de 10 registros por página donde lista los modelos de los equipos, junto con las opciones de registrar, ver, eliminar, actualizar y comandos, el usuario selecciona la opción “comandos”, el sistema le muestra la lista de comandos registrados, junto con un estado, si fue seleccionado o no.

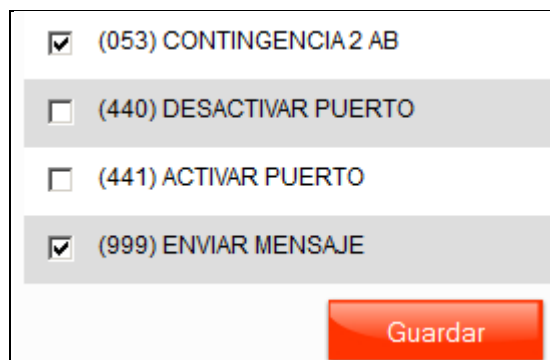
**Descripción de la secuencia secundaria – Asociar:** En la lista de comandos procedentes del flujo anterior, el usuario selecciona la opción de asignar comandos, el sistema muestra la misma lista junto con una opción de selección por cada comando, el usuario selecciona los comandos que desee que estén asignados al modelo del equipo y selecciona la opción guardar, el sistema asocia estos nuevos comandos al modelo del equipo y muestra un mensaje de éxito confirmando la operación.

**Post-Condición:** Es necesario asignar por lo menos un comando a un equipo, los comandos del modelo de equipo Genérico, serán heredados a los demás modelos de equipos

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 10 segundos.

#### Prototipo del Caso de Uso del Sistema



Prototipo de selección de comandos. Muestra una lista de comandos con casillas de selección y un botón 'Guardar'.

<input checked="" type="checkbox"/>	(053) CONTINGENCIA 2 AB
<input type="checkbox"/>	(440) DESACTIVAR PUERTO
<input type="checkbox"/>	(441) ACTIVAR PUERTO
<input checked="" type="checkbox"/>	(999) ENVIAR MENSAJE

Guardar

Figura 3.23 Selección de Comandos a fin de asociarlos a un Modelo de Equipo

### 3.4.2.3 CUS – Mantenimiento de Comandos

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar nuevos comandos al sistema que serán usados por los modelos de equipos.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Comandos” del menú de Configuración del sistema, el sistema muestra una lista paginada de 10 registros por página donde lista los comandos de los equipos, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el código, nombre y descripción del comando, junto con una matriz de 10 argumentos que requiere el nombre del argumento, la cantidad de caracteres e información de ayuda, el usuario selecciona la opción “Registrar”, el sistema registra la información del comando junto con sus argumentos y muestra un mensaje de éxito confirmando la operación .

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del comando junto con la opción de eliminar, el usuario confirma eliminar el modelo del equipo, el sistema verifica que el comando no tenga registros asociados y el usuario es dirigido a la lista de comandos indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un comando con registros asociados.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Código:  \*

Nombre:  \*

Descripción:

#### Información de Argumentos

	Nombre del Argumento	Caracteres	Información de Ayuda
Argumento 01	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 02	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 03	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 04	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 05	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 06	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 07	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 08	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 09	<input type="text"/>	<input type="text"/>	<input type="text"/>
Argumento 10	<input type="text"/>	<input type="text"/>	<input type="text"/>

Registrar

Cancelar

Figura 3.24 Formulario de registro de Comando junto con sus argumentos

#### 3.4.2.4 CUS – Mantenimiento de Eventos

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar eventos al sistema que serán usados para describir a más detalle las transmisiones de los equipos.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eventos” del menú de Configuración del sistema, el sistema muestra una lista paginada de 10 registros por página donde lista los eventos del sistema, junto con las opciones de registrar, ver, eliminar y actualizar.

Descripción de la secuencia secundaria – Registrar: El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el identificador del evento o ID, nombre, cliente que los solicitó, descripción del evento, así mismo el sistema muestra la opción para indicar si el evento es una alarma, proviene de un sensor o es un evento normal, si el evento debería considerarse como alerta y su prioridad. El usuario ingresa los datos indicados y selecciona la opción “Registrar”, el sistema

valida los datos ingresados y muestra un mensaje de éxito indicando la conformidad de la operación.

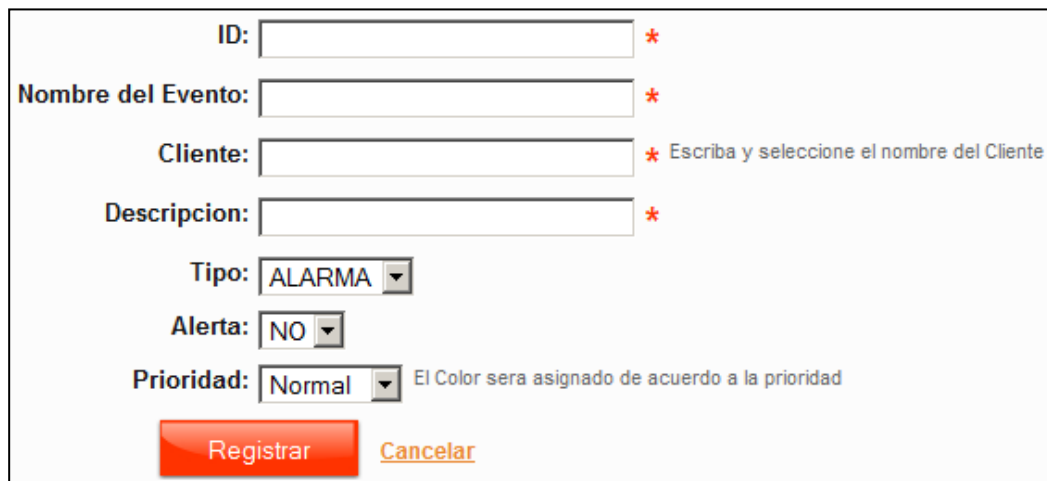
**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del evento junto con la opción de eliminar, el usuario confirma eliminar el evento del sistema, el sistema verifica que el evento no tenga registros asociados y el usuario es dirigido a la lista de eventos indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un evento con tramas asociadas.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema



El formulario de registro de Evento contiene los siguientes campos y controles:

- ID:** Campo de texto con un asterisco rojo (\*).
- Nombre del Evento:** Campo de texto con un asterisco rojo (\*).
- Cliente:** Campo de texto con un asterisco rojo (\*) y el texto "Escriba y seleccione el nombre del Cliente".
- Descripcion:** Campo de texto con un asterisco rojo (\*).
- Tipo:** Selector de lista desplegable con el valor "ALARMA".
- Alerta:** Selector de lista desplegable con el valor "NO".
- Prioridad:** Selector de lista desplegable con el valor "Normal" y el texto "El Color sera asignado de acuerdo a la prioridad".
- Botones:** "Registrar" (botón rojo) y "Cancelar" (enlace azul).

Figura 3.25 Formulario de registro de Evento

#### 3.4.2.5 CUS – Mantenimiento de Tipos de Puntos de Interés

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar tipos de puntos de interés al sistema que serán usados posteriormente en el mapa web para mostrar los puntos de interés.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Tipos POI” del menú de Configuración del sistema, el sistema muestra una lista paginada de 10 registros por página donde lista los Tipos de POI, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el nombre. El usuario ingresa el nombre y selecciona la opción “Registrar”, el sistema valida que el nombre no se



repita en el sistema y muestra un mensaje de éxito indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Actualizar Icono:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Ver” de la secuencia principal del caso de uso, el sistema muestra el nombre de Tipo de Punto de Interés junto con la opción de cargar un ícono, el usuario selecciona el ícono para el Tipo de Punto de Interés seleccionado y el sistema actualiza el ícono del Tipo de punto de interés.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del Tipo de Punto de Interés junto con la opción de eliminar, el usuario confirma eliminar el Tipo de Punto de Interés del sistema, el sistema verifica que el Tipo de Punto de Interés no tenga Puntos de Interés asociados y el usuario es dirigido a la lista de Tipo de Punto de Interés indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un Tipo de Punto de Interés con Puntos de Interés asociados.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema



Ingrese los datos del nuevo Tipo de POI

Nombre:  \*

Registrar Cancelar

Figura 3.26 Formulario de registro de Tipo de Punto de Interés

### 3.4.3 Descripción de Casos de Uso del Sistema – Paquete Administración

#### 3.4.3.1 CUS – Mantenimiento de Clientes

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar, actualizar y listar un clientes del sistema.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Clientes” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista los Clientes del sistema, que pueden ser persona jurídicas como naturales, así mismo el

sistema muestra una opción para ver solo Clientes que sean personas jurídicas, junto con las opciones de registrar, ver, eliminar, configuración y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el nombre de la cuenta cliente, el correo, la contraseña del usuario de la cuenta cliente, la confirmación de la contraseña, su estado con las opciones de activo e inactivo, los nombres y apellidos, el DNI, teléfono, celular, fax, página web y dirección de correspondencia. El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y presenta un mensaje en confirmando el éxito del procedimiento.

**Descripción de la secuencia secundaria – Configuración:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Configuración” de la secuencia principal del caso de uso, el sistema permite ingresar el máximo de Geocercas permitidas, el máximo de puntos de interés permitidos, así como la opción de seleccionar un nuevo usuario administrador del cliente. El usuario efectúa los cambios necesarios y selecciona la opción “Registrar”, el sistema valida los datos y presenta un mensaje de confirmación de la operación.

**Post-Condición:** No es posible eliminar un Cliente del sistema, solo desactivarlos.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

## Prototipo del Caso de Uso del Sistema

### Datos de la Empresa

RUC:  \*

Razon Social:  \*

Nombre Corto:  \*

Estado: Activo  \*

La Cuenta Expira el:  31  Agosto  2012  \* (Dia/Mes/Año)

Sector Económico: -- Seleccione --  \*

Representante Legal:  \*

Número de Empleados:

Vehículos interesados:

Perfil de la Empresa:

Teléfono:

Celular:

Fax:

Página Web:

Dirección Fiscal

País:  \*

Región:  \*

Ciudad:  \*

Distrito:  \*

Dirección:  \*

Nombres del Contacto:  \*

Apellidos del Contacto:  \*

Correo Correspondencia:  \*

Dirección de Facturación

País:

Región:

Ciudad:

Distrito:

Dirección:

Nombres del Contacto:

Apellidos del Contacto:

Correo Facturación:

Figura 3.27 Formulario de registro de Cliente

### 3.4.3.2 CUS – Mantenimiento de Equipos de Localización

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar los equipos de localización o comúnmente llamados Cards.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Cards” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista los Equipos, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el identificador del equipo (su ID), su número de serie, su clave de autenticación, el modelo del equipo, la fecha de expiración del equipo y si está activo o no, es decir si transmite. El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida que

número de serie junto con el identificador en el sistema y muestra un mensaje de éxito indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Subir por CSV:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Subir por CSV” de la secuencia principal del caso de uso, el sistema muestra un formulario donde permite cargar un documento de texto con el siguiente formato: ID, Número de Serie, Código de Autenticación, Fecha de Expiración y Estado de la Transmisión. El usuario selecciona el archivo que contiene los cards o equipos a cargar al sistema y selecciona la opción “Subir CSV”, el sistema verifica que los datos en el archivo ingresado tengan el formato correcto y verifica que no haya repetición de ID o Número de Serie. El sistema muestra un mensaje indicando el resultado de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del Equipo junto con la opción de eliminar, el usuario confirma eliminar el Equipo del sistema, el sistema verifica que el Equipo no esté asignado a ninguna unidad vehicular y el usuario es dirigido a la lista de Equipos indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un Equipo que posea una unidad asignada.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Formulario de registro de Equipo de Localización o Cards:

- ID:  \*
- Nro de Serie:  \*
- Clave de Autenticación:  \*
- Modelo:  ▼
- La Cuenta Expira el:  ▼  ▼  ▼ \* (Día/Mes/Año)
- Activo:  ▼
- Botones:

Figura 3.28 Formulario de registro de Equipo de Localización o Cards

#### 3.4.3.3 CUS – Mantenimiento de Unidades

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar las unidades vehiculares que serán presentadas en el mapa web.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Unidades” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista las unidades, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando la flota (o grupo de unidad a la que pertenecerá), la descripción de la unidad, el número de placa, el tipo de carga, la opción de seleccionar el chofer de la unidad, el Nickname de la unidad (que es el nombre de la unidad con la cual será identificada de manera más rápida), el teléfono de contacto, la marca, el modelo, el color, la eficiencia (los kilómetros por galón), y si la unidad esta activa o no. . El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y muestra un mensaje de éxito indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle de la unidad junto con la opción de eliminar, el usuario confirma eliminar la unidad del sistema, el sistema verifica que la unidad no tenga tramas emitidas, posteriormente el sistema muestra un mensaje indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar una unidad que haya emitido tramas.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Flota:

Nro de Serie:  \*

Clave de Autenticacion:  \*

Descripcion:

Nro de Placa:  \*

Tipo Carga:

Chofer:

Nickname:  \*

Telefono Contacto:  \*

Marca:  \*

Modelo:

Color:

Eficiencia:  Kilometros x Galón de combustible

Activo:

[Cancelar](#)

Figura 3.29 Formulario de registro de Unidades

#### 3.4.3.4 CUS – Mantenimiento de Flotas

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar las flotas o grupos de unidades vehiculares que serán presentadas en el mapa web.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Flotas” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista las flotas, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando el nombre de la flota y en caso de ser Administrador del Sistema, obliga a seleccionar a que cuenta cliente pertenece . El usuario ingresa los datos solicitados y selecciona la opción

“Registrar”, el sistema valida los datos ingresados y muestra un mensaje indicando la conformidad de la operación.

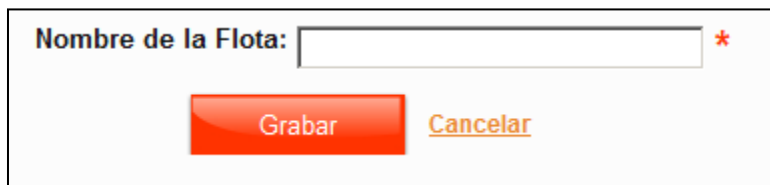
**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle de la flota junto con la opción de eliminar, el usuario confirma eliminar la flota del sistema, el sistema verifica que la flota no tenga unidades asociadas, posteriormente el sistema muestra un mensaje indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar una flota que posea unidades asociadas.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema



Nombre de la Flota:  \*

Figura 3.30 Formulario de registro de Flotas

#### 3.4.3.5 CUS – Mantenimiento de Conductores

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar conductores de unidades vehiculares o comúnmente llamados Choferes.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Conductores” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista los conductores, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando la cuenta cliente (en caso de que el usuario sea el súper administrador del sistema), el correo, nombres, apellidos, el tipo de documento, el estado (activo o inactivo), Llave Id, teléfono, celular, fax, pagina web, país, región, ciudad, distrito y dirección. El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y muestra un mensaje indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del conductor junto con la opción de eliminar, el usuario confirma eliminar al conductor del sistema, el sistema verifica que el



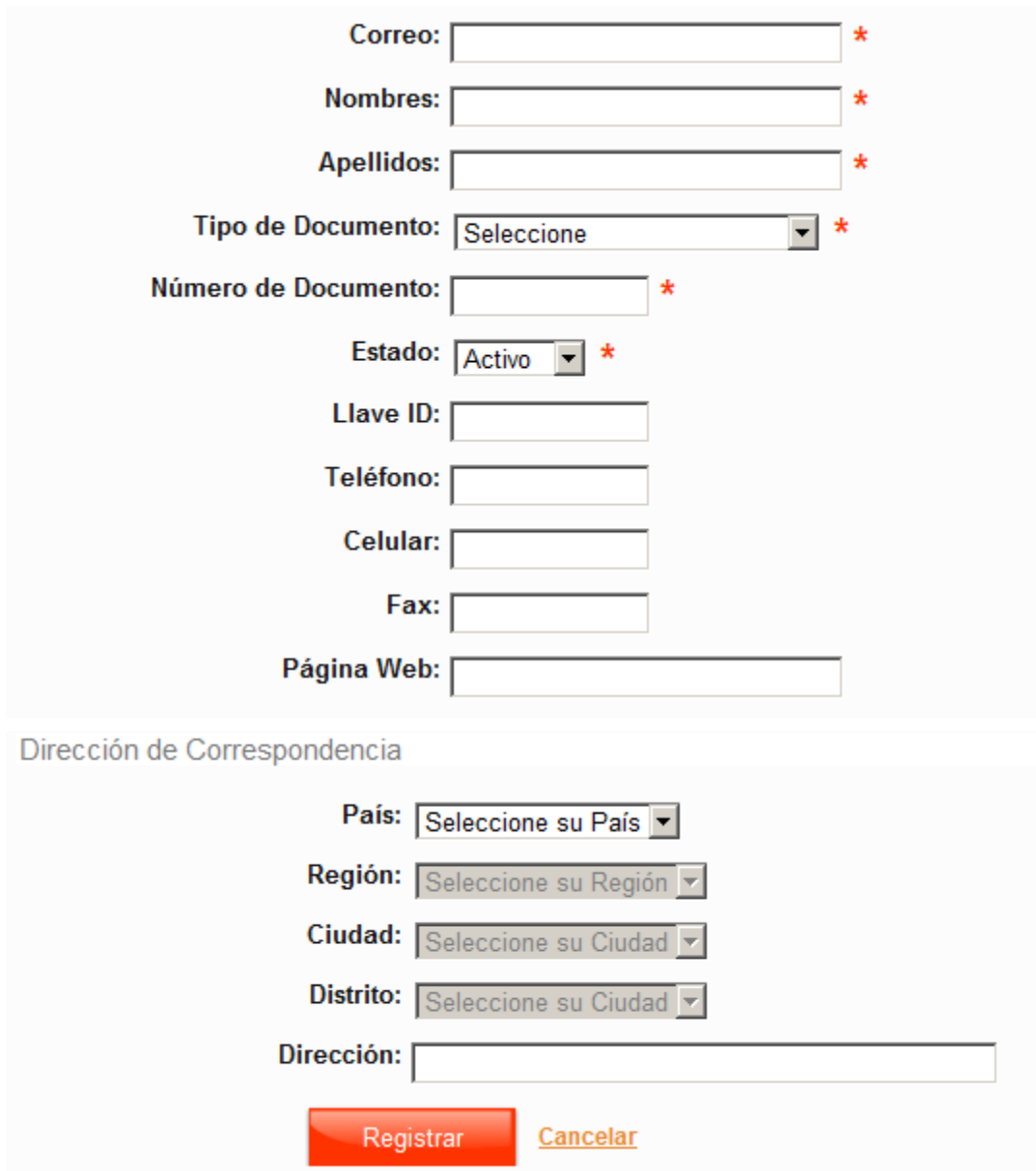
conductor no esté asociado a ninguna unidad vehicular, posteriormente el sistema muestra un mensaje indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar un conductor que este asignado a una unidad.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema



Formulario de registro de Conductores. El formulario está dividido en dos secciones: 'Datos Personales' y 'Dirección de Correspondencia'. La sección 'Datos Personales' incluye campos para Correo, Nombres, Apellidos, Tipo de Documento (lista desplegable), Número de Documento, Estado (lista desplegable), Llave ID, Teléfono, Celular, Fax y Página Web. La sección 'Dirección de Correspondencia' incluye campos para País, Región, Ciudad, Distrito (todos listas desplegables) y Dirección. Al final del formulario hay dos botones: 'Registrar' (rojo) y 'Cancelar' (naranja).

**Correo:**  \*

**Nombres:**  \*

**Apellidos:**  \*

**Tipo de Documento:** Seleccione  \*

**Número de Documento:**  \*

**Estado:** Activo  \*

**Llave ID:**

**Teléfono:**

**Celular:**

**Fax:**

**Página Web:**

**Dirección de Correspondencia**

**País:** Seleccione su País

**Región:** Seleccione su Región

**Ciudad:** Seleccione su Ciudad

**Distrito:** Seleccione su Ciudad

**Dirección:**

**Registrar** **Cancelar**

Figura 3.31 Formulario de registro de Conductores

### 3.4.3.6 CUS – Mantenimiento de Geocercas

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar geocercas que determinarán el comportamiento de las unidades vehiculares.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Geocercas” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista las geocercas, junto con las opciones de registrar, ver, eliminar, actualizar y flotas.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando la cuenta cliente (en caso de que el usuario sea el súper administrador del sistema), el nombre de la geocercas, el color de la geocerca, la velocidad máxima, la altitud, el tiempo de velocidad máxima 1 y el tiempo de velocidad máxima 2. El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y muestra un mensaje indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Actualizar Coordenadas:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Ver” de la secuencia principal del caso de uso, el sistema muestra el detalle de la geocerca junto con la opción de actualizar coordenadas, el usuario selecciona la opción de actualizar coordenadas, el sistema el muestra un mapa web donde podrá crear un polígono que represente la geocerca, así mismo el sistema muestra un formulario donde podrá indicar la exactitud a de cada punto del polígono y actualizarlo en el mapa. El usuario ingresa las coordenadas de la geocerca y el sistema valida que las coordenadas sean las correctas, posteriormente el sistema le muestra al usuario un mensaje indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle de la geocerca junto con la opción de eliminar, el usuario confirma eliminar a la geocerca del sistema, el sistema verifica que la geocerca no tenga tramas asociadas, el sistema le muestra un mensaje al usuario indicando el resultado de la operación.

**Post-Condición:** No es posible eliminar una geocerca con tramas asociadas.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Empresa:	<input type="text"/>	*
Nombre	<input type="text"/>	*
Color de la Geocerca:	<input type="text"/>	* <a href="#">Escojer Color</a>
Sombra de la Geocerca:	<input type="text"/>	* <a href="#">Escojer Sombra</a>
Velocidad Máxima (Km/h):	<input type="text"/>	Velocidad Máxima
Altitud:	<input type="text"/>	Altitud Máxima
TVM 1 (Segundos)	<input type="text"/>	Tiempo de Velocidad Máxima 1
TVM 2 (Segundos)	<input type="text"/>	Tiempo de Velocidad Máxima 2
<input type="button" value="Registrar"/>		<a href="#">Cancelar</a>

Figura 3.32 Formulario de registro de Geocerca



Latitud	Longitud
-11.99398761	-77.06492900
-11.99627543	-77.05027341
-11.99957069	-77.05289125
-11.99835334	-77.05937147
-12.00041024	-77.06480026

Figura 3.33 Formulario de actualización de coordenadas

### 3.4.3.7 CUS – Asociar Geocerca a Flotas

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad asociar geocercas a flotas a fin de hacer efectiva la influencia directamente con la unidad vehicular.

**Descripción de la secuencia principal – Ver Flotas:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Geocercas” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista las geocercas, junto con las opciones de registrar, ver, eliminar, actualizar y flotas. El sistema muestra la lista de flotas que pertenecen a la cuenta cliente, indicando si están asociadas o no a la geocerca seleccionada junto con la opción “Actualizar Flotas”.

**Descripción de la secuencia principal – Asociar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Actualizar Flotas” de la secuencia principal, el sistema presenta la lista de flotas asociadas a la cuenta cliente junto con la opción de asociar cada flota a la geocerca, así mismo por cada flota el sistema muestra opciones para indicar si la geocerca será de control, exclusiva, prohibida y el rango horario para aplicar dichas directivas a cada flota, el usuario selecciona las flotas que desea asociar a la geocerca y escoge la opción “Guardar”, el sistema valida los datos ingresado y presenta un mensaje donde indica el resultado de la operación.

**Post-Condición:** Una vez que las geocercas son asociadas a las flotas, las directivas seleccionadas afectaran a todas las unidades comprendidas dentro de las flotas o grupos de unidades.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

#### Prototipo del Caso de Uso del Sistema

<input type="checkbox"/>	Exclusivo	TEST	Hora Inicio: 00:00	Hora Fin: 00:00
<input type="checkbox"/>	Prohibido	DEMO	Hora Inicio: 00:00	Hora Fin: 00:00
<input type="button" value="Guardar"/> <input type="button" value="Limpiar"/> <input type="button" value="Cancelar"/>				

Figura 3.34 Formulario de asociación de geocerca a flotas

### 3.4.3.8 CUS – Mantenimiento de Puntos de Interés

**Actores:** Administrador del Sistema, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar puntos de interés que serán de referencia en el mapa web, ya sea para ubicar lugares importantes o para centrar el mapa en dichos puntos de interés.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “POI” del menú de Administración del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista los Puntos de Interés, junto con las opciones de registrar, ver, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario indicando la Empresa (En caso de que el usuario sea el súper administrador del sistema), el nombre del Punto de Interés, la dirección, teléfono, página web, correo, descripción y el tipo de punto de interés. El usuario ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y muestra un mensaje de éxito indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Actualizar posición:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Ver” de la secuencia principal del caso de uso, el sistema muestra la información relacionada al Punto de Interés junto con un mapa web indicando la opción “Actualizar posición”, el usuario selecciona la opción, el sistema muestra un mapa web junto con la opción de ubicar una posición con un marcador de mapa y un cuadro de coordenadas, el usuario indica la posición con el marcador y selecciona la opción “Actualizar”, el sistema valida la información ingresada y muestra un mensaje indicando el resultado de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del Punto de Interés junto con la opción de eliminar, el usuario confirma eliminar al Punto de Interés del sistema, el sistema le muestra un mensaje al usuario indicando el resultado de la operación.

**Post-Condición:** No presenta post-condiciones.

Excepción: No presenta excepciones.

Rendimiento: El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Nombre:  \*

Dirección:

Teléfono:

Página Web:  Ejemplo: <http://www.pagina.com>

Correo:

Descripción:

Tipo POI:  \*

Figura 3.35 Formulario de registro de punto de interés

## 3.4.4 Descripción de Casos de Uso del Sistema – Reportes

### 3.4.4.1 CUS – Generar Reportes de Recorrido

**Actores:** Operario de Soporte, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad generar el reporte de recorrido de una unidad vehicular y exportar el resultado del reporte en diferentes formatos.

**Descripción de la secuencia principal:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Recorrido” del menú de Reportes, el sistema muestra un formulario con los siguientes campos: flota, placa del vehículo, fecha de inicio, fecha de fin, hora de inicio y hora de fin, el usuario ingresa los datos solicitados y selecciona la opción “Siguiente”, el sistema muestra un reporte indicando los siguientes campos: Placa, evento, velocidad, geocerca, Latitud, longitud, altitud, rumbo, fecha y hora, distancia entre puntos, así mismo el sistema muestra la opción de exportar el reporte de recorrido en los formatos XML, XLS, CSV, PDF, KML.

**Post-Condición:** Una vez generado el reporte, los formatos de exportación deberán ser descargables.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

Flota : Seleccione la Flota \*

Placa : \* Escriba y seleccione la Placa

Fecha Inicio: 25/08/2011 [seleccione](#)

Fecha Fin: 25/08/2011 [seleccione](#)

Hora Inicio: 00 : 00

Hora Fin: 23 : 59

[Siguiete](#) [Cancelar](#)

Figura 3.36 Formulario de Generación de Reporte de Recorrido

#### 3.4.4.2 CUS – Generar Reporte de Alertas

**Actores:** Operario de Soporte, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad generar el reporte de alertas de una unidad vehicular y exportar el resultado del reporte en diferentes formatos.

**Descripción de la secuencia principal:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Alertas” del menú de Reportes, el sistema muestra un formulario con los siguientes campos: flota, placa del vehículo, fecha de inicio, fecha de fin, hora de inicio, hora de fin, rango de velocidades, geocercas, selección de eventos de tipo alerta, el usuario ingresa los datos solicitados y selecciona la opción “Siguiete”, el sistema muestra un reporte indicando los siguientes campos: alerta (nombre del evento), velocidad, geocerca, latitud, longitud, altitud, rumbo y fecha de activación, así mismo el sistema muestra la opción de exportar el reporte de alertas en los formatos XML,XLS, CSV, PDF.

**Post-Condición:** Una vez generado el reporte, los formatos de exportación deberán ser descargables.

**Excepción:** No presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema

The form contains the following fields and controls:

- Flota: Seleccione la Flota \* (dropdown menu)
- Placa: - Seleccione - (dropdown menu)
- Estado: Pendiente (dropdown menu)
- Fecha Inicio: 25/08/2011 (text input) with a "seleccione" link
- Fecha Fin: 25/08/2011 (text input) with a "seleccione" link
- Hora Inicio: 00 : 00 (time selection)
- Hora Fin: 23 : 59 (time selection)
- Velocidad: Seleccione (dropdown menu) and a text input field with "km/h" label
- Velocidad: Seleccione (dropdown menu) and a text input field with "km/h" label
- Geocerca: Seleccione la Geocerca (dropdown menu)
- Section: "Selecion de Alertas :"
- List of alerts: ALARMA CONDUCCION SIN DESCANSO, ALARMA - LOG LLENO, Apertura de Boveda, Apertura de Puerta, Apertura de Ventana, Bateria Baja de Gabinete, BOTON DE PANICO, Corte de Energia Principal, ENCENDIDO DEL EQUIPO - SKYPATROL, EVENTO DE CONMUTACION A BATERIA INTERNA
- Navigation buttons: >>, Todo >>, <<, Todo <<
- Buttons: "Siguiete" (red) and "Cancelar" (orange)

Figura 3.37 Formulario de Generación de Reporte

#### 3.4.4.3 CUS – Generar Reporte de Comandos

**Actores:** Operario de Soporte, Cliente

**Descripción:** El presente caso de uso del sistema tiene por finalidad generar el reporte envío de comandos a una unidad vehicular y exportar el resultado del reporte en diferentes formatos.

**Descripción de la secuencia principal:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Comandos” del menú de Reportes, el sistema muestra un formulario con los siguientes campos: flota, placa del vehículo, el tipo de comando, fecha de fin, hora de inicio, hora de fin y el estado del envío, el usuario ingresa los datos solicitados y selecciona la opción “Siguiete”, el sistema muestra un reporte indicando los siguientes campos: placa del vehículo, número de serie del equipo, autor del envío, argumentos, fecha de envío, fecha de llegada, nombre del comando, y el estado del envío, así mismo el sistema muestra la opción de exportar el reporte de comandos en los formatos XML,XLS, CSV, PDF.



**Post-Condición:** Una vez generado el reporte, los formatos de exportación deberán ser descargables.

**Excepción:** No presenta excepciones.

**Rendimiento:** El formulario no deberá tardar en presentarse más de 6 segundos.

### Prototipo del Caso de Uso del Sistema



Figura 3.38 Formulario de Generación de Reporte

## 3.4.5 Descripción de Casos de Uso del Sistema – Seguridad

### 3.4.5.1 CUS – Actualizar Cuenta

**Actores:** Usuario Autenticado

**Descripción:** El presente caso de uso del sistema tiene por finalidad actualizar los datos relacionados a la cuenta del usuario del sistema.

**Descripción de la secuencia principal – Detalle de la Cuenta de Usuario:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Cuenta” del menú de Seguridad, el sistema muestra la siguiente información:

Información de la cuenta: El nombre de la cuenta cliente, el número de geocercas permitidas, el número de puntos de interés permitidos.

Información de acceso: Correo, rol, estado (activo o inactivo)

Datos personales: Nombres, apellidos, DNI, teléfono, celular, fax, página web, región, ciudad, distrito y dirección.

Configuración del Mapa: Vista del mapa con la posición inicial de búsqueda, activar la alerta de sonido.

El sistema presenta toda esta información, junto con las opciones de: Ver información completa de la cuenta cliente, actualizar datos de ingreso al sistema, actualizar datos personales, actualizar la configuración del mapa.

**Descripción de la secuencia secundaria – Actualizar datos de ingreso al sistema:** La secuencia secundaria se da inicio cuando el usuario dentro de la secuencia principal, selecciona la opción “Actualizar ingreso al Sistema”, el sistema

muestra un formulario con los siguientes campos: contraseña, confirmación de contraseña, correo electrónico y rol de usuario, si el usuario deja la contraseña en blanco y selecciona la opción “Actualizar”, el sistema no actualizará la contraseña, si el usuario introduce otro correo electrónico o escoge otro rol de usuario y selecciona la opción “Actualizar” el sistema obligará al usuario a iniciar la sesión nuevamente dentro del sistema.

**Descripción de la secuencia secundaria – Actualizar datos personales:** La secuencia secundaria se da inicio cuando el usuario dentro de la secuencia principal, selecciona la opción “Actualizar datos personales”, el sistema muestra un formulario con los siguientes campos: Apellidos, nombres, tipo de documento, número de documento, teléfono, celular, fax, página web, país región ciudad, distrito, dirección, el usuario actualiza la información solicitada y selecciona la opción “Actualizar”, el sistema valida la información ingresa y presentará un mensaje confirmando la operación.

**Descripción de la secuencia secundaria – Actualizar configuración del mapa:** La secuencia secundaria se da inicio cuando el usuario dentro de la secuencia principal, selecciona la opción “Actualizar configuración del mapa”, el sistema muestra un formulario indicando la opción de seleccionar si se desea activar el sonido de la alerta, junto con un mapa donde el usuario puede indicar arrastrando un marcador la posición inicial donde el mapa se centrará cuando el usuario inicie el caso de uso “Ver Mapa”, el usuario selecciona si desea activar el sonido de la alerta, indica con el marcado el centro del mapa y selecciona la opción “Actualizar”, el sistema valida los datos ingresados y presenta un mensaje de confirmación indicando el resultado de la operación.

**Post-Condición:** Los datos de la cuenta del usuario han sido actualizados.

**Excepción:** En caso de que el usuario cambie su correo electrónico o rol, deberá de iniciar sesión nuevamente.

**Requerimiento no Funcional:** Cada formulario de los flujos secundarios no deberán tardar en cargarse más de 5 segundos.

### Prototipo del Caso de Uso del Sistema

Actualiza tu Contraseña

Contraseña:  \* Longitud mínima 4 caracteres

Repite tu Contraseña:  \*

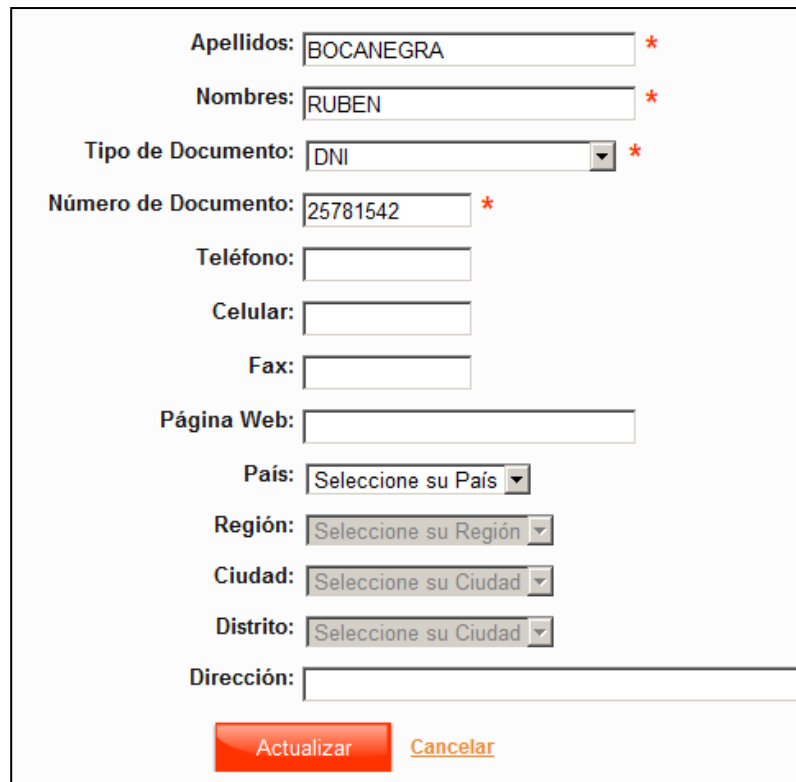
Actualiza tu Correo y Tipo de Cuenta

Correo Electrónico:  \*

Mi tipo de cuenta:  \*

Figura 3.39 Formulario de Actualización de Datos de Acceso

## Prototipo del Caso de Uso del Sistema



Apellidos: BOCANEGRA \*

Nombres: RUBEN \*

Tipo de Documento: DNI \*

Número de Documento: 25781542 \*

Teléfono:

Celular:

Fax:

Página Web:

País: Seleccione su País ▼

Región: Seleccione su Región ▼

Ciudad: Seleccione su Ciudad ▼

Distrito: Seleccione su Ciudad ▼

Dirección:

Figura 3.40 Formulario de Actualización de Datos de Personales

### 3.4.5.2 CUS – Ingresar al Sistema

**Actores:** Usuario no Autenticado

**Descripción:** El presente caso de uso del sistema tiene por finalidad iniciar sesión en el sistema a través de un formulario de identificación.

**Descripción de la secuencia principal – Ingresar al Sistema:** El Caso de Uso se da inicio cuando el usuario selecciona la opción “Acceder” en la pantalla de inicio del sistema, el sistema presenta un formulario solicitando el correo electrónico y la contraseña del usuario, el usuario ingresa los datos solicitados y selecciona la opción acceder, el sistema valida la correspondencia entre el correo y la contraseña, de ser positiva la validación se creará una sesión en el sistema, en caso contrario el sistema mostrará un mensaje indicando que el correo y/o la contraseña son incorrectas.

**Post-Condición:** El usuario habrá iniciado sesión en el sistema.

**Excepción:** Este caso de uso no presenta excepciones.

**Requerimiento no Funcional:** El formulario no deberá tardar en cargarse más de 4 segundos.

## Prototipo del Caso de Uso del Sistema

Correo Electrónico:  \*

Contraseña:  \* [Olvidaste tu Contraseña, ¡Haz click aqui!](#)

[Cancelar](#)

Figura 3.41 Formulario de Ingresar al Sistema

### 3.4.5.3 CUS – Recuperar Contraseña

**Actores:** Usuario no Autenticado

**Descripción:** El presente caso de uso del sistema tiene por finalidad reemplazar la contraseña existente por una nueva, a través de un procedimiento de identificación.

**Descripción de la secuencia principal – Enviar Instrucciones:** El Caso de Uso se da inicio cuando el usuario selecciona la opción “Recordar Contraseña” del flujo principal del caso de uso “Ingresar al Sistema”. El sistema muestra un formulario solicitando el correo electrónico, el usuario ingresa un correo electrónico válido y selecciona la opción “Enviar”, el sistema verifica que el correo electrónico corresponda a un usuario del sistema, en caso de no corresponder mostrará un mensaje indicando el error, en caso de que la correspondencia sea positiva, el sistema enviará un mensaje a la cuenta de correo electrónico ingresada indicando el procedimiento de recuperación de contraseña.

**Descripción de la secuencia secundaria – Restaurar Contraseña:** La secuencia secundaria se da inicio cuando el usuario no autenticado abre el correo electrónico enviado por el sistema y hace clic en enlace de recuperación de contraseña, el sistema valida en código de autenticación del enlace y presenta el formulario de restauración solicitando una nueva contraseña y la confirmación de la nueva contraseña, el usuario ingresa los datos solicitados y selecciona la opción “Actualizar Contraseña”, el sistema valida que los datos ingresados sean correctos, en caso no ser así el sistema mostrará un mensaje de error, en caso de que la validación sea correcta, el sistema dirigirá al usuario al flujo principal del caso de uso “Ingresar al Sistema”.

**Post-Condición:** El usuario deberá poder recuperar su contraseña.

**Excepción:** En caso de que la antigüedad del enlace supere las 24 horas, se declarará como inválido.

**Rendimiento:** El formulario no deberá tardar en cargarse más de 5 segundos.

## Prototipo del Caso de Uso del Sistema



El formulario muestra un campo de entrada de texto con el label "Correo Electrónico:" y un asterisco rojo a la derecha. Debajo del campo hay un botón rojo con el texto "Enviar" y un enlace "Cancelar" a la derecha.

Figura 3.42 Formulario de Envío de Instrucciones

### 3.4.5.4 CUS – Salir del Sistema

**Actores:** Usuario Autenticado

**Descripción:** El presente caso de uso del sistema tiene por finalidad eliminar la sesión del usuario y salir del sistema.

**Descripción de la secuencia principal – Salir del Sistema:** El Caso de Uso se da inicio cuando el usuario selecciona la opción “Salir del Sistema” que estará siempre presente mientras el usuario tenga una sesión válida en el sistema, una vez que la opción “Salir del Sistema” hay sido seleccionado por el usuario, el sistema destruirá la sesión del usuario y dirigirá al usuario al flujo principal del caso de uso “Ingresar al Sistema”

**Post-Condición:** El usuario podrá iniciar sesión nuevamente.

**Excepción:** No presenta ninguna excepción.

**Requerimiento no Funcional:** El formulario no deberá tardar en cargarse más de 5 segundos.

### 3.4.5.5 CUS – Mantenimiento de Usuarios

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar, actualizar y eliminar usuarios registrados en el sistema.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Usuarios” del menú de Seguridad del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista a los usuarios, junto con las opciones de registrar, ver, eliminar, actualizar y asociar flotas.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario solicitando la cuenta cliente (En caso de que el usuario sea administrador del sistema), el correo electrónico, la contraseña, la confirmación de la contraseña, el rol, el estado (Activo o Inactivo), nombres, apellidos, tipo de documento, número de documento, teléfono, celular, fax, página web, país, región, ciudad, distrito y dirección, el usuario ingresa los datos

solicitados y selecciona la opción “Registrar”, el sistema valida los datos ingresados y muestra un mensaje indicando la conformidad de la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el detalle del usuario junto con la opción de eliminar, el usuario confirma eliminar al usuario del sistema, el sistema verifica que el usuario no posea registros asociados, posteriormente el sistema muestra un mensaje indicando el resultado de la operación.

**Descripción de la secuencia secundaria – Asociar Flotas:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Flotas” de la secuencia principal del caso de uso, el sistema muestra la lista de flotas de la cuenta cliente a la que el usuario pertenece, el usuario selecciona la opción “Actualizar Flotas”, el sistema muestra la lista de flotas junto con la opción de seleccionar cada una de las flotas, el usuario selecciona la flotas y selecciona la opción “Guardar”, el sistema valida la información ingresada y presenta un mensaje confirmando la operación.

**Post-Condición:** El usuario registrado podrá iniciar sesión en el sistema y verá las unidades de las flotas a las que fue asociado.

**Excepción:** No presenta ninguna excepción.

**Requerimiento no Funcional:** El formulario no deberá tardar en cargarse más de 5 segundos.

## Prototipo del Caso de Uso del Sistema

Correo Electrónico:  \*

Contraseña:  \* Longitud mínima 4 caracteres

Repite la Contraseña:  \*

Rol: SuperAdministrador ▼ \*

Estado del Usuario: Activo ▼ \*

Datos de la Persona

Nombres:  \*

Apellidos:  \*

Tipo de Documento: Seleccione ▼ \*

Número de Documento:  \*

Ingrese el teléfono:

Ingrese el celular:

Fax:

Página Web:

Dirección de Correspondencia

País: Seleccione su País ▼

Región: Seleccione su Región ▼

Ciudad: Seleccione su Ciudad ▼

Distrito: Seleccione su Ciudad ▼

Dirección:

Figura 3.43 Formulario de Registro de Usuario

### 3.4.5.6 CUS – Mantenimiento de Roles

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad registrar, actualizar y eliminar roles en el sistema.

**Descripción de la secuencia principal – Listar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Roles” del menú de Seguridad del sistema, el sistema muestra una lista paginada de 20 registros por página donde lista a los roles, junto con las opciones de registrar, ver permisos, eliminar y actualizar.

**Descripción de la secuencia secundaria – Registrar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Registrar” de la secuencia principal del caso de uso, el sistema muestra un formulario solicitando el nombre del rol, el usuario ingresa el nombre del rol y selecciona la opción “Registrar”, el sistema valida que el nombre del rol sea único y muestra un mensaje confirmando la operación.

**Descripción de la secuencia secundaria – Eliminar:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Eliminar” de la secuencia principal del caso de uso, el sistema muestra el nombre del rol junto con la opción de eliminar, el usuario confirma eliminar el rol del sistema, el sistema verifica que el rol no posea registros asociados, posteriormente el sistema muestra un mensaje indicando el resultado de la operación.

**Post-Condición:** El Rol registrado podrá ser asociado posteriormente a un usuario del sistema.

**Excepción:** No presenta ninguna excepción.

**Requerimiento no Funcional:** El formulario no deberá tardar en cargarse más de 5 segundos.

#### Prototipo del Caso de Uso del Sistema

Nombre:  \* Máximo 20 caracteres

[Cancelar](#)

Figura 3.44 Formulario de Registro de Rol

### 3.4.5.7 CUS – Asociar Permisos

**Actores:** Administrador del Sistema

**Descripción:** El presente caso de uso del sistema tiene por finalidad asociar permisos a funcionalidades del sistema a roles de usuario.

**Descripción de la secuencia principal – Cambiar Permisos:** El Caso de Uso se inicia cuando el usuario selecciona la opción “Ver Permisos” del flujo principal del caso de uso “CUS\_Mantenimiento de Roles”, el sistema muestra la lista de funcionalidades del sistema, junto con las acciones Ver, Registrar, Editar, Eliminar, Activar y Publicar, el usuario selecciona la opción “Asociar Permisos”, el sistema le



presenta la misma lista con la opción de activar cada acción por funcionalidad, el usuario selecciona que actividades desea activar por funcionalidad y selecciona la opción “Establecer Permisos”, el sistema actualiza los permisos asociándolos al rol seleccionado y muestra un mensaje confirmando la operación.

**Post-Condición:** Los permisos serán asociados al rol seleccionado.

**Excepción:** No presenta ninguna excepción.

**Requerimiento no Funcional:** El formulario no deberá tardar en cargarse más de 5 segundos.

### Prototipo del Caso de Uso del Sistema

Código	Recurso	Ver	Reg	Edi	Eli	Act	Pub
Rol	SuperAdministrador						
A000	Administracion						
A001	Clientes	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A002	Flotas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A003	Cards	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A004	Unidades	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A005	Conductores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A006	Geocercas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
A007	POI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Establecer Permisos

Figura 3.45 Formulario de asociación de permisos

## 3.5 Benchmarking

### 3.5.1 Soluciones Encontradas

#### 3.5.1.1 Find me Spot

##### Descripción del servicio

“Find me Spot” [SPOT 11], es el servicio de rastreo de personas que brinda la compañía SPOT LLC, esta empresa tiene como principal producto al localizador GPS vía satélite personal SPOT, que es utilizado como instrumento de mensajería y reporte de emergencias con una amplia cobertura global basándose en tecnología satelital al 100%.



Imagen obtenida de findmespot.com

**Figura 3.46 Localizador GPS SPOT**

Este equipo es usado en su mayoría por personas que realizan actividades al aire libre en tierra, aire y mar, y generalmente están separados de cualquier cobertura celular que les facilite establecer comunicación con la civilización.

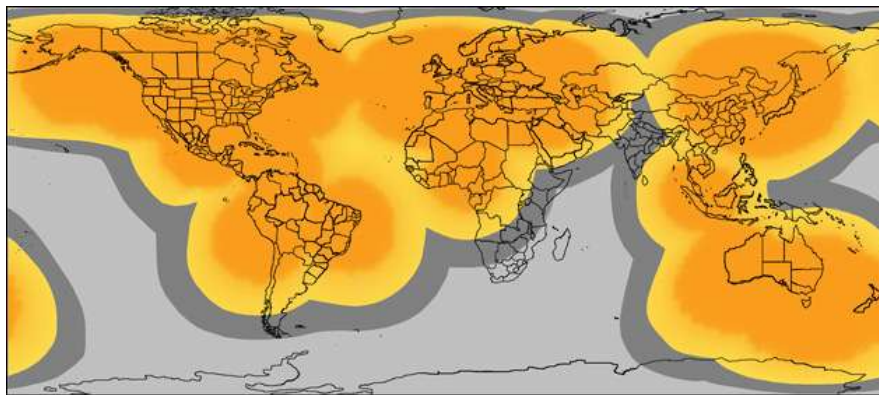


Imagen obtenida de findmespot.com

**Figura 3.47 Cobertura Satelital SPOT.**

La compañía SPOT LLC, comercializa el equipo mediante el pago anual de 150 dólares americanos, este monto incluye el servicio de “Seguimiento Compartido” y de “Alertas de Rescate”, el servicio de “Seguimiento Compartido” es presentado en una aplicación web que informa la posición actual de la persona que porta el equipo SPOT y su ruta hecha desde que empezó a transmitir el equipo, el acceso esta información puede ser visualizada por 10 contactos previamente registrados por el usuario del servicio, en caso de que el portador del equipo SPOT tenga alguna emergencia, los contactos se darán por enterados y podrán tomar acciones pertinentes, el servicio de “Alertas de Rescate” consta de un convenio que tiene SPOT LLC con los principales centros de rescate organizados, que efectuarán un rescate si el usuario del equipo SPOT envía un mensaje de alerta SOS, estos centros de rescate podrán ver la posición actual del usuario SPOT mediante el “Seguimiento Compartido” que brinda el servicio de “Find me Spot”.

## Descripción funcional de la aplicación

La aplicación web es utilizada para el seguimiento de la persona que utiliza el equipo SPOT, esta aplicación permite guardar la localización exacta del usuario y permite a sus contactos seguir su progreso prácticamente en tiempo real usando Google Maps. Da la posibilidad de configurar una página compartida de SPOT que le permite a cuantas personas desee compartir sus localizaciones a través de una URL en Google Maps.

Una vez activada, SPOT adquiere y envía sus coordenadas GPS a la cuenta SPOT del usuario automáticamente cada 10 minutos durante 24 horas o hasta que el usuario decida cancelar la transmisión. El servicio de seguimiento deberá ser restablecido para continuar.

La aplicación web permite visualizar la última posición y el recorrido del portador del equipo SPOT, esta información es presentada a través del Mapa de Google usando como framework el API de Google Maps.



Imagen obtenida de findmespot.com

Figura 3.48 Presentación del recorrido y la última posición del portador SPOT

Podemos apreciar como cada posición del recorrido esta numerada con la etiqueta "Track" junto con el número de posición "Track 2", "Track 3", etc. de esta forma se puede realizar un correcto seguimiento de la ruta que realiza el portador del equipo SPOT, así también podemos apreciar la etiqueta "Ok" que representa el estado del portador del equipo cuando llegue a algún punto de referencia, podemos rescatar 2 cosas importantes, el orden del seguimiento y la presentación del estado del portador del equipo.



Imagen obtenida de findmespot.com

**Figura 3.49 Detalle de cada posición**

La aplicación de seguimiento no sólo muestra el recorrido del portador del equipo, además muestra el detalle de cada posición, ya que al hacer clic en cada posición, una venta flotante es presentada detallando la posición en Latitud y Longitud.

Generalmente en un recorrido presenta decenas e inclusive cientos de posiciones, para lo cual presentarlos todos en una vista de mapa suele ser difícil de interpretar, para lo cual la aplicación de seguimiento utiliza un panel de seguimiento o “Tracking Panel”, donde el usuario puede visualizar las posiciones de manera ordenada.

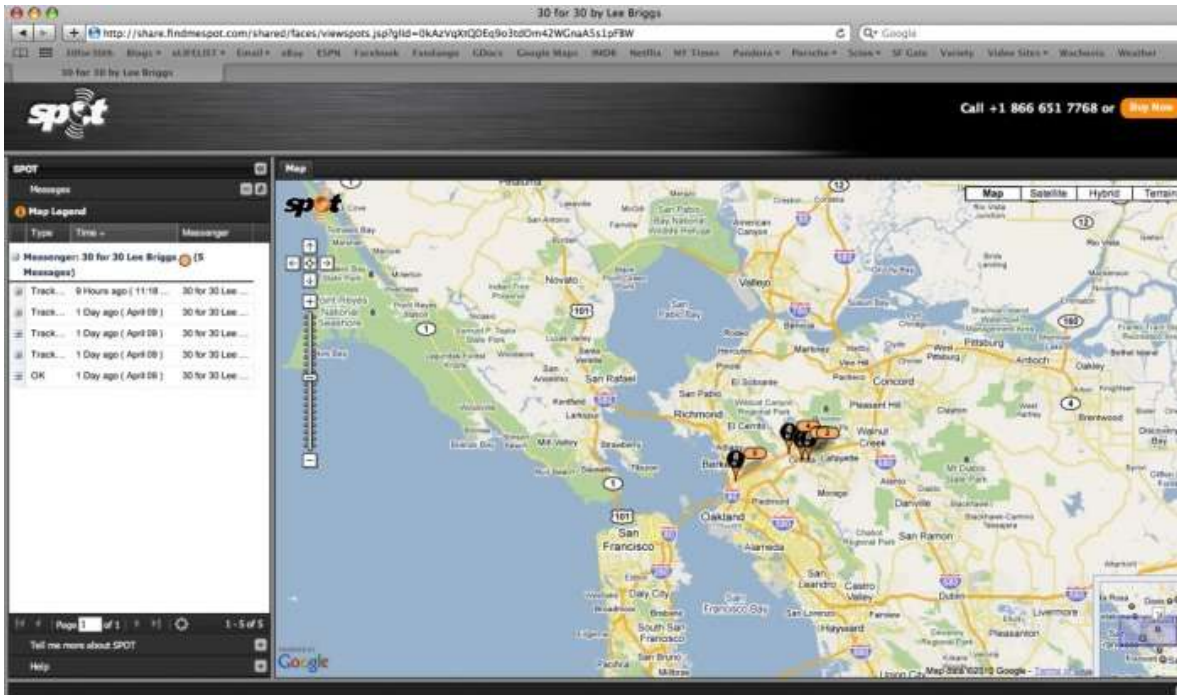


Imagen obtenida de findmespot.com

Figura 3.50 El panel de seguimiento está ubicado al lado izquierdo del gráfico.

### 3.5.1.2 Absolute Software

#### Descripción de la Compañía

“Absolute Software” [ABSOLUTE SOFTWARE 11] es una compañía que provee soluciones para monitorear la ubicación y el estado de los activos móviles tanto en tierra como en mar. La aplicación de monitoreo que presenta Absolute Software es llamada “The Fleet Information System” que es una completa solución web para el monitoreo de embarcaciones con soporte para la mayoría de navegadores web incluyendo Internet Explorer, Firefox, Chrome y Safari. Si bien esta aplicación tiene una versión para el monitoreo de vehículos terrestres, es reconocida por el uso de embarcaciones marítimas ya que la mayoría de sus funcionalidades están orientadas a vehículos que usan la vía marítima como medio de transporte. Absolute Software es usado para monitorear alrededor de 12,00 barcos alrededor de 90 países diariamente.

#### Descripción funcional de la aplicación

La aplicación que Absolute Software presenta es “The Fleet Information System”, esta es una aplicación que puede ser ejecutada en la mayoría de los navegadores ya que su plataforma de mapas está basada en Google Maps API, así mismo presenta la última posición y el rumbo de cada embarcación o unidad.

Esta aplicación permite la presentación de recorridos simultáneos, indicando el rumbo de la posición final, no obstante, no muestra el detalle de cada posición en la presentación del recorrido.

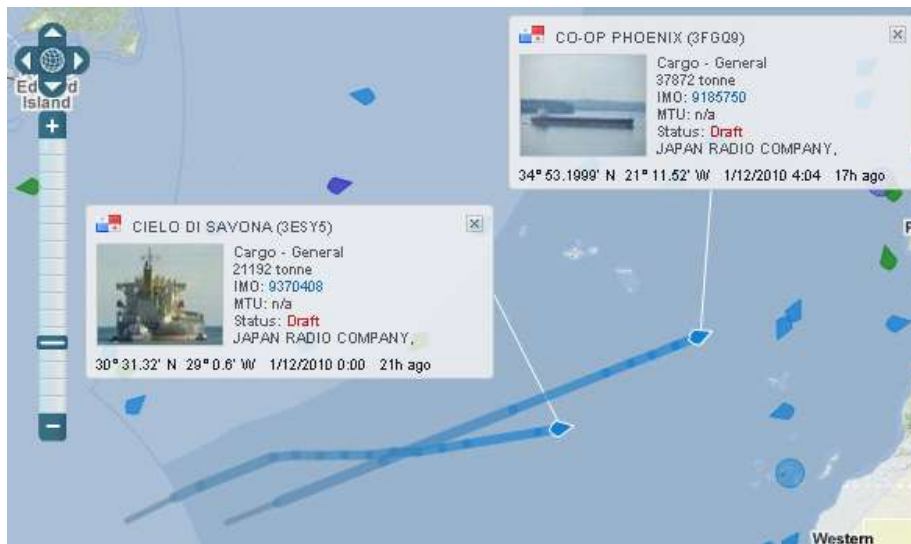


Imagen obtenida de [www.absolute.com](http://www.absolute.com)

**Figura 3.51 Recorridos simultáneos (Ver líneas azules)**

En la figura superior podemos apreciar que al hacer clic en la última posición de cada unidad, se muestra una ventana emergente mostrando el detalle de cada unidad, donde presenta el nombre de la unidad, el cliente, el estado, la latitud, longitud, la fecha, el tiempo transcurrido de la última transmisión, y la foto de cada unidad.



Imagen obtenida de [www.absolute.com](http://www.absolute.com)

**Figura 3.52 Agrupamiento de unidades en áreas circulares**

En la figura superior se muestran unidades en forma de rombo que indican el rumbo de cada unidad y otras en forma circular, se da el caso que en un área del mapa y en función del nivel de acercamiento más de una unidad se encuentra muy cerca de otra haciendo difícil la visualización de estas, por lo tanto, la aplicación agrupa estas unidades en áreas circulares que representan dos o más unidades.

Se dan ocasiones en las que son decenas o cientos de unidades que son visualizadas en el mapa, por tal motivo esta aplicación web, brinda otro medio de presentar la información, la cual es hecha a través de mosaicos, donde en cada mosaico muestra el detalle de cada unidad, de esa forma el responsable del monitoreo puede visualizar el estado de todas las unidades sin necesidad de usar el mapa web.

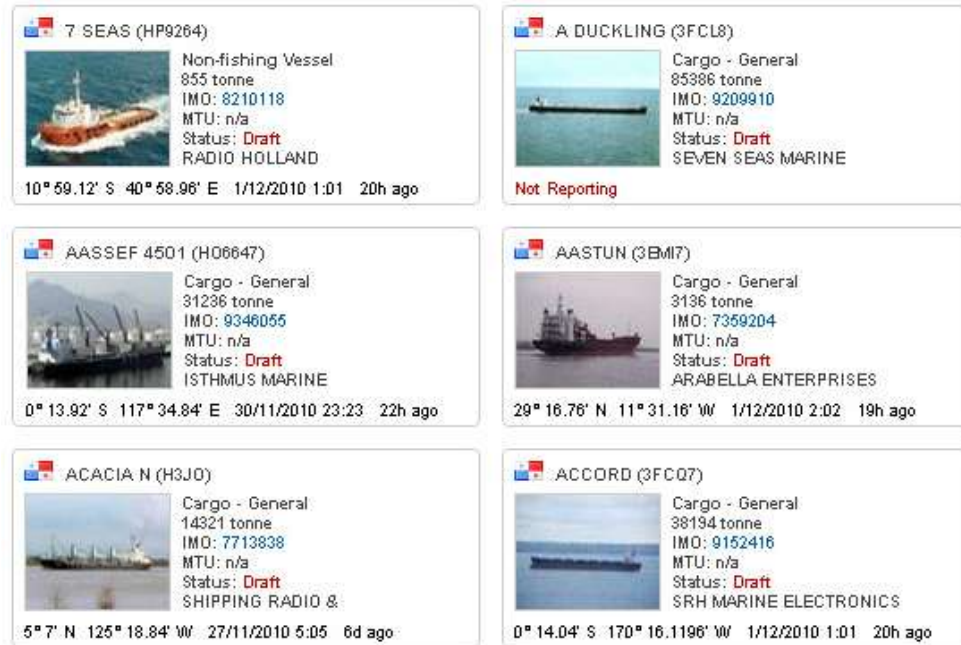


Imagen obtenida de [www.absolute.com](http://www.absolute.com)

**Figura 3.53 Presentación de unidades en mosaicos como alternativa al mapa web.**

Otras funcionalidades relacionadas son presentadas por Absolute Software como registro y rastreo de embarcaciones, manejo de polígonos o geocercas, generación de reportes y administración de permisos y licencias, todo a través de una aplicación web 2.0.

Los polígonos de pesca representan una zona geográfica donde las embarcaciones pueden navegar y efectuar sus operaciones, este tipo de polígonos son conocidos como geocercas exclusivas, o de control donde la unidad solo puede movilizarse.

### 3.5.1.3 Scope Technologies

#### Descripción de la Compañía

“Scope Technologies Holdings Limited” [SCOPE TECHNOLOGIES 11] es una compañía especializada en proveer servicios de administración telemática de flotas y fuerza de trabajo móvil. Fundada en 1997 provee servicios a 37 países con soluciones modulares, entre ellas, la aplicación de seguimiento vehicular Mzone 4.

Esta aplicación web permite el seguimiento de unidades vehiculares que poseen el dispositivo de rastreo GPS llamado MHub. Este equipo no solo envía la latitud y longitud de la unidad vehicular, además provee una serie de información a través de sensores, valiosa para la administración de la flota y el correcto uso de recursos, todo esto a través de la aplicación web Mzone 4.

#### Descripción funcional de la aplicación

Mzone 4 es una aplicación web que está orientada al manejo logístico de flotas de una compañía, posee una serie de funcionalidades administrativas, que permiten manejar la información de las actividades operacionales, administrar la utilización de recursos e informar periódicamente el comportamiento de las flotas a través de reportes programados que detallan información arrojada por el equipo Mhub. Así mismo permite el seguimiento de las unidades a través de un mapa web, teniendo como plataforma Microsoft Silverlight.

Esta aplicación web presenta mapas de diferentes proveedores de mapas como LeadDog, MapIT, MapQuest, entre otros, dando la facilidad de escoger al usuario el mapa que presente la información más relevante con respecto al entorno de la unidad vehicular.

Así mismo presenta el recorrido de cada unidad el cual lo denomina “Track History”, donde representa en una animación las locaciones por donde atravesó el vehículo, señalando de color rojo el recorrido de la unidad junto con su rumbo.

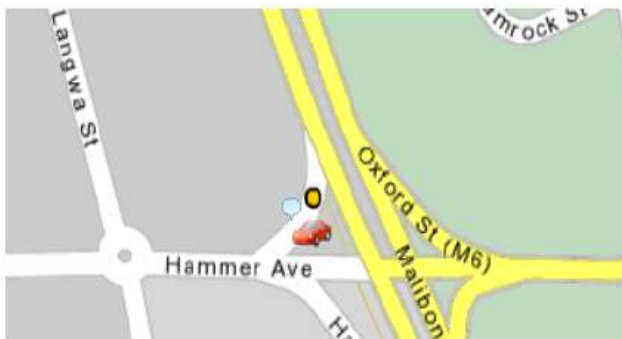


Imagen obtenida de scopetechnology.com

Figura 3.54 Posición actual de la unidad con su “Track History” inactivo.





Imagen obtenida de scopetechnology.com

Figura 3.55 Representación animada del recorrido del vehículo con su "Track History" activo.

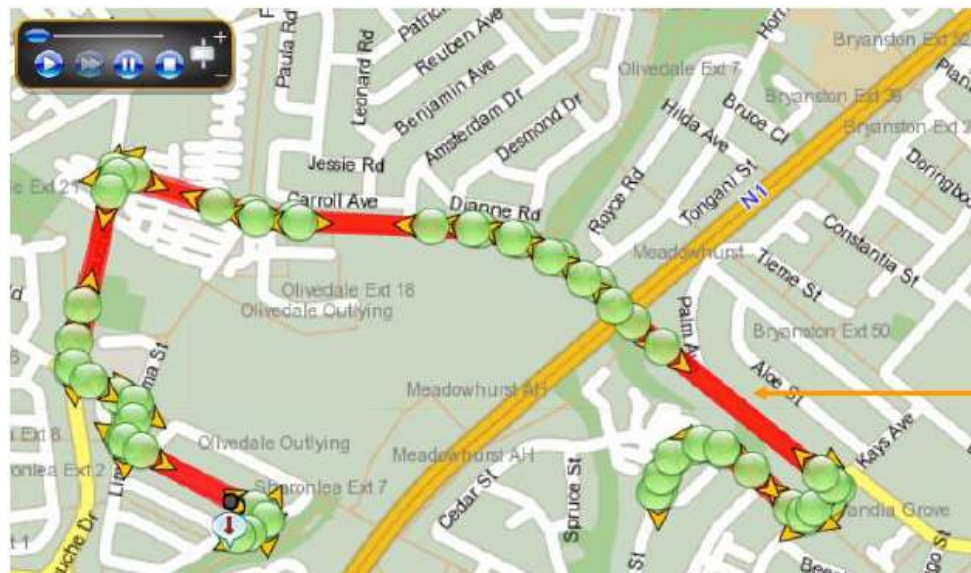


Imagen obtenida de scopetechnology.com

Figura 3.56 Controles de la animación del recorrido (Parte superior izquierda de la figura)

Otra característica muy útil que presenta esta aplicación, es la representación de áreas geográficas de interés, que pueden ser de referencia al usuario que utiliza la aplicación web para monitorear los vehículos, Mzone 4 permite agregar polígonos al mapa para señalar áreas de referencia a fin de establecer puntos de control o geocercas de exclusivas o prohibidas.



Imagen obtenida de scopetechnology.com

**Figura 3.57** El área de color azul es la zona geográfica de referencia

Así mismo posee funcionalidades para administrar los recursos de la flota, como por ejemplo el combustible, y permite la generación automática de reportes y el envío de estos en diferentes formatos a los interesados por correo electrónico.

### **3.5.1.4 GPS INSIGHT**

#### **Sobre el producto**

GPS INSIGHT [GPS INSIGHT 11] es definido por el proveedor como un producto de mantenimiento de flotas amigable para el usuario que tiene un paquete de reportes completo, mapas alertas y gráficos. Permite personalizar funcionalidades accesibles para requerimientos avanzados hechos por los usuarios incluyendo rutas, horarios, herramientas para medir la eficiencia de los conductores y mapeo de capas.

#### **Mapas en 2D y 3D**

GPS INSIGHT utiliza Google Maps como mapa web a fin de visualizar la posición y/o ruta de las unidades vehiculares, así mismo es posible visualizar la ruta del vehículo en 3D a través de Google Earth, de esta manera el usuario puede tener una visualización más detallada del entorno por donde transitó el vehículo.



Imagen obtenida de www.gpsinsight.com

Figura 3.58 Visualización de la posición del vehículo en 2D con Google Maps

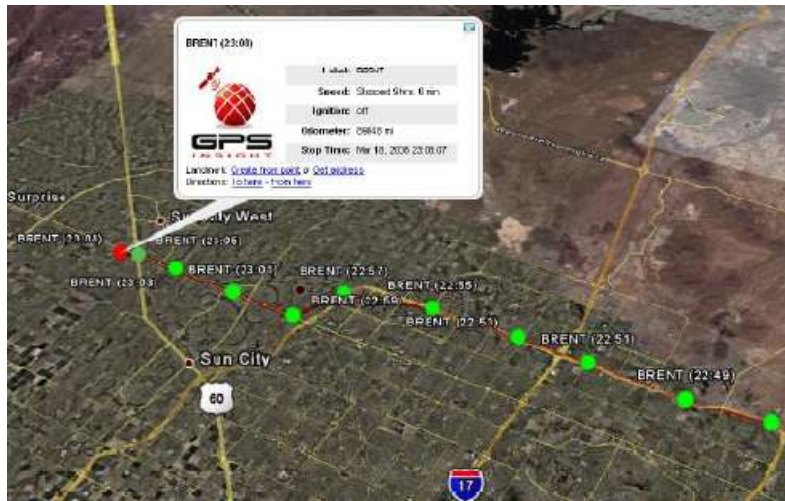


Imagen obtenida de [www.gpsinsight.com](http://www.gpsinsight.com)

Figura 3.59 Visualización del recorrido en 3D con Google Maps

### Presentación de geocercas

GPS INSIGHT permite visualizar geocercas poligonales que cubren un área geográfica y a su vez presentan las unidades que están comprendidas dentro de dicha área. Así mismo registra la entrada y la salida de las unidades de las geocercas, esta información es presentada a través de reportes tabulares.

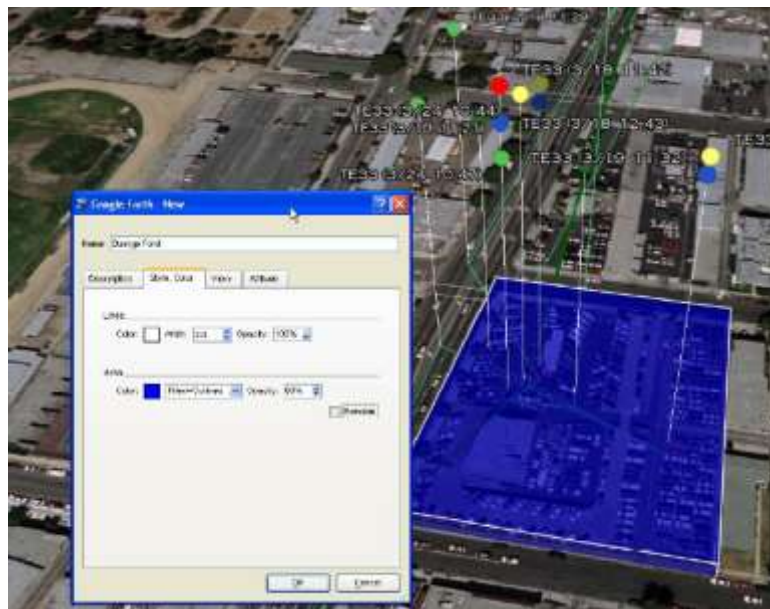


Imagen obtenida de [www.gpsinsight.com](http://www.gpsinsight.com)

Figura 3.60 Visualización de unidades en la geocerca

## El Panel de Control

El producto presenta un panel de control que permite organizar la visualización de las unidades en el mapa, este panel ubicado al lado derecho del mapa permite seleccionar que unidades visualizar e informa la última posición, la velocidad, el estado (En caso de ser alerta o no) y la fecha y hora de la última posición.



Imagen obtenida de [www.gpsinsight.com](http://www.gpsinsight.com)

Figura 3.61 Visualización del panel de control en el mapa

## 3.5.2 Evaluación de las soluciones encontradas

### 3.5.2.1 Evaluación - Find me Spot

#### Funcionalidades destacadas

Las funcionalidades más representativas de esta aplicación web son las siguientes:

1. Visualización de la última posición.
2. Visualización del recorrido del usuario.
3. Presentación del estado el usuario en cada posición.
4. Detalle de latitud y longitud de cada posición.
5. Numeración y etiquetado de cada posición en un recorrido.
6. Panel de seguimiento de posiciones.

#### Aporte de esta solución

El presente sistema aporta funcionalidades básicas que una aplicación web de monitoreo vehicular debe de considerar, como la visualización de la última posición, el recorrido, el estado de cada unidad, el detalle de la última posición, la numeración del recorrido y el panel de seguimiento de posiciones.

### **3.5.2.2 Evaluación – Absolute Software**

#### **Funcionalidades destacadas**

Las funcionalidades más representativas de esta aplicación web son las siguientes:

- 1 Visualización de la última posición junto con su recorrido.
- 2 Presentación del rumbo de la aplicación en forma de rombo.
- 3 Presentación de recorridos simultáneos.
- 4 Agrupación de unidades en áreas circulares.
- 5 Lista de unidades en mosaicos como presentación alternativa.
- 6 Geocercas o polígonos.

#### **Aporte de esta solución**

La aplicación que Absolute Software aporta funcionalidades importantes de desempeño de visualización en el mapa dando un uso más cómodo al usuario, así mismo aporta funcionalidades no vistas en otras soluciones como los recorridos simultáneos y la agrupación de las unidades en áreas circulares o agrupamiento de marcadores.

### **3.5.2.3 Evaluación – Scope Technologies**

#### **Funcionalidades destacadas**

Las funcionalidades más representativas de esta solución son las siguientes

- 1 Recorrido animado de la unidad vehicular.
- 2 Puntos de interés como geocercas.
- 3 Reportes automáticos en diferentes formatos.
- 4 Publicación de diferentes mapas a selección del usuario.

#### **Aporte de esta solución**

Mzone presenta una funcionalidad particular que es de mucha utilidad para sus usuarios, la cual es la presentación de la información en diferentes formatos, ya que permite enviar la información a través de medios electrónicos y poder ser reproducida en otros dispositivos GIS, como por ejemplo, es posible exportar un recorrido en KML desde el mapa web, este archivo puede ser abierto en otro computador por un GIS que reproduzca este formado como por ejemplo Google Earth.

### **3.5.2.4 Evaluación – GPS INSIGHT**

#### **Aporte de esta solución**

GPS INSIGHT es una aplicación web de monitoreo vehicular que sin duda da aportes funcionales muy importantes en el mercado de aplicación de tracking, por ejemplo la visualización de unidades vehiculares en un geocerca, así como la visualización del recorrido de la unidad en Google Earth, así mismo da un excelente aporte de cómo organizar la visualización de las unidades en el mapa a través del panel de control.

### **3.5.2.5 Benchmarking competitivo**

El objetivo del benchmarking competitivo en la presente investigación aplicada, es identificar las funcionalidades de mayor utilidad de las mejores aplicaciones web de monitoreo en el mercado mundial, y compararlos con el producto a desarrollar en la presente investigación aplicada.

#### **Identificación de funcionalidades críticas del producto a desarrollar**

1. Presentación del detalle de la posición de la Unidad.
2. Presentación del recorrido de la unidad.
3. Panel de seguimiento de posiciones.
4. Agrupación de unidades cercanas.
5. Gestión de geocercas.
6. Presentación de puntos de interés en el mapa.
7. Exportación del recorrido.

#### **Identificación de productos en el mercado**

A continuación se presentarán los productos en el mercado que presentan dichas funcionalidades:

**Presentación del detalle de la posición de la Unidad:** Los productos “Find me Spot”, “The Fleet Information System”, “GIS INSIGHT” y “MZone4”, presentan esta funcionalidad, cabe destacar que todos estos presentan un resumen del estado de la unidad cuando son presentados en el mapa, y presentan un detalle más profundo de la última transmisión de la unidad cuando se escoge ver el detalle de la transmisión.

Así mismo cabe resaltar que es importante conocer el rumbo de la unidad cuando esta está presente en el mapa, de esa manera se conocerá a que dirección se dirige, esta funcionalidad que es parte del detalle de la posición de la unidad está presente en los siguientes productos: “The Fleet Information System” y “Mzone 4”.

**Presentación del recorrido de la unidad:** Los productos “Find me Spot”, “The Fleet Information System”, “GIS INSIGHT” y “MZone4”, presentan el recorrido de la unidad junto con las coordenadas de cada posición, una característica importante de la presentación del recorrido de la unidad es el etiquetado del mismo, indicando la numeración del recorrido, “Find me Spot” es el único producto de los cuatro mencionados que presentan esta funcionalidad.

En ciertas ocasiones es de suma utilidad ver recorridos simultáneos de diferentes unidades vehiculares a fin de realizar ciertos análisis, por ejemplo la hora exacta en que dos unidades pasaron por el mismo lugar y en que direcciones se dirigían. Esta funcionalidad está presente solo en el producto “The Fleet Information System”.

**Panel de seguimiento de posiciones:** El panel de seguimiento de posiciones, es una lista situada en un extremo de la pantalla de monitoreo que enumera las unidades presentes en el mapa y su recorrido. Esta herramienta puede llegar a ser muy útil si se poseen muchas unidades y estas transmiten con regularidad, esta funcionalidad está presente en los siguientes productos: “Find me Spot”, “GIS INSIGHT” y “MZone4”.

**Agrupación de unidades cercanas:** En el caso de que existan tantas unidades presentes en una determinada área del mapa que dificulte la visibilidad de estas, es necesario agrupar dichas unidades en función a su cercanía y nivel de zoom. Esta funcionalidad lleva a ser muy útil si se monitorean cientos de unidades, esta funcionalidad está presente en el producto “The Fleet Information System”.

**Gestión de geocercas:** Las geocercas son polígonos en dos dimensiones que definen una zona en el mapa, con el uso de las geocercas es posible establecer parámetros de control para las unidades que estén dentro o fuera de sus geocercas asociadas. Los productos que gestionan geocercas son los siguientes: “The Fleet Information System”, “GIS INSIGHT” y “Mzone 4”.

**Presentación de puntos de interés en el mapa:** El punto de interés representa las coordenadas de un lugar que desea presentarse en el mapa para fines informativos, esto puede ser muy útil en el caso que se desee visualizar estaciones de gas y auxilio mecánico cerca de las unidades. Esta funcionalidad, está presente en el producto “Mzone 4”.

**Exportación del recorrido:** La exportación del recorrido consiste en presentar el recorrido visualizado en el mapa en un formato que puede ser compartido por otros usuarios o enviado por correo electrónico, el formato generalmente es KML (Keyhole Markup Language). Los productos que presentan esta funcionalidad son: “The Fleet Information System”, “GIS INSIGHT”, “Mzone 4”.

### **Tabla comparativa de funcionalidades**

A continuación se presenta las funcionalidades mencionadas anteriormente en una matriz donde es posible apreciar que productos presentan dichas funcionalidades.

<i>Funcionalidad/Producto</i>	<b>Find me Spot</b>	<b>The Fleet Information System</b>	<b>GPS INSIGHT</b>	<b>Mzone 4</b>	
<b>Presentación del detalle de la unidad</b>	SI	SI	SI	SI	<b>4</b>
<b>Presentación del recorrido de la unidad</b>	SI	SI	SI	SI	<b>4</b>
<b>Panel de seguimiento de posiciones</b>	SI	NO	SI	SI	<b>3</b>
<b>Agrupación de unidades cercanas</b>	NO	SI	NO	NO	<b>1</b>
<b>Gestión de geocercas</b>	NO	SI	SI	SI	<b>3</b>
<b>Presentación de puntos de interés en el mapa</b>	NO	NO	NO	SI	<b>1</b>
<b>Exportación del recorrido</b>	NO	SI	SI	SI	<b>3</b>
<b>Suma de funcionalidades</b>	<b>3</b>	<b>5</b>	<b>5</b>	<b>6</b>	

En función a la tabla comparativa podemos apreciar que 5 de 7 funcionalidades están presentes en casi todos los productos, dando a entender que son de suma importancia y por lo tanto el producto a desarrollar deberá presentarlos también en definitiva. Así mismo se puede apreciar que solo dos funcionalidades están presentes en un producto cada una respectivamente, estas son: La agrupación de unidades cercanas y la presentación de puntos de interés en el mapa. Si bien estas funcionalidades no son muy comunes en la lista de productos anteriormente mencionados, la dificultad de implementarlas son mínimas y aportarían de mucho valor al producto software a desarrollar, por lo tanto a fin de tener una solución completa, estas funcionalidades también son consideradas como requerimientos del sistema.

### **3.6 Conclusiones:**

Los requerimientos del proyecto es un documento que representa el compromiso del equipo de desarrollo en implementar todas las funcionalidades planteadas. A lo largo de cada una de sus secciones, se han presentado las diversas prestaciones que tendrá el producto software y se han analizado las principales funcionalidades de herramientas similares en el mercado mundial. Si bien el software a desarrollar es producto de una investigación aplicada, se han tomado en cuenta las mejores funcionalidades de las herramientas más usadas en el mercado, a fin de hacerlas trabajar juntas en un solo producto. Los requerimientos presentados en el presente documento abrazan fuertemente la visión del proyecto y perfilan el proyecto de desarrollo como suficiente para cumplir los objetivos planteados inicialmente.



## CAPÍTULO IV - ARQUITECTURA

### 4.1 Introducción:

En el presente capítulo presentaremos 2 disciplinas importantes del proceso de desarrollo de software, la primera es el análisis del sistema y posteriormente el diseño del sistema.

El Análisis del Sistema presenta una solución conceptual a los requisitos del sistema, esta solución conceptual tendrá como fundamento los casos de uso del sistema junto con sus especificaciones, producto de esto se definirán clases de tipo interfaz, controlador y entidad, los cuales describirán su interacción en diagramas de colaboración, teniendo como base los requisitos del sistema podremos identificar que clases serán las responsables de llevar a cabo diversas funcionalidades, por ejemplo: Análisis y presentación de geocercas, presentación de recorridos simultáneos, presentación de la última posición de la unidad vehicular, entre otros. Producto de los diagramas de colaboración podremos concebir posteriormente en el diseño, que clases intervendrán en los diagramas de secuencia y como se relacionarán, al final del análisis se presentará el diagrama de clases de tipo entidad, que será uno de los fundamentos para el diseño físico de la base de datos, ya que veremos una aproximación de las relaciones entre clases que representan contenedores de información.

El Diseño del Sistema presenta diversos artefactos que servirán de base a la implementación de la solución software, entre los artefactos a presentar son los siguientes:

Diseño de la Arquitectura, donde se presentará la base de datos, el lenguaje de programación, los sistemas operativos y diversos componentes que formarán la solución software.

La correspondencia entre clases de análisis y clases de diseño, donde identificaremos el origen de cada clase de diseño tomando como raíz las clases de análisis por cada caso de uso, Así mismo se presentarán las principales funciones que intervendrán para cumplir los requisitos del sistema.

El diagrama físico de la Base de Datos que tomará en cuenta artificios para optimizar recursos, consistencia y velocidad en función al gestor de base de datos seleccionado, si se toma en cuenta que una sola unidad vehicular puede transmitir cada 10, 5, 3, 1 minutos en función de cómo esté configurada y un cliente puede tener entre 1 a 100 unidades vehiculares, el Diagrama Físico de la Base de datos deberá tener en cuenta puntos clave como índices, redundancia de datos, registros temporales y relaciones óptimas a fin de tener los resultados esperados.

Diagrama de componentes, que ilustrará donde cada funcionalidad programada será ubicada físicamente en un directorio de archivos, en función a su extensión y al servicio que ofrezca a fin de organizar de manera adecuada los archivos compilados de la solución software.

## 4.2 Análisis del sistema

### 4.2.1 Realización de casos de uso de análisis

Se entiende por realización de Caso de Uso a la descripción de cómo el Caso de Uso se comportará de manera interna, la representación UML de una realización es la siguiente:



Figura 4.1 Realización de Modelos

A continuación se presenta la realización de Casos de Uso de Análisis agrupados por paquetes:

Paquete	Caso de Uso del Sistema	Caso de Uso de Análisis
Consulta	CUS_Ver Mapa	CUA_Ver Mapa
	CUS_Ver Unidades	CUA_Ver Unidades
	CUS_Seleccionar Cliente en Mapa	CUA_Seleccionar Cliente en Mapa
	CUS_Ver Detalle de Posición	CUA_Ver Detalle de Posición
	CUS_Ver Recorrido	CUA_Ver Recorrido
	CUS_Ver Alertas	CUA_Ver Alertas
	CUS_Atender Alertas	CUA_Atender Alertas
	CUS_Ver Comandos	CUA_Ver Comandos
	CUS_Enviar Comandos	CUA_Enviar Comandos
	CUS_Ver Punto de Interes	CUA_Ver Punto de Interes
	CUS_Buscar Unidad	CUA_Buscar Unidad
CUS_Buscar Punto de Interes	CUA_Buscar Punto de Interes	

Paquete	Caso de Uso del Sistema	Caso de Uso de Análisis
Configuración	CUS_Mantenimiento de Modelo de Equipos	CUA_Mantenimiento de Modelo de Equipos
	CUS_Mantenimiento de Eventos	CUA_Mantenimiento de Eventos
	CUS_Mantenimiento de Comandos	CUA_Mantenimiento de Comandos
	CUS_Asociar Comando a Modelo de Equipo	CUA_Asociar Comando a Modelo de Equipo
	CUS_Mantenimiento de Tipo de Punto de Interés	CUA_Mantenimiento de Tipo de Punto de Interés

Paquete	Caso de Uso del Sistema	Caso de Uso de Análisis
Administración	CUS_Asociar Geocerca a Flota	CUA_Asociar Geocerca a Flota
	CUS_Mantenimiento de Clientes	CUA_Mantenimiento de Clientes
	CUS_Mantenimiento de Conductores	CUA_Mantenimiento de Conductores
	CUS_Mantenimiento de Equipos de Localización	CUA_Mantenimiento de Equipos de Localización
	CUS_Mantenimiento de Flotas	CUA_Mantenimiento de Flotas
	CUS_Mantenimiento de Geocercas	CUA_Mantenimiento de Geocercas
	CUS_Mantenimiento de Puntos de Interés	CUA_Mantenimiento de Puntos de Interés

Paquete	Caso de Uso del Sistema	Caso de Uso de Análisis
Reportes	CUS_Generar Reportes de Alertas	CUA_Generar Reportes de Alertas
	CUS_Generar Reporte de Comandos	CUA_Generar Reporte de Comandos
	CUS_Generar Reporte de Recorrido	CUA_Generar Reporte de Recorrido

Paquete	Caso de Uso del Sistema	Caso de Uso de Análisis
Seguridad	CUS_Actualizar Cuenta	CUA_Actualizar Cuenta
	CUS_Ingresar al Sistema	CUA_Ingresar al Sistema
	CUS_Asociar Permisos	CUA_Asociar Permisos
	CUS_Mantenimiento de Roles	CUA_Mantenimiento de Roles
	CUS_Mantenimiento de Usuarios	CUA_Mantenimiento de Usuarios
	CUS_Recuperar Contraseña	CUA_Recuperar Contraseña
	CUS_Salir del Sistema	CUA_Salir del Sistema

## 4.2.2 Realización de casos de uso de análisis – Paquete Consulta

### 4.2.2.1 CUA - Ver Mapa

#### Diagrama de Clases

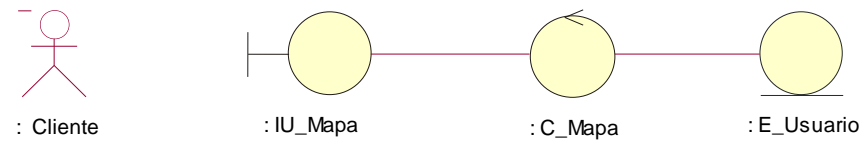


Figura 4.2 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Cargar Mapa

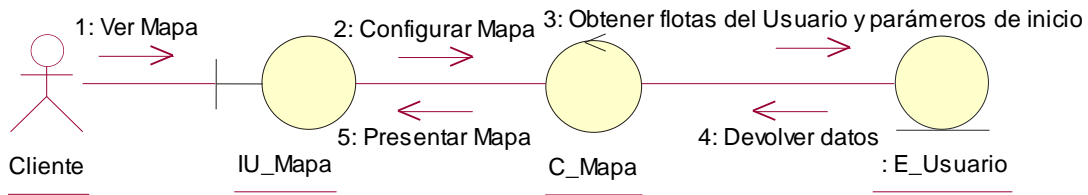


Figura 4.3 Diagrama de colaboración “Cargar Mapa”

**Descripción del diagrama de colaboración – Cargar Mapa:** El usuario selecciona la opción “Ver Mapa” de la interfaz IU\_Map, la interfaz solicita la carga de la configuración del mapa a la controladora C\_Map, la clase controladora obtiene los parámetros de personalización y las flotas asociadas a la clase entidad E\_Usuario, una vez obtenidas las flotas asociadas del usuario y los parámetros de configuración del mapa, la clase controladora C\_Map inicializa los parámetros de configuración a la clase interfaz IU\_Map y la clase IU\_Map presenta el mapa al usuario.

#### 4.2.2.2 CUA - Ver Unidades

##### Diagrama de Clases

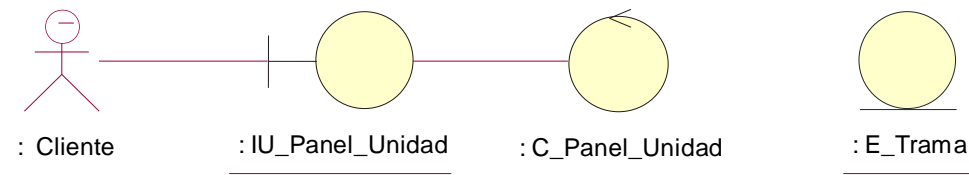


Figura 4.4 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Ver Unidades

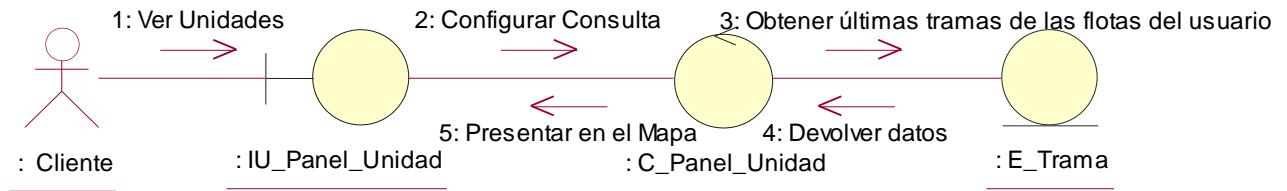


Figura 4.5 Diagrama de colaboración “Ver Unidades”

**Descripción del diagrama de colaboración – Ver Unidades:** El usuario ya sea al cargar el mapa o al refrescar el mapa hace la solicitud de ver las unidades a la interfaz de usuario IU\_Panel\_Unidad, esta le pide a la clase controladora C\_Panel\_Unidad que construya la consulta de las últimas tramas de las flotas

asociadas al usuario, la clase C\_Panel\_Unidad, le solicita a la clase E\_Trama que le entregue la última trama de cada unidad asociada a las flotas del usuario, esta le devuelve los registros a la clase controladora C\_Panel\_Unidad, la clase controladora entrega los registros a la interfaz de usuario IU\_Panel\_Unidad, y esta clase presenta las unidades en el mapa y en el panel de unidades.

### 4.2.2.3 CUA - Ver Recorrido

#### Diagrama de Clases

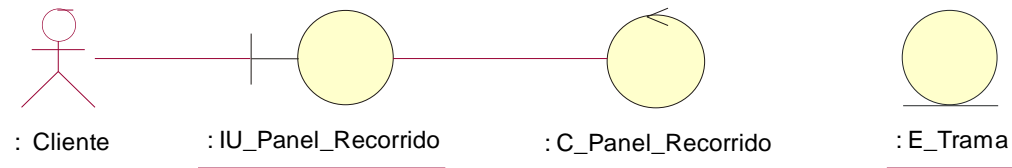


Figura 4.6 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Ver Recorrido

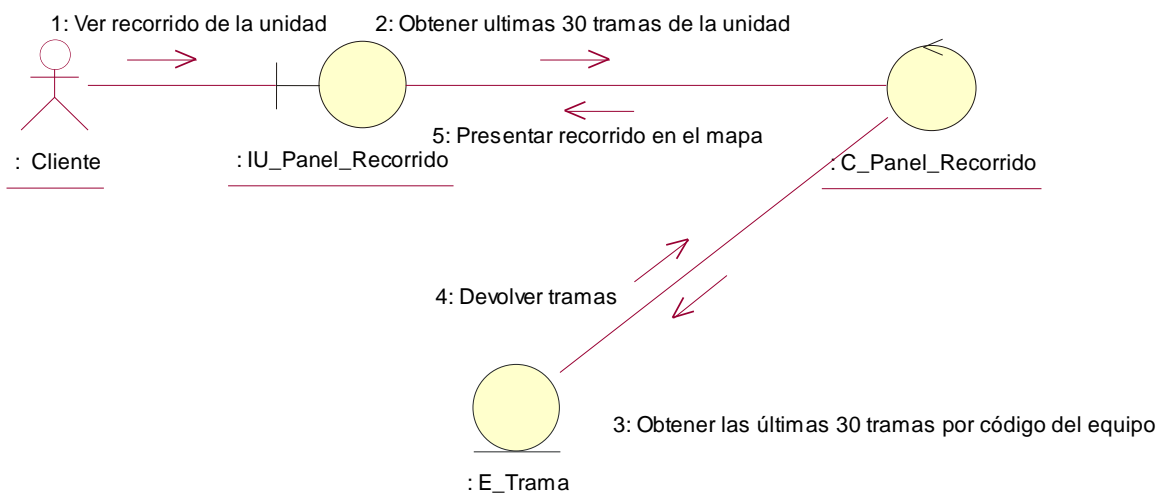


Figura 4.7 Diagrama de colaboración “Ver Recorrido”

**Descripción del diagrama de colaboración – Ver Recorrido:** Una vez que el usuario selecciona la opción “Ver Recorrido de la Unidad” de la lista de unidades, se le solicita a la interfaz de usuario IU\_Panel\_Recorrido que presente el recorrido de la unidad, esta clase le solicita la clase controladora C\_Panel\_Recorrido las últimas 30 tramas de la unidad señalada entregándole el código del equipo, la clase controladora le solicita a la clase entidad E\_Trama que devuelva las última 30 tramas emitidas por el equipo, esta clase le devuelve las últimas 30 tramas a la clase controladora y le presenta los datos a la clase IU\_Panel\_Recorrido, la clase IU\_Panel\_Recorrido presenta el recorrido en el mapa y en el panel presenta el recorrido de la unidad paginado en 30 registros por página.

#### 4.2.2.4 CUA - Ver Alertas

##### Diagrama de Clases



Figura 4.8 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Ver Alertas

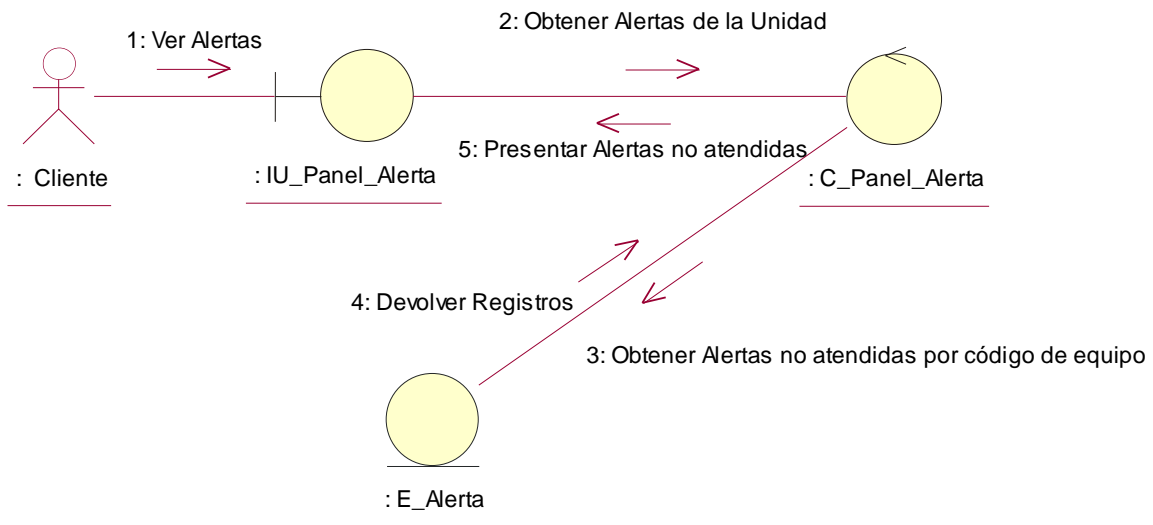


Figura 4.9 Diagrama de colaboración “Ver Alertas”

**Descripción del diagrama de colaboración – Ver Alertas:** El usuario ya sea al cargar el mapa o al refrescar el mapa hace la solicitud de ver las alertas no atendidas a la interfaz de usuario IU\_Panel\_Alerta, esta le pide a la clase controladora C\_Panel\_Alerta que obtenga las alertas de la unidad, la clase controladora le solicita a la clase entidad E\_Alerta que entregue las alertas no atendidas de la unidad por el código del equipo, la clase E\_Alerta devuelve los registros a la clase controladora y la clase controladora le entrega los registros a la clase IU\_Panel\_Alerta quien presenta las alertas no atendidas en el panel.

#### 4.2.2.5 CUA - Ver Comandos

##### Diagrama de Clases

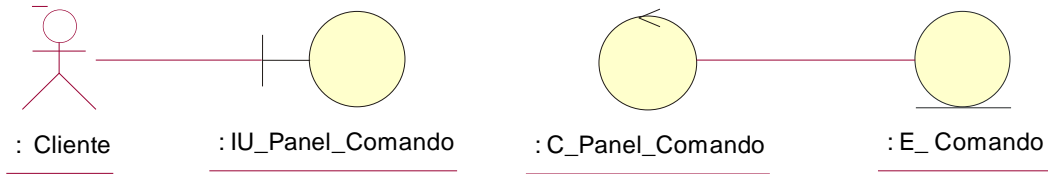


Figura 4.10 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Ver Comandos

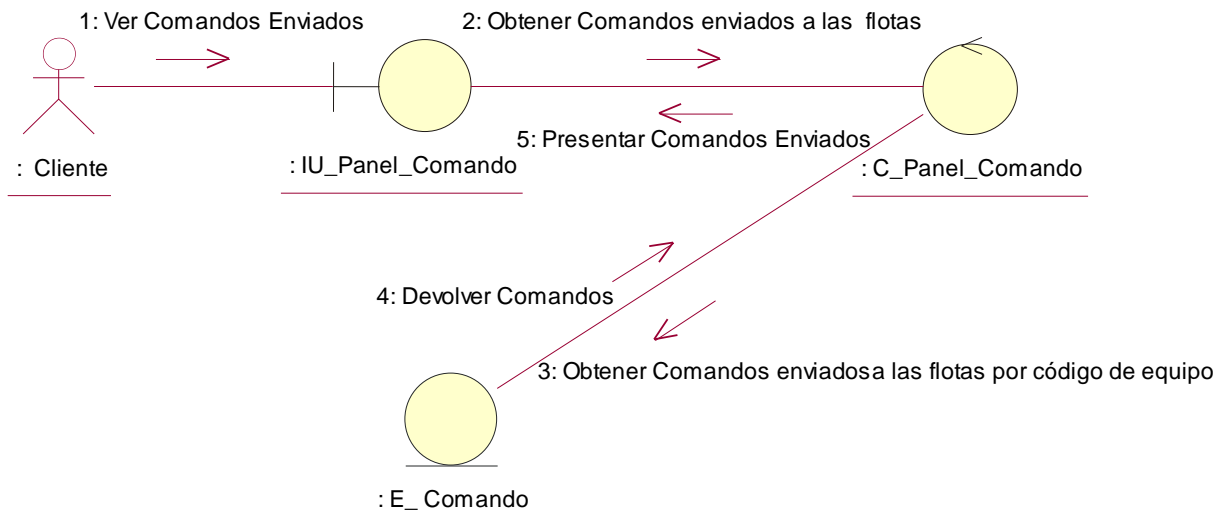


Figura 4.11 Diagrama de colaboración “Ver Comandos”

**Descripción del diagrama de colaboración – Ver Comandos:** El usuario ya sea al cargar el mapa o al refrescar el mapa hace la solicitud de ver los comandos enviados a la interfaz de usuario IU\_Panel\_Comando, esta le pide a la clase controladora C\_Panel\_Comando que obtenga comandos enviados a las flotas del usuario, la clase controladora le solicita a la clase entidad E\_Comando que entregue los comandos enviados a las flotas por código de equipo, la clase E\_Comando devuelve los registros a la clase controladora y la clase controladora le entrega los registros a

la clase IU\_Panel\_Comando quien presenta los comandos enviados y su estado en el panel.

#### 4.2.2.6 CUA – Enviar Comando

##### Diagrama de Clases

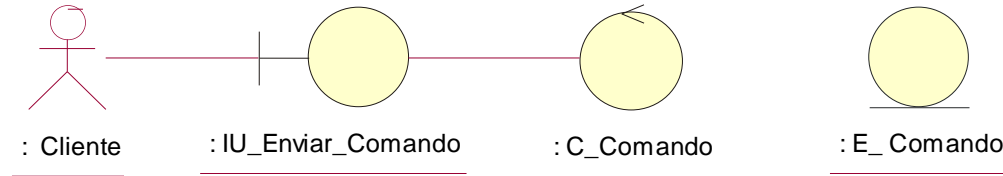


Figura 4.12 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Enviar Comando

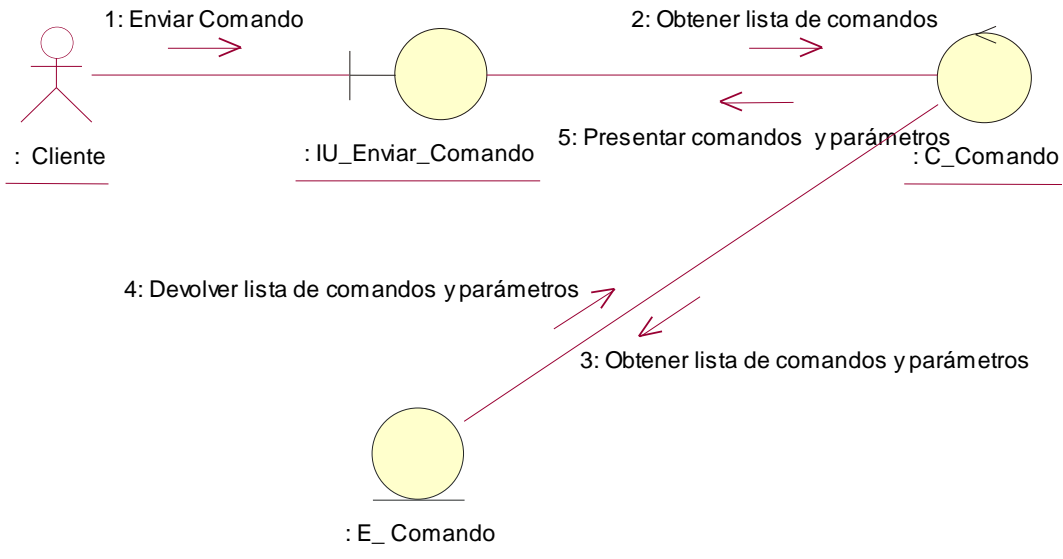


Figura 4.13 Diagrama de colaboración “Enviar Comando”

**Descripción del diagrama de colaboración – Enviar Comando:** El usuario selecciona la opción enviar comando seleccionando una unidad del panel de unidades, el sistema a través de la interfaz de usuario IU\_Enviar\_Comando, solicita a la controladora C\_Comando la lista de comandos junto con sus parámetros por comando, la clase controladora C\_Comando solicita a la clase entidad E\_Comando, que le devuelva la lista de comandos junto con sus parámetros, la clase controladora una vez que tiene los comandos y sus parámetros, devuelve los datos a la clase interfaz IU\_Enviar\_Comando quien prepara el formulario de envío de comando junto con sus parámetros.



## Diagrama de Colaboración – Confirmar Envío

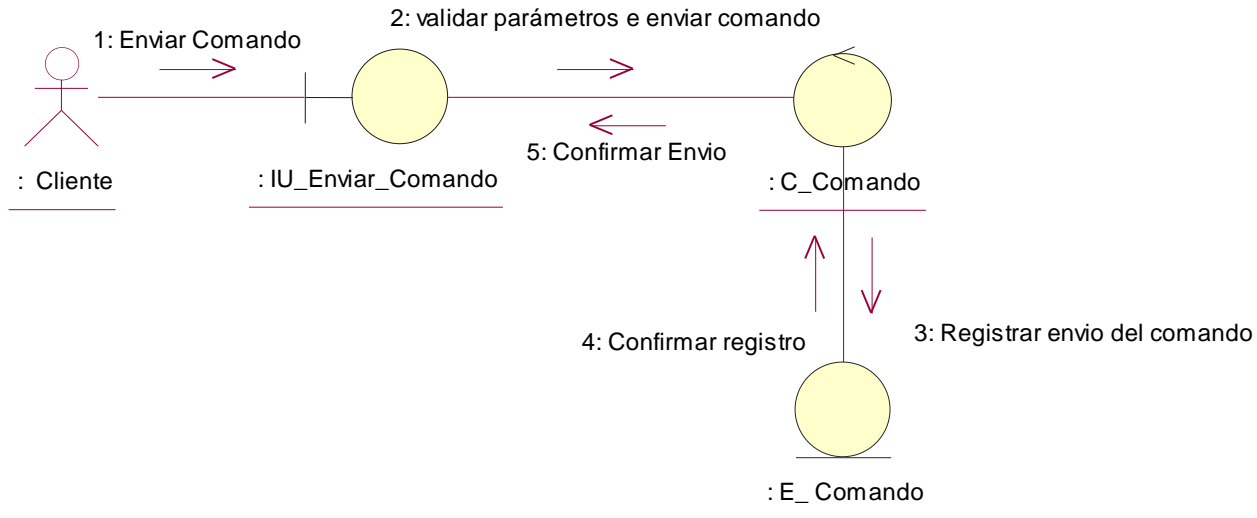


Figura 4.14 Diagrama de colaboración “Confirmar Envío”

**Descripción del diagrama de colaboración – Confirmar Envío:** El usuario una vez que llena los parámetros de envío de comando y confirma su envío, la clase interfaz de usuario IU\_Enviar\_Comando envía los datos a la clase controladora C\_Comando, quien valida si los parámetros han sido correctamente ingresados y envía el comando al equipo, una vez que el comando ha sido enviado, se comunica con la clase E\_Comando y registra el envío del comando, la clase E\_Comando confirma el registro del comando a la clase controladora y esta confirma el envío del comando a la clase interfaz IU\_Enviar comando, quien cierra la ventana de envío de comando.

### 4.2.2.7 CUA - Ver detalle de la posición

#### Diagrama de Clases

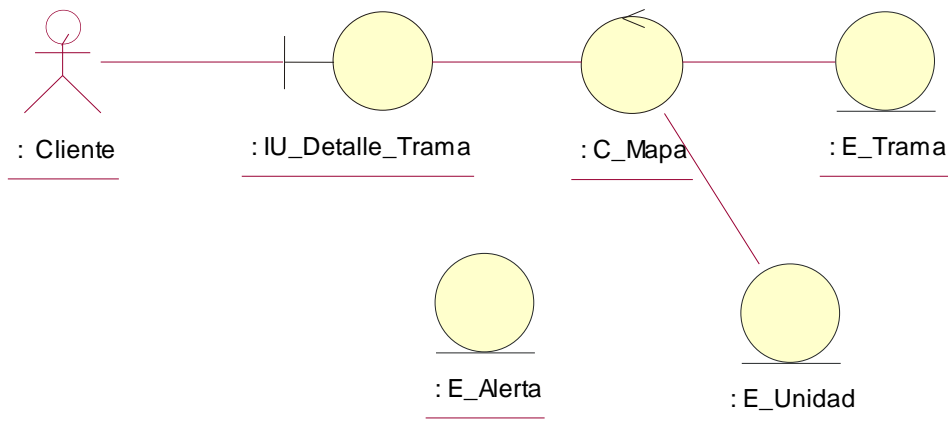
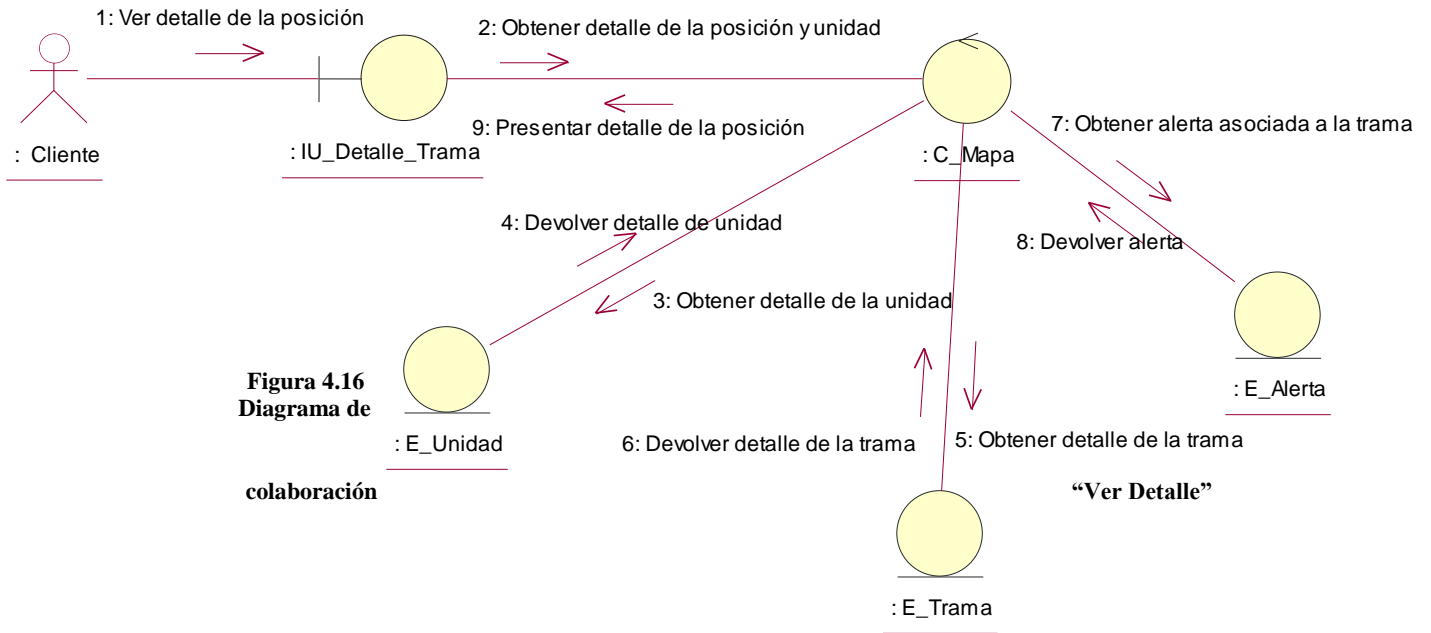


Figura 4.15 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Ver Detalle



**Descripción del diagrama de colaboración – Ver Detalle:** El usuario selecciona la opción “Ver Detalle” seleccionando una unidad del panel de unidades o del panel de recorrido, el sistema a través de la interfaz de usuario IU\_Detalle\_Trama solicita a la clase controladora C\_Mapas que obtenga el detalle de la posición y de la unidad, la clase controladora se comunica con la clase E\_Unidad a quien le solicita el detalle de la unidad a través del identificador de la unidad, así mismo se comunica con la clase E\_Trama solicitándole el detalle de la trama, una vez obtenido el detalle de la trama, identifica si hubo una alerta, de ser así se comunica con la clase E\_Alerta y solicita el detalle de la alerta, una vez que reúne toda la información, la presenta a la clase interfaz de usuario IU\_Detalle\_Trama que a su vez presenta la ventana de información al usuario.

### 4.2.2.8 CUA – Atender alerta

#### Diagrama de Clases



Figura 4.17 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Atender Alerta

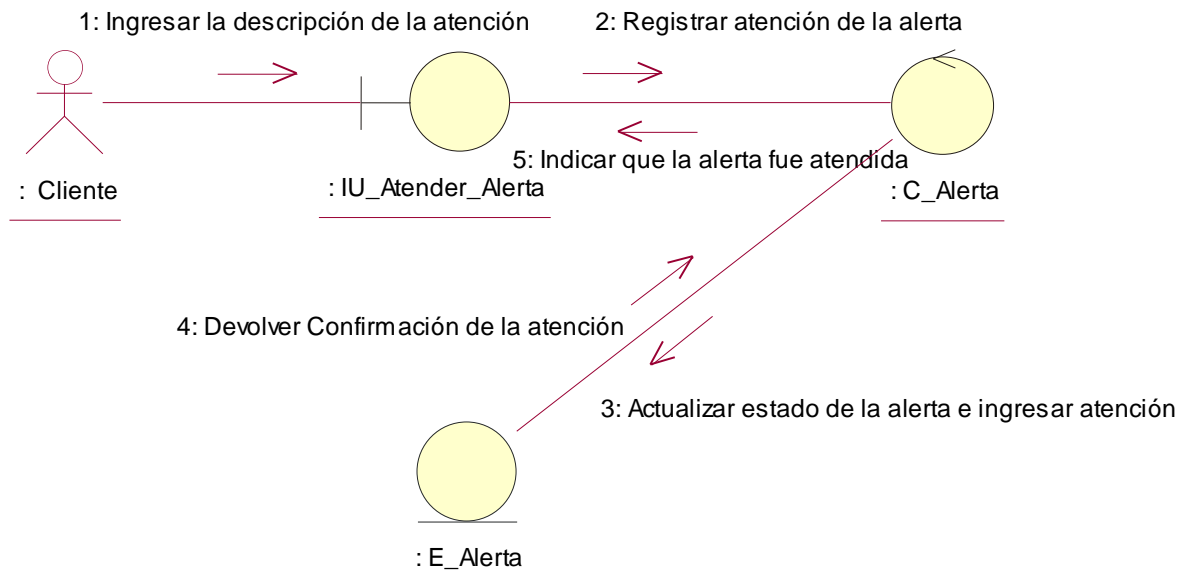


Figura 4.18 Diagrama de colaboración “Atender Alerta”

**Descripción del diagrama de colaboración – Atender Alerta:** El usuario selecciona la opción “Ver Detalle”, una vez que la ventana de detalle de la posición es cargada, en caso de ser alerta, el sistema presenta la interfaz de usuario IU\_Atender\_Alerta, cuando el usuario ingresa la descripción de la atención y selecciona la opción “Registrar”, el sistema a través de la clase IU\_Atender\_Alerta solicita el registro de la atención a la clase controladora C\_Alerta, quien a través de la entidad E\_Alerta actualiza el estado de la alerta seleccionada a “Atendida” y registra la atención, esta clase confirma el registro de los datos a la clase controladora y la clase controladora le informa a la interfaz de usuario que la alerta fue atendida, la clase interfaz de usuario IU\_Atender\_Alerta se cierra y actualiza el panel de alertas.

#### 4.2.2.9 CUA – Buscar puntos de interés

##### Diagrama de Clases

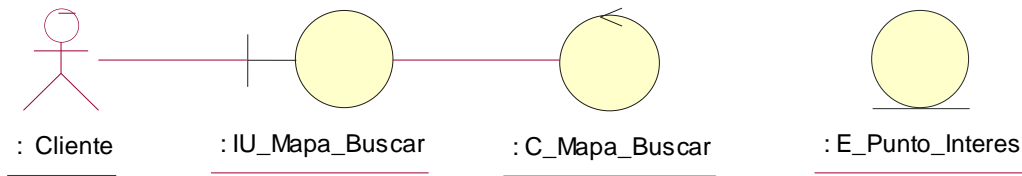


Figura 4.19 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Buscar Punto de Interés

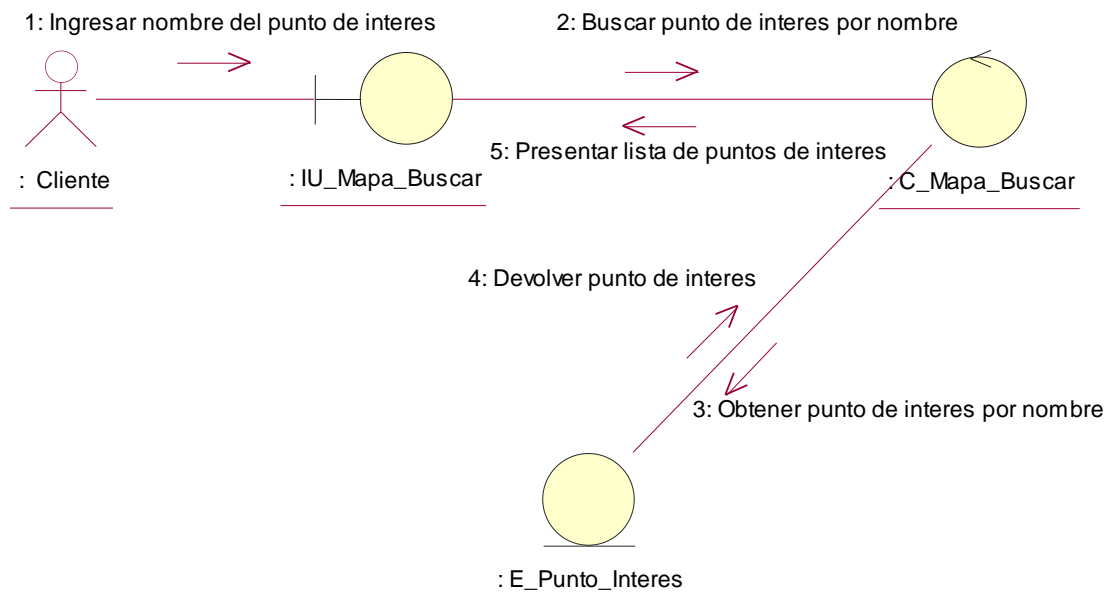


Figura 4.20 Diagrama de colaboración “Buscar Punto de Interés”

**Descripción del diagrama de colaboración – Buscar Punto de Interés:** El usuario a través de la interfaz de usuario IU\_Mapa\_Buscar, ingresa el nombre del punto de interés que desea ubicar en el mapa, el sistema a través de la clase IU\_Mapa\_Buscar invoca a la clase controladora C\_Mapa\_Buscar quien busca el punto de interés por nombre solicitándole dicha consulta a la clase entidad E\_Punto\_Interés, la clase E\_Punto\_Interés devuelve los puntos de interés que coinciden con el nombre ingresado, la clase controladora entrega la lista de registro a la interfaz de usuario IU\_Mapa\_Buscar, quien se las presenta al usuario, el usuario cuando selecciona un punto de interés de la lista, la clase IU\_Mapa\_Buscar actualiza el centro del mapa en función a la latitud y longitud del mapa ingresado.

#### 4.2.2.10 CUA – Buscar unidad

##### Diagrama de Clases

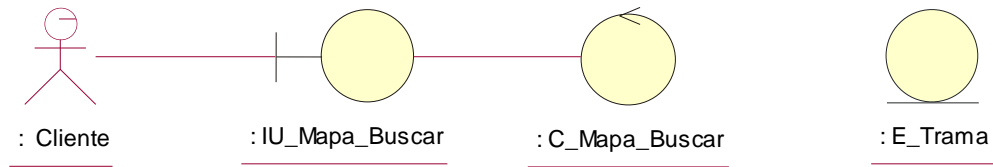


Figura 4.21 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Buscar Unidad

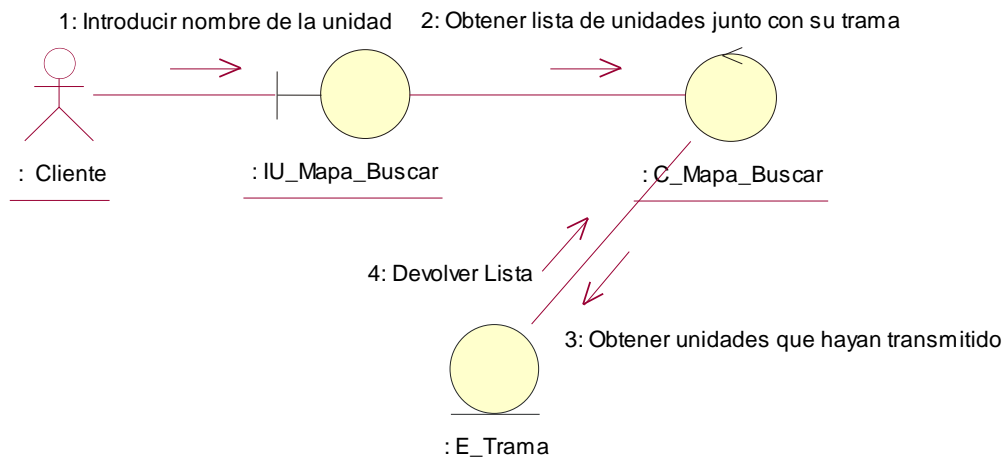


Figura 4.22 Diagrama de colaboración “Buscar Unidad”

**Descripción del diagrama de colaboración – Buscar Unidad:** El usuario a través de la interfaz de usuario IU\_Mapa\_Buscar, ingresa la placa de la unidad, la clase IU\_Mapa\_Buscar invoca a la clase controladora C\_Mapa\_Buscar solicitándole la lista de unidades que hayan transmitido, la clase controladora se comunica con la clase entidad E\_Trama quien devuelve la lista de unidades junto con su última transmisión, la clase controladora presenta los registros a la interfaz de usuario IU\_Mapa\_Buscar, cuando el usuario selecciona a alguna unidad de la lista, la clase IU\_Mapa\_Buscar centra el mapa en función a la latitud y longitud de la unidad seleccionada.

#### 4.2.2.11 CUA – Ver puntos de interés

##### Diagrama de Clases



Figura 4.23 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Ver Punto de Interés

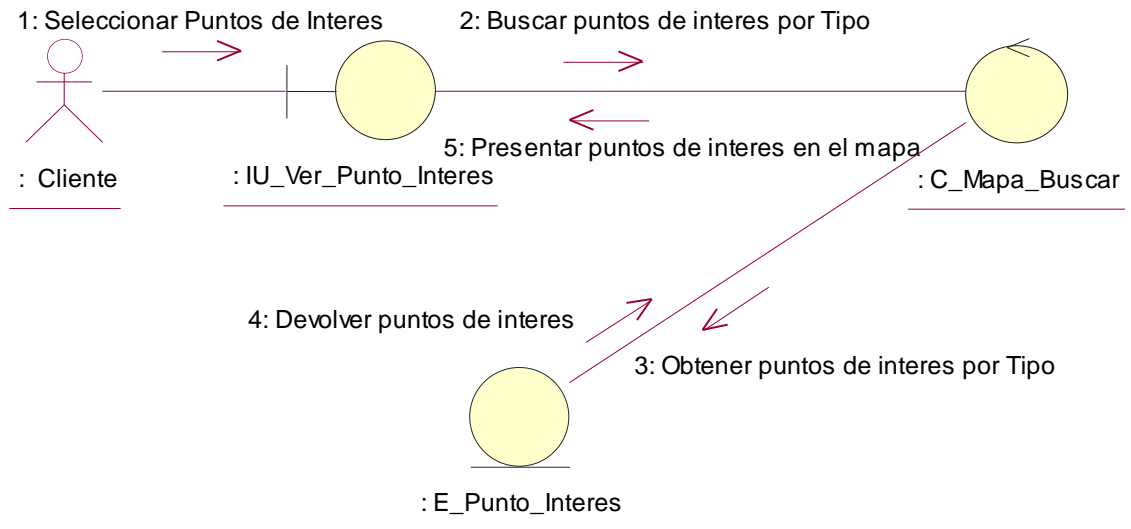


Figura 4.24 Diagrama de colaboración “Ver Punto de Interés”

**Descripción del diagrama de colaboración – Ver Punto de Interés:** El usuario a través de la interfaz de usuario IU\_Mapa\_Buscar, selecciona la opción “Punto de Interés”, el sistema invoca a la interfaz de usuario IU\_Ver\_Punto\_Interés quien presenta los tipos de puntos de interés al usuario, el usuario selecciona los puntos de interés y la interfaz de usuario IU\_Ver\_Punto\_Interés a través de la clase controladora C\_Mapa\_Buscar busca los puntos de interés asociados por tipo, la clase controladora invoca a la clase entidad E\_Punto\_Interés solicitándole los puntos de interés asociados a la cuenta cliente del usuario con los tipos de puntos de interés solicitados, la clase entidad E\_Punto\_Interés entrega la lista de puntos de interés a la clase controladora, la clase controladora C\_Mapa\_Buscar entrega los registros a la clase interfaz de usuario IU\_Ver\_Punto\_Interés quien presenta los puntos de interés en el mapa.

#### 4.2.2.12 CUA – Seleccionar cliente en el mapa

##### Diagrama de Clases

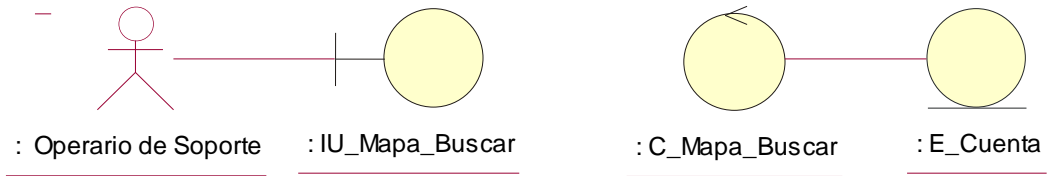


Figura 4.25 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Seleccionar Cliente en el Mapa

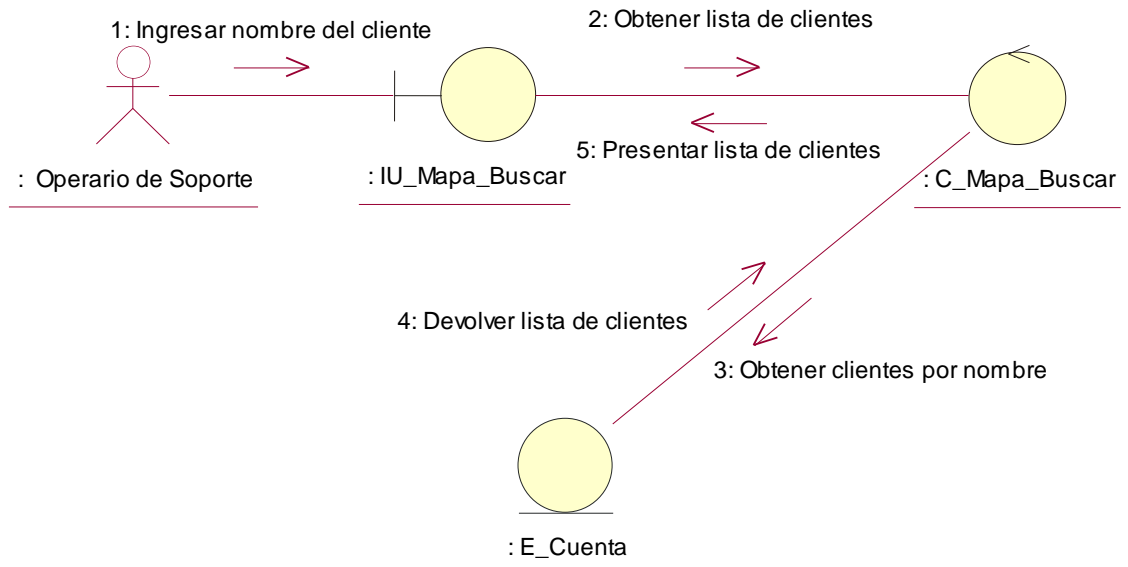


Figura 4.26 Diagrama de colaboración “Seleccionar Cliente en el Mapa”

**Descripción del diagrama de colaboración – Seleccionar Cliente en el Mapa:** El usuario a través de la interfaz de usuario IU\_Mapa\_Buscar, ingresa el nombre de la cuenta cliente a fin de visualizar las unidades de esta, la clase interfaz solicita la lista de cliente a la clase controladora C\_Mapa\_Busca, esta clase se conecta con la clase entidad E\_Cuenta a fin de obtener la lista de clientes que coinciden con el nombre ingresado, la clase E\_Cuenta devuelve la lista de clientes a la controladora, la controladora devuelve la lista de clientes a la interfaz quien se las presenta al usuario del sistema, el usuario selecciona la cuenta cliente, y el mapa es cargado en función a la nueva cuenta seleccionada.

## 4.2.3 Realización de casos de uso de análisis – Paquete Configuración

### 4.2.3.1 CUA – Mantenimiento de modelos de equipos

#### Diagrama de Clases

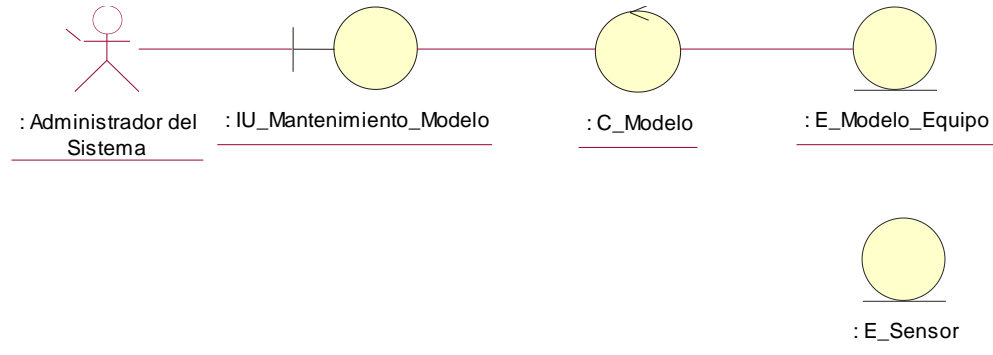


Figura 4.27 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Listar

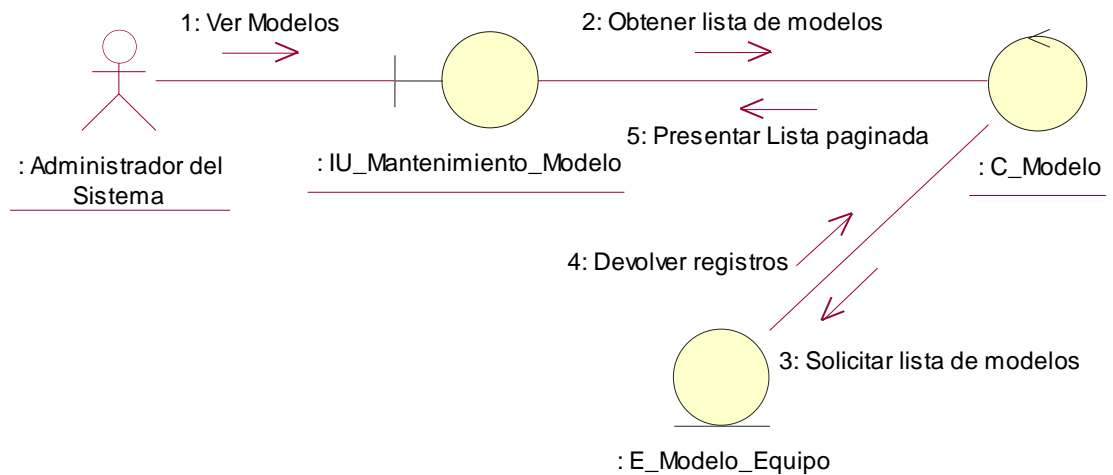


Figura 4.28 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Modelo solicita ver la lista de modelos de equipos, la interfaz de usuario IU\_Mantenimiento\_Modelo solicita a la clase controladora C\_Modelo la lista de modelos del sistema, la clase controladora se comunica con la clase entidad E\_Modelo\_Equipo quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de modelos a la clase IU\_Mantenimiento\_Modelo de forma paginada.



### Diagrama de Colaboración – Registrar

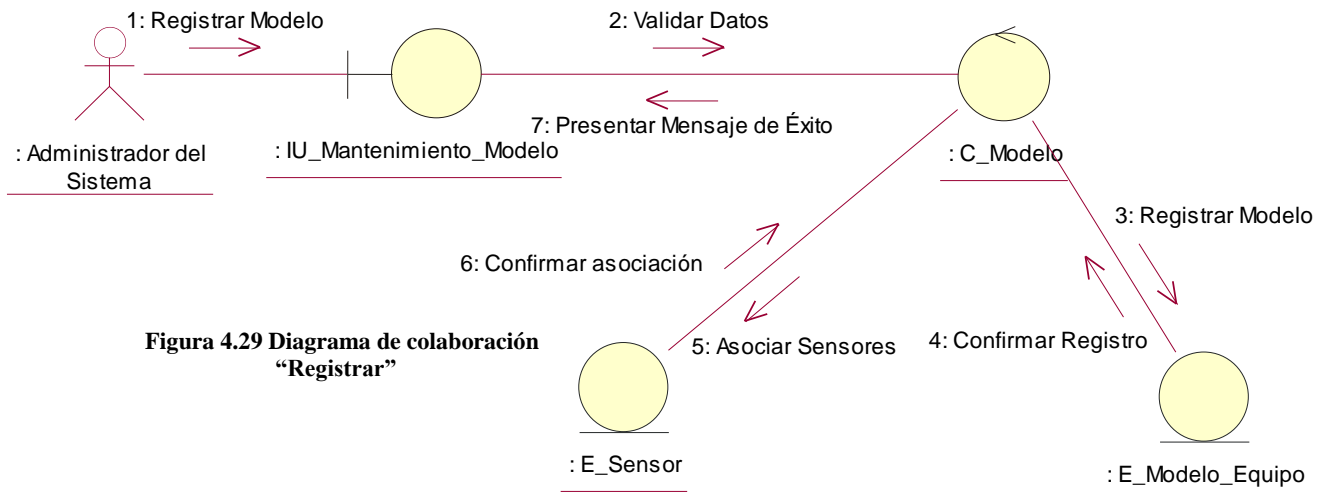


Figura 4.29 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Modelo ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Modelo que valida los datos ingresados verificando que el nombre de modelo no se repita, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Modelo y registra el modelo, la clase entidad E\_Modelo confirma el registro y la clase controladora envía la lista de sensores asociados a la clase E\_Sensor quien asocia los sensores a los modelos, luego, la clase controladora confirma la asociación a la clase IU\_Mantenimiento\_Modelo, quien presenta un mensaje al usuario confirmando el resultado de la operación.

### Diagrama de Colaboración – Eliminar

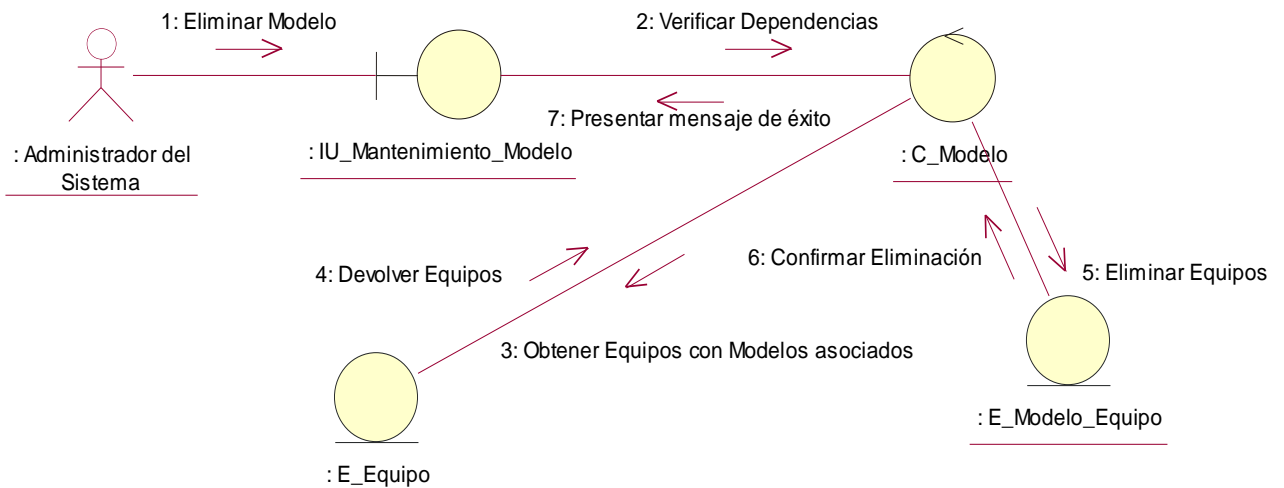


Figura 4.30 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Modelo selecciona la opción eliminar un modelo de equipo, la clase interfaz de usuario invoca a clase controladora C\_Modelo que verifica que el modelo de equipo no tenga equipos asociados consultándolo a la clase entidad E\_Equipo, la clase E\_Equipo devuelve la lista de equipos asociados, en caso de que no hayan registros asociados solicita a la clase entidad E\_Modelo\_Equipo eliminar el modelo, la clase entidad E\_Modelo\_Equipo confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Modelo.

#### 4.2.3.2 CUA – Asociar comando a modelo de equipo

##### Diagrama de Clases

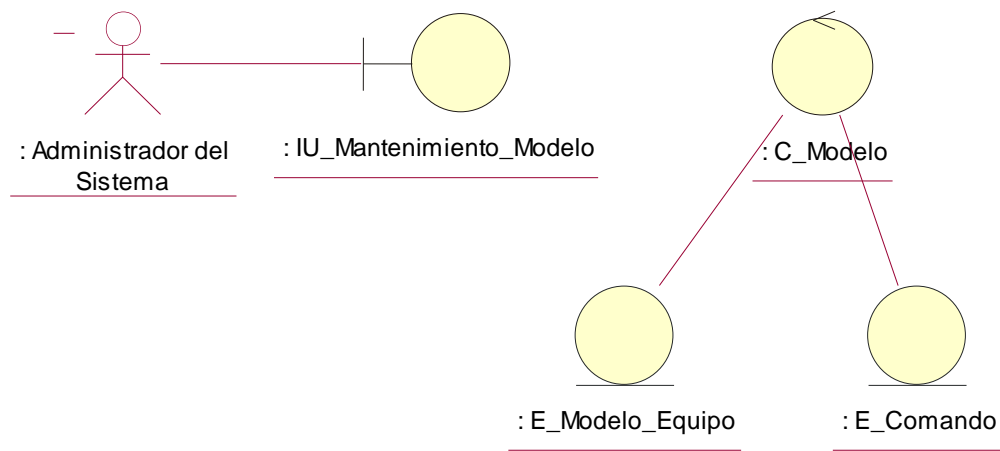
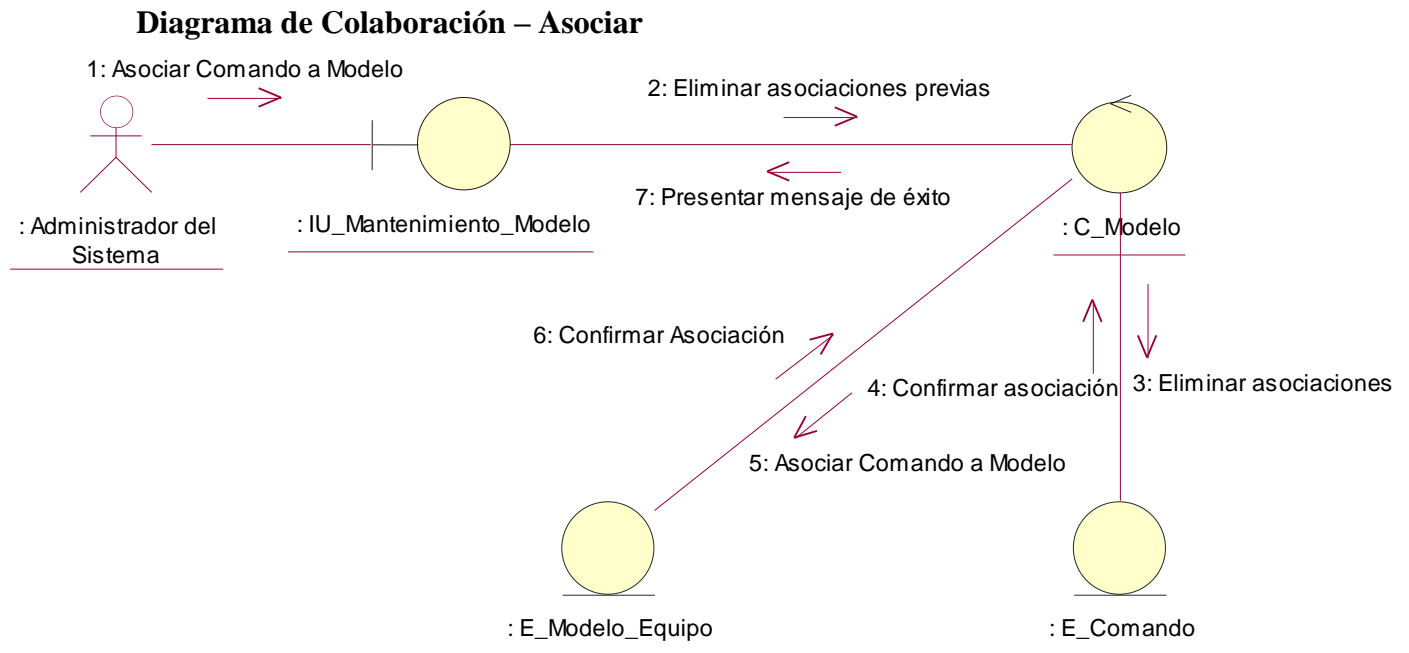


Figura 4.31 Diagrama de clases del caso de uso



**Figura 4.32 Diagrama de colaboración “Asociar”**

**Descripción del diagrama de colaboración – Asociar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Modelo selecciona los comandos que deberán estar asociados al modelo del equipo, la interfaz de usuario solicita la asociación a la clase controladora C\_Modelo quien elimina las asociaciones previas a través de la clase entidad E\_Comando, una vez que la clase entidad E\_Comando confirma la eliminación, la clase controladora asocia los comandos al modelo solicitando dicha operación a la clase entidad E\_Modelo\_Equipo, la clase E\_Modelo\_Equipo confirma la asociación a la clase controladora quien solicita presentar un mensaje de confirmación de operación a la clase IU\_Mantenimiento\_Modelo.

### 4.2.3.3 CUA – Mantenimiento de comandos

#### Diagrama de Clases

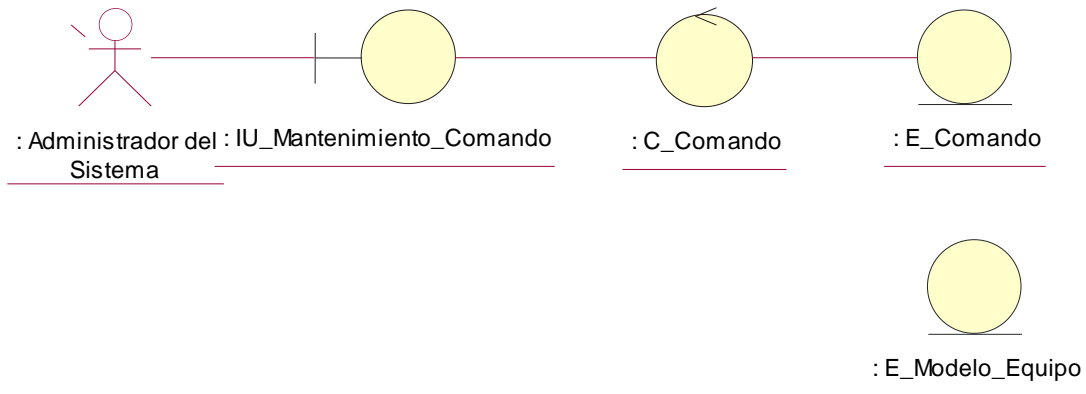


Figura 4.33 Diagrama de clases del caso de uso

#### Diagrama de Colaboración - Listar

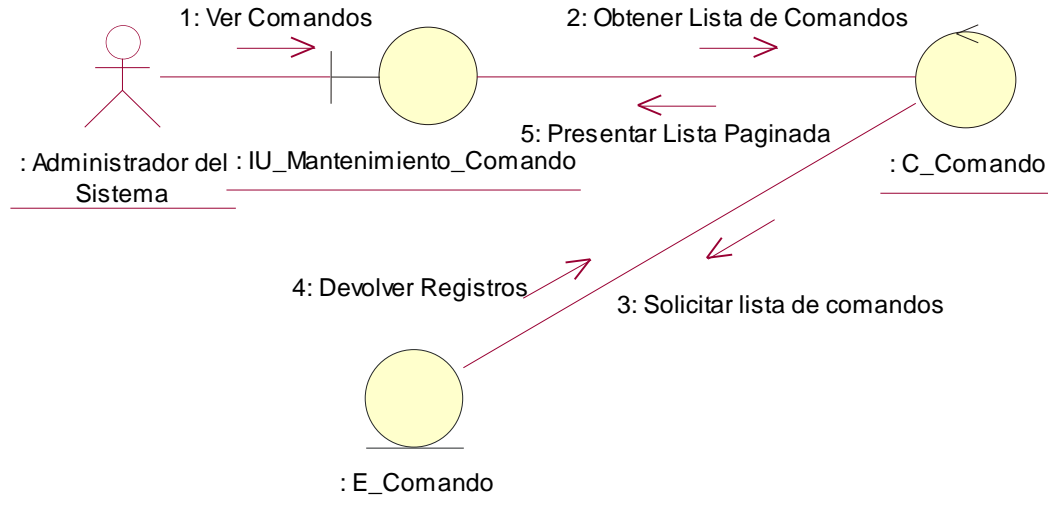


Figura 4.34 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Comando solicita ver la lista de comandos, la interfaz de usuario IU\_Mantenimiento\_Comando solicita a la clase controladora C\_Comando la lista de comandos del sistema, la clase controladora se comunica con la clase entidad E\_Comando quien devuelve los registros a la clase controladora, la

clase controladora devuelve la lista de comandos a la clase IU\_Mantenimiento\_Modelo de forma paginada.

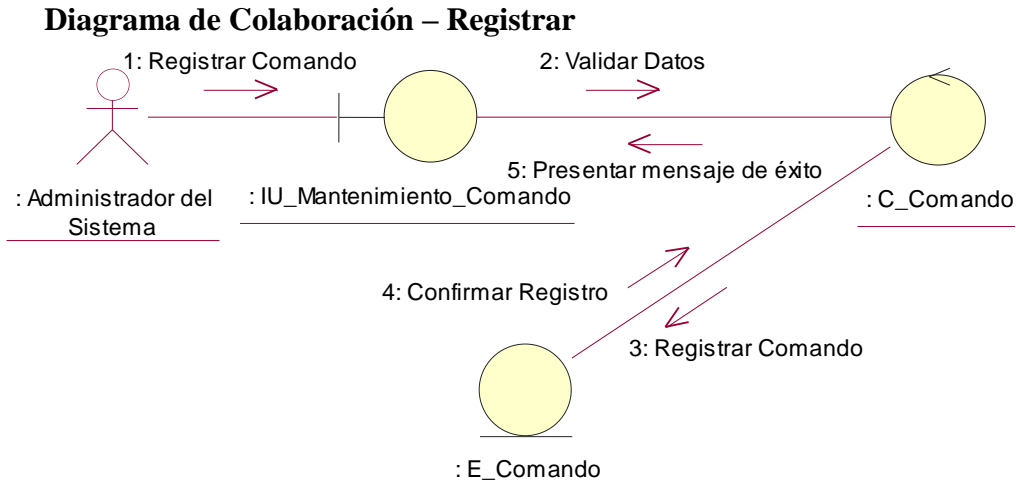


Figura 4.35 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Comando ingresa los datos del comando, la lista de argumento y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Comando que valida los datos ingresados verificando que el nombre del comando no se repita y que los parámetros sean correctos, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Comando y registra el comando, la clase entidad E\_Comando confirma el registro y la clase controladora solicita confirmar la operación a la interfaz IU\_Mantenimiento\_Comando.

### Diagrama de Colaboración – Eliminar

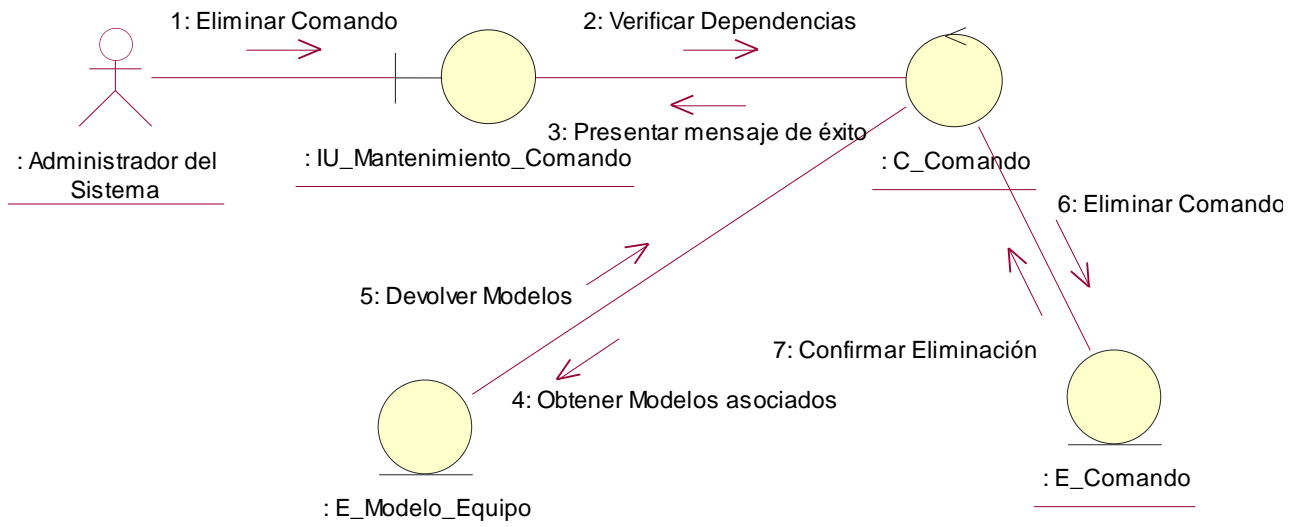


Figura 4.36 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Comando selecciona la opción eliminar un comando del sistema, la clase interfaz de usuario invoca a clase controladora C\_Comando que verifica que el comando no tenga modelos de equipos asociados consultándolo a la clase entidad E\_Modelo\_Equipo, la clase E\_Modelo\_Equipo devuelve la lista de modelos asociados, en caso de que no hayan registros asociados solicita a la clase entidad E\_Comando eliminar el comando, la clase entidad E\_Comando confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Comando.

### 4.2.3.4 CUA – Mantenimiento de eventos

#### Diagrama de Clases

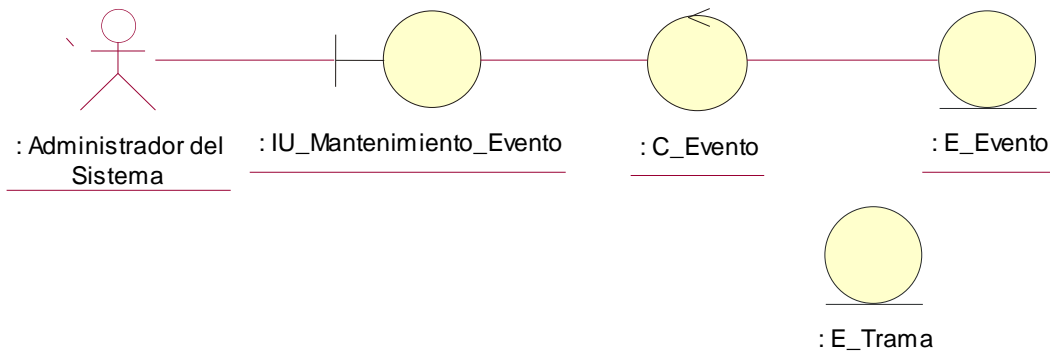


Figura 4.37 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Listar

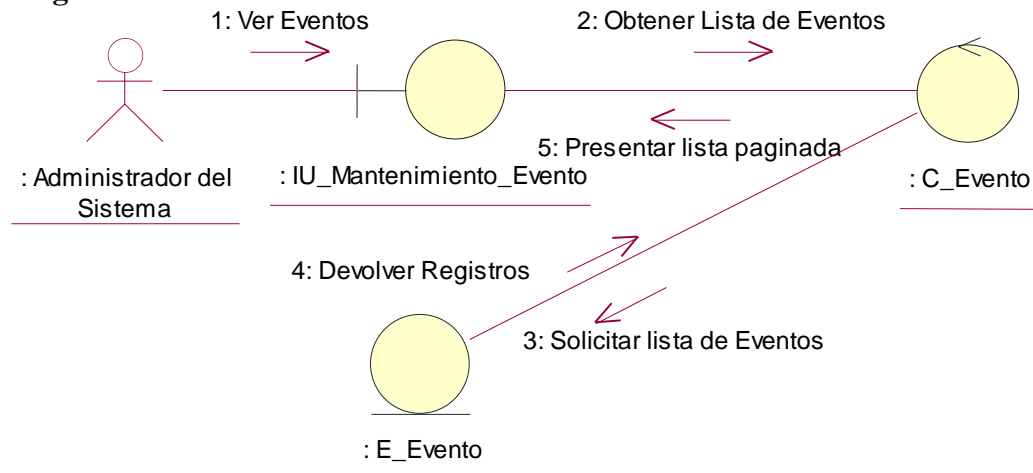


Figura 4.38 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Evento solicita ver la lista de eventos, la interfaz de usuario IU\_Mantenimiento\_Evento solicita a la clase controladora C\_Evento la lista de eventos del sistema, la clase controladora se comunica con la clase entidad E\_Evento quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de eventos a la clase IU\_Mantenimiento\_Evento de forma paginada.

### Diagrama de Colaboración – Registrar

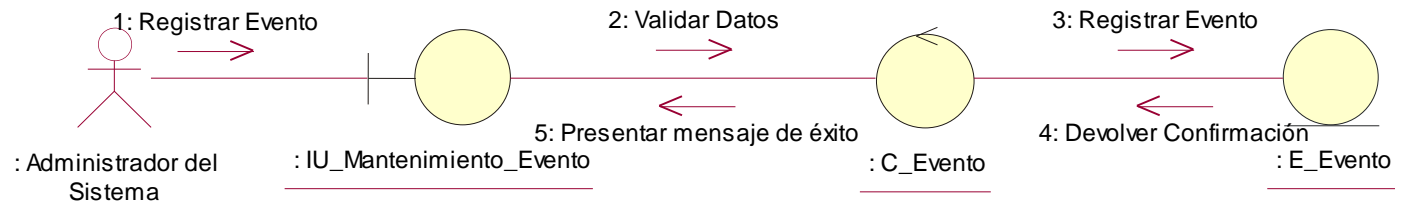


Figura 4.39 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Evento ingresa los datos del evento y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Evento que valida los datos ingresados verificando que el nombre del evento no se repita, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Evento y registra el evento, la clase entidad E\_Evento confirma el registro y la clase controladora solicita confirmar la operación a la interfaz IU\_Mantenimiento\_Evento.

## Diagrama de Colaboración – Eliminar

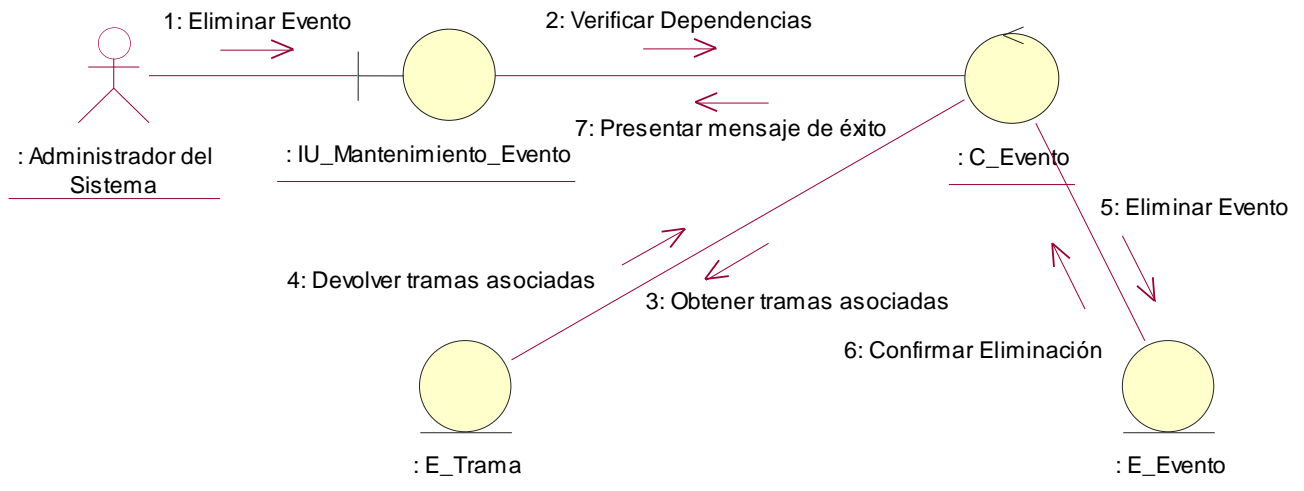


Figura 4.40 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Evento selecciona la opción eliminar un evento del sistema, la clase interfaz de usuario invoca a clase controladora C\_Evento que verifica que el evento no tenga tramas asociadas consultándolo a la clase entidad E\_Trama, la clase E\_Trama devuelve la lista de tramas que contienen el evento, en caso de que no hayan registros asociados solicita a la clase entidad E\_Evento eliminar el evento, la clase entidad E\_Evento confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Comando.



### 4.2.3.5 CUA – Mantenimiento de Tipos de Puntos de Interés

#### Diagrama de Clases



Figura 4.41 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Listar

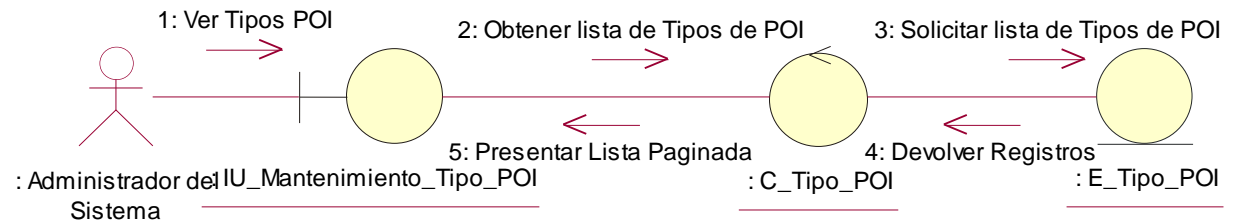


Figura 4.42 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Tipo\_POI solicita ver la lista de Tipos de Puntos de Interés, la interfaz de usuario IU\_Mantenimiento\_Tipo\_POI solicita a la clase controladora C\_Tipo\_POI la lista de Tipos de Puntos de Interés del sistema, la clase controladora se comunica con la clase entidad E\_Tipo\_POI quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de Tipos de Puntos de Interés a la clase IU\_Mantenimiento\_Tipo\_POI de forma paginada.

#### Diagrama de Colaboración – Registrar

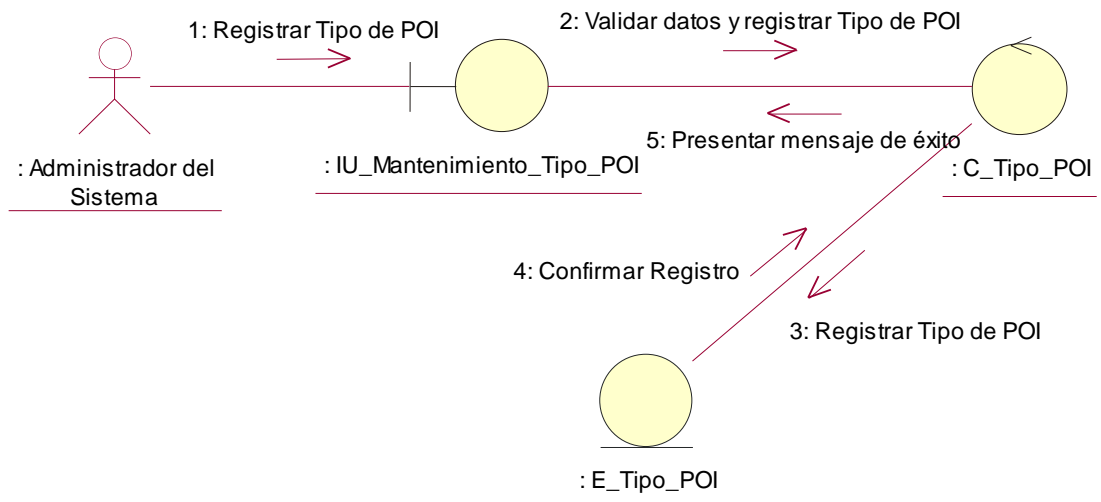


Figura 4.43 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Tipo\_POI ingresa los datos del Tipo de Puntos de interés y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Tipo\_POI que valida los datos ingresados verificando que el nombre del Tipo de Punto de Interés no se repita, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Tipo\_POI y registra el Tipo de Punto de Interés, la clase entidad E\_Tipo\_POI confirma el registro y la clase controladora solicita confirmar la operación a la interfaz IU\_Mantenimiento\_Tipo\_POI.

### Diagrama de Colaboración – Eliminar

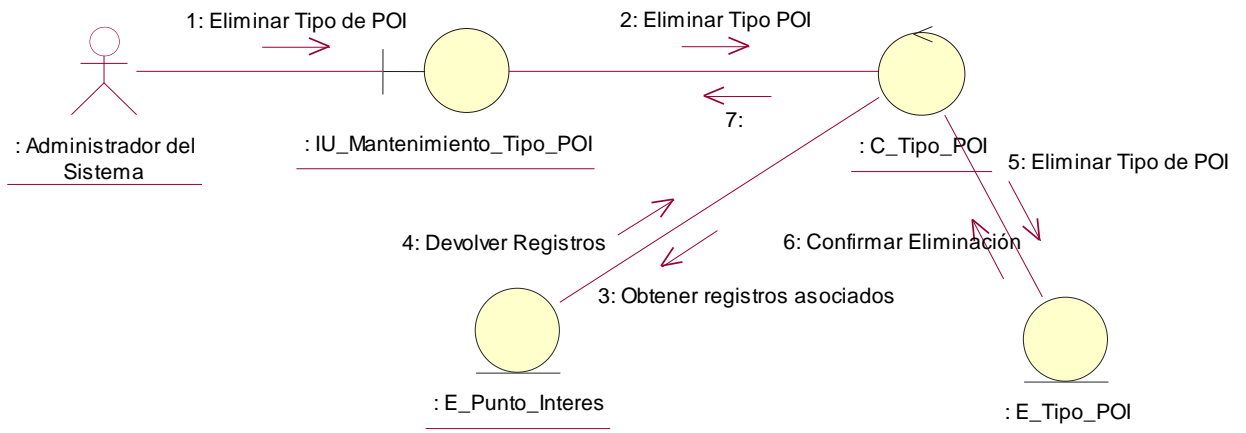


Figura 4.44 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Tipo\_POI selecciona la opción eliminar un Tipo de Punto de Interés del sistema, la clase interfaz de usuario invoca a clase controladora C\_Tipo\_POI que verifica que el Tipo de Punto de interés no tenga Puntos de Interés asociadas consultándolo a la clase entidad E\_Punto\_Interés, la clase E\_Punto\_Interés devuelve la lista de Puntos de Interés que contienen el Tipo de Punto de Interés asociado, en caso de que no hayan registros asociados solicita a la clase entidad E\_Tipo\_POI eliminar el Tipo de Punto de Interés, la clase entidad E\_Punto\_Interés confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Tipo\_POI.

## 4.2.4 Realización de casos de uso de análisis – Paquete Administración

### 4.2.4.1 CUA – Mantenimiento de clientes

#### Diagrama de Clases

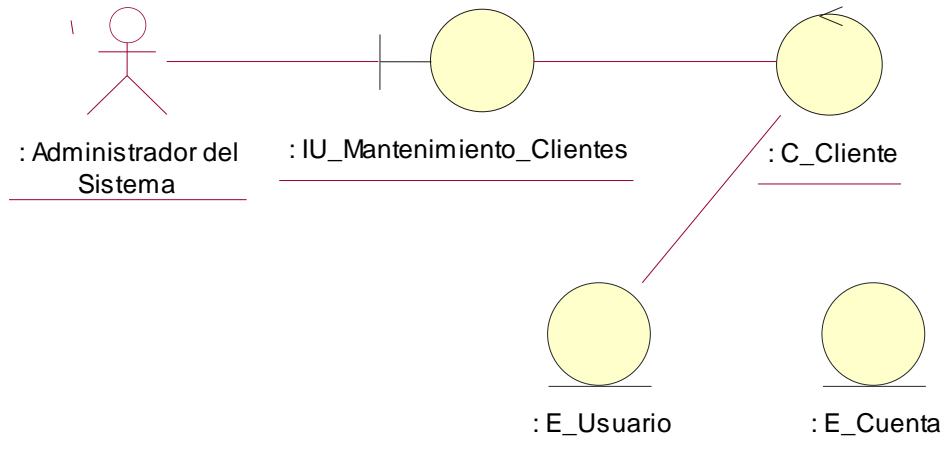


Figura 4.45 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Listar

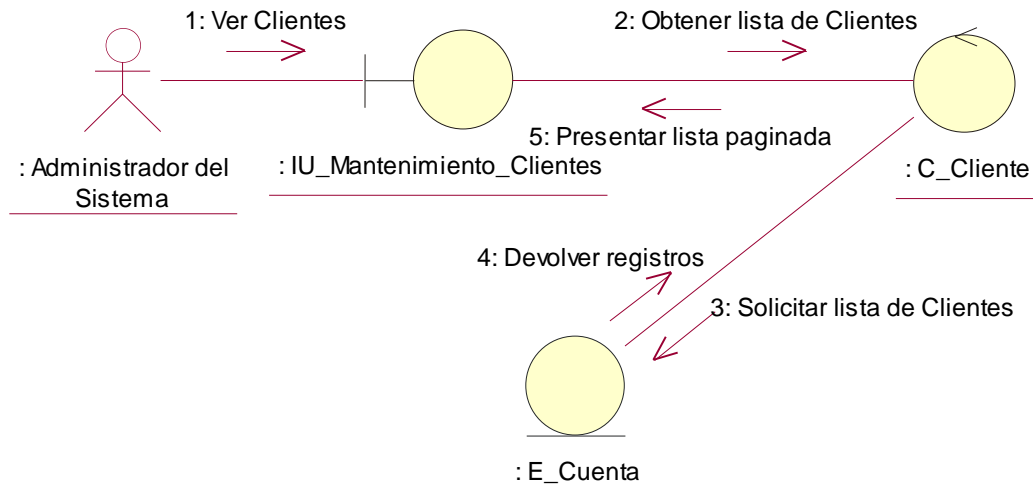
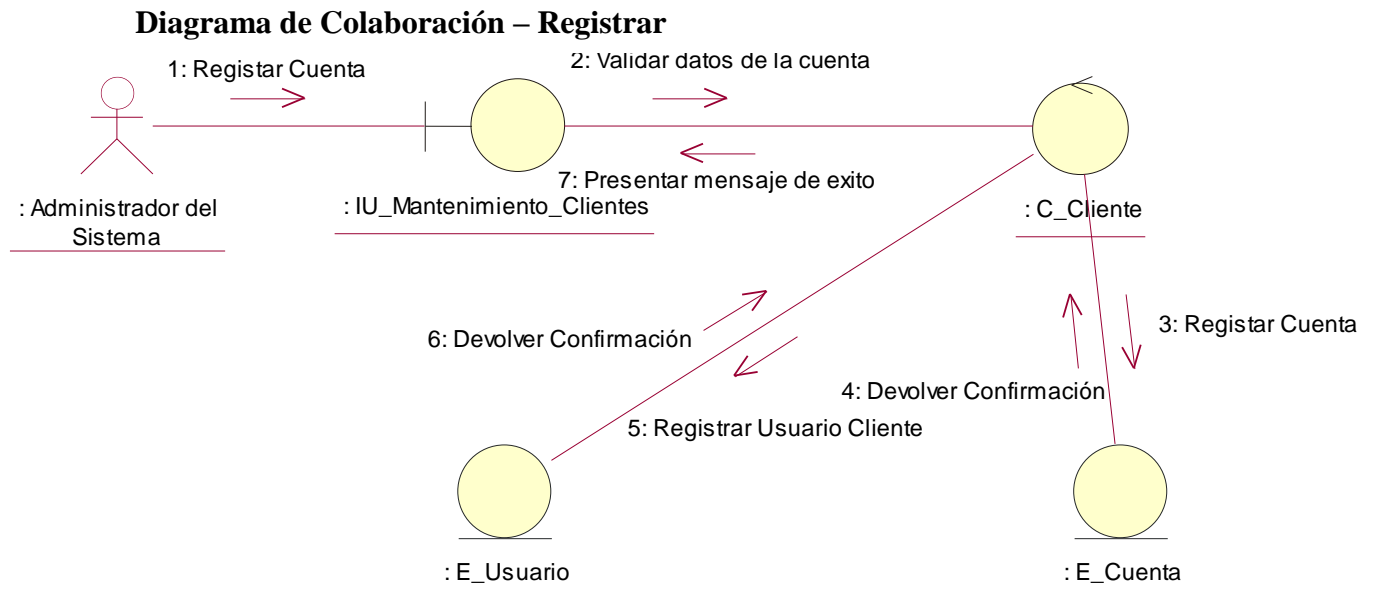


Figura 4.46 Diagrama de colaboración "Listar"

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario `IU_Mantenimiento_Clientes` solicita ver la lista de cuentas clientes, la interfaz de usuario `IU_Mantenimiento_Clientes` solicita a la clase controladora `C_Cliente` la lista de cuentas clientes registrados en el sistema, la clase controladora se comunica con la clase entidad `E_Cuenta` quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de cuentas clientes a la clase `IU_Mantenimiento_Clientes` de forma paginada.



**Figura 4.47 Diagrama de colaboración “Registrar”**

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Clientes ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Cliente que valida los datos ingresados verificando que los datos ingresados sean los correctos, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Cuenta y registra a la cuenta cliente, la clase entidad E\_Cuenta confirma el registro y la clase controladora invoca a la clase E\_Usuario a fin de registrar al usuario cliente quien administrará los registros de la cuenta cliente recién registrada, una vez que el usuario haya sido registrado, la clase controladora C\_Cuenta solicitará a la clase interfaz de usuario IU\_Mantenimiento\_Clientes presentar al usuario un mensaje de confirmación acerca de la operación.

## Diagrama de Colaboración – Configurar

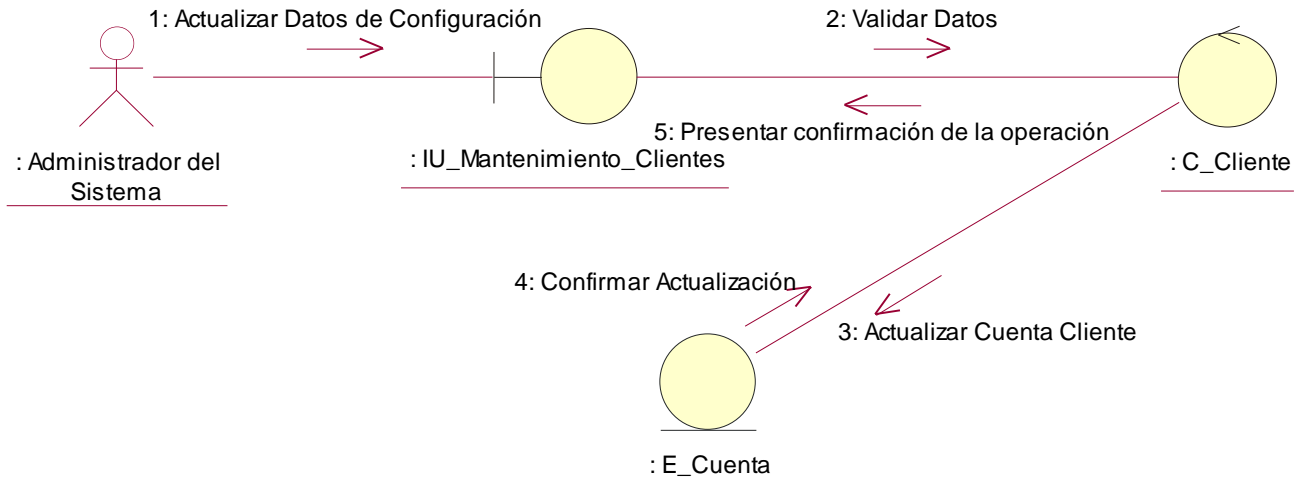


Figura 4.48 Diagrama de colaboración “Configurar”

**Descripción del diagrama de colaboración – Configurar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Clientes ingresa los datos la cantidad máxima de geocercas permitidas y la cantidad máxima de Puntos de Interés, cuando selecciona la opción “Actualizar”, la interfaz de usuario IU\_Mantenimiento\_Clientes se comunica con la clase controladora C\_Cliente que valida los datos ingresados y solicita a la clase entidad E\_Cuenta actualizar los datos, la clase entidad confirma la actualización a la clase controladora quien solicita la confirmación de la operación a la interfaz de usuario IU\_Mantenimiento\_Clientes.

### 4.2.4.2 CUA – Mantenimiento de Equipos de Localización

#### Diagrama de Clases

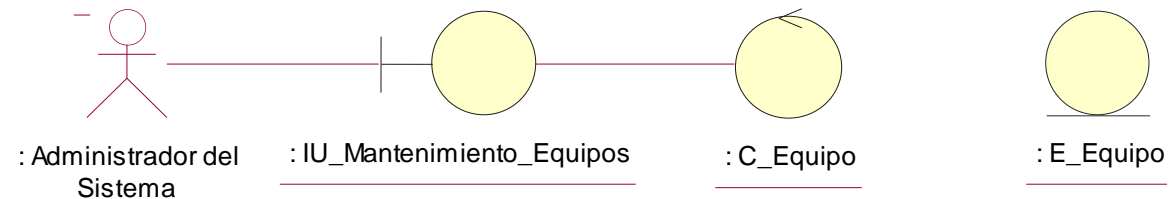


Figura 4.49 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Listar

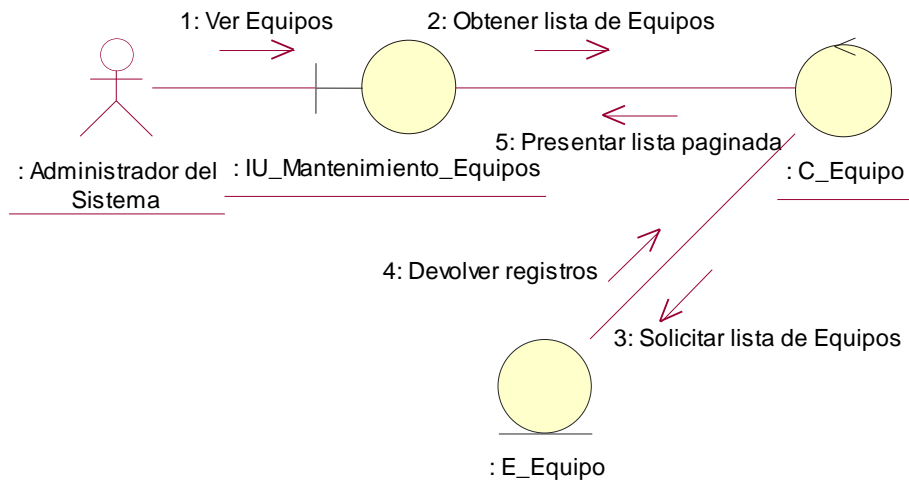


Figura 4.50 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Equipo solicita ver la lista de equipos, la interfaz de usuario IU\_Mantenimiento\_Equipo solicita a la clase controladora C\_Equipo la lista de equipos registrados en el sistema, la clase controladora se comunica con la clase entidad E\_Equipo quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de equipos a la clase IU\_Mantenimiento\_Equipos de forma paginada.

## Diagrama de Colaboración – Registrar

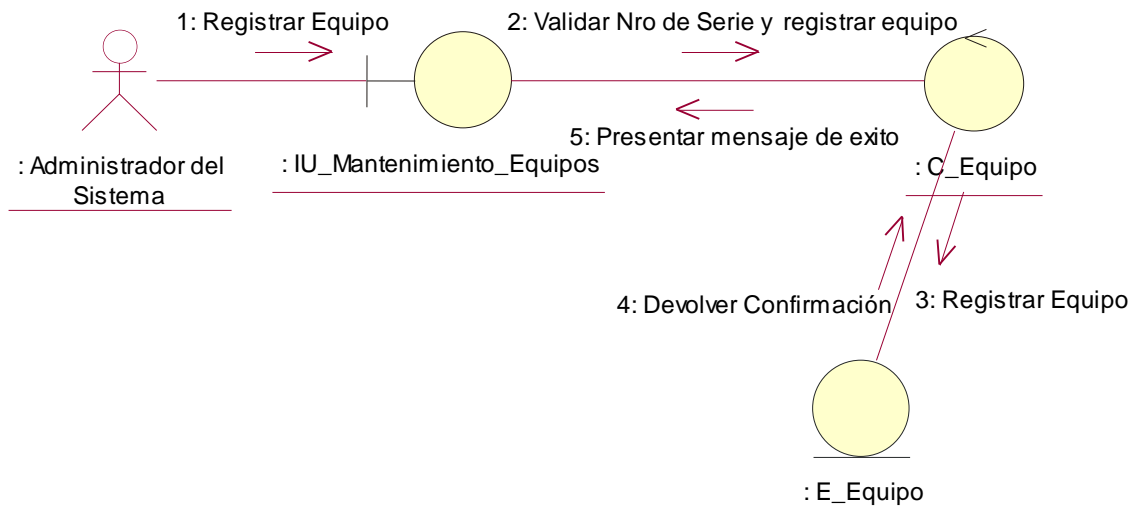


Figura 4.51 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Equipos ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Equipo que valida los datos ingresados verificando que el número de serie no se repita, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Equipo y registra el equipo, la clase entidad E\_Equipo confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Equipos presentar un mensaje al usuario confirmando el resultado de la operación.

**Diagrama de Colaboración – Eliminar**

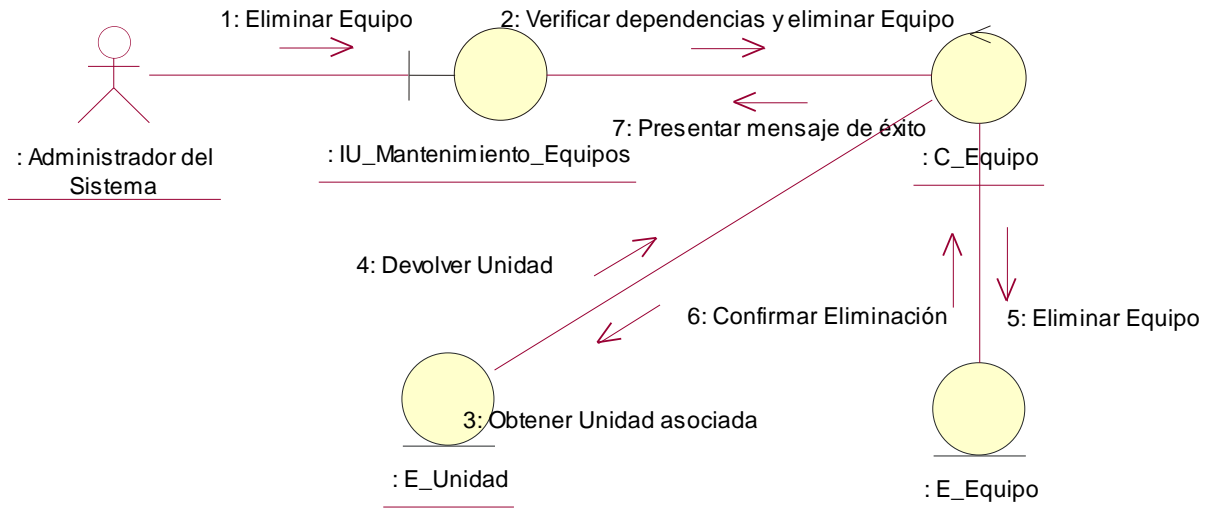


Figura 4.52 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Equipos selecciona la opción eliminar un equipo, la clase interfaz de usuario invoca a clase controladora C\_Equipo que verifica que el equipo no tenga unidades asociadas consultándolo a la clase entidad E\_Unidad, la clase E\_Unidad devuelve la unidad asociada, en caso de que no hayan registros asociados solicita a la clase entidad E\_Equipo eliminar el equipo, la clase entidad E\_Equipo confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Equipos.

### 4.2.4.3 CUA – Mantenimiento de Unidades

#### Diagrama de Clases

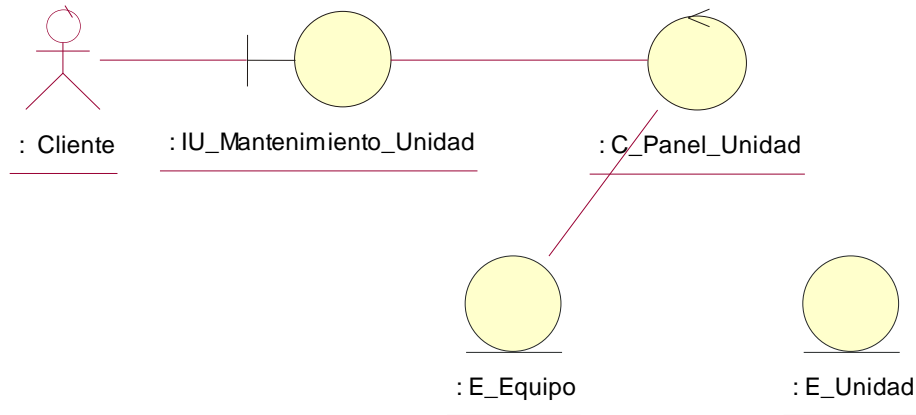


Figura 4.53 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Listar

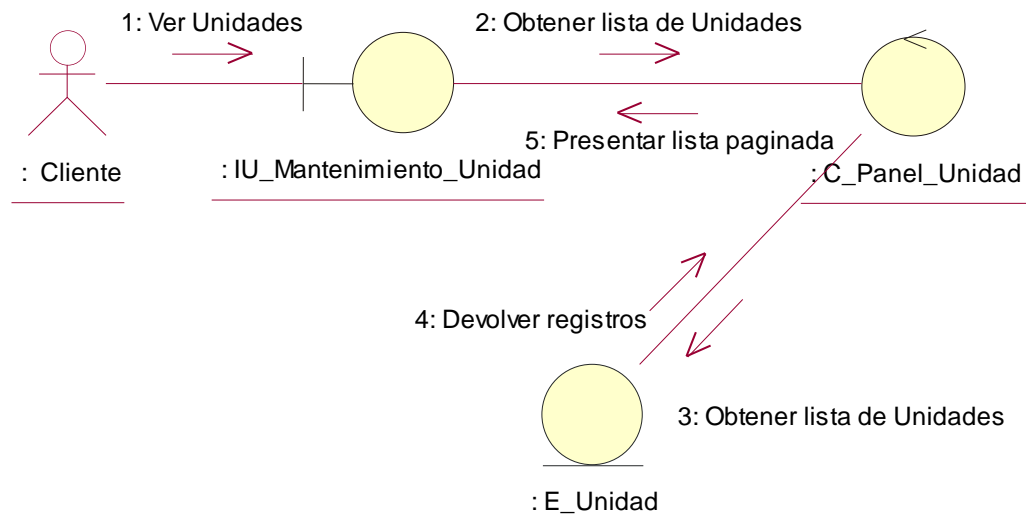


Figura 4.54 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario `IU_Mantenimiento_Unidad` solicita ver la lista de unidades vehiculares registradas en el sistema, la interfaz de usuario `IU_Mantenimiento_Unidad` solicita a la clase controladora `C_Unidad` la lista de unidades, la clase controladora se comunica con la clase entidad `E_Unidad` quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de unidad a la clase `IU_Mantenimiento_Unidad` de forma paginada.



## Diagrama de Colaboración – Registrar

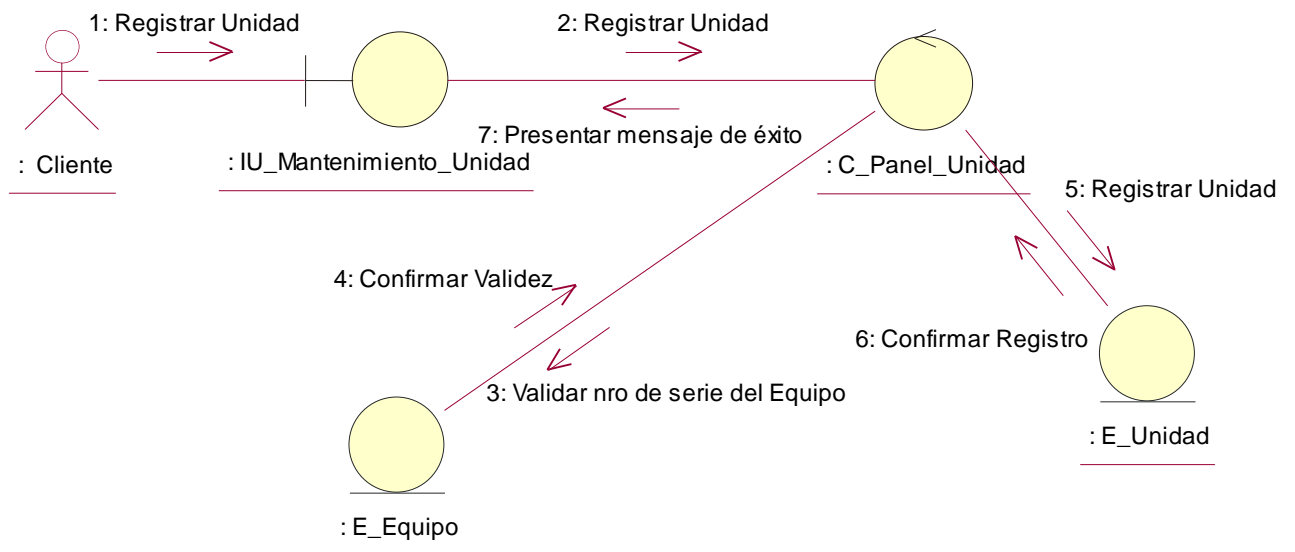


Figura 4.55 Diagrama de colaboración "Registrar"

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Unidad ingresa los datos solicitados junto con los datos del equipo para activarlo y selecciona la opción "Registrar", el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Unidad que valida los datos ingresados verificando que el número de serie sea válido solicitando a la clase entidad E\_Equipo el número de serie y si el equipo ha sido activado por otra unidad, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Unidad y registra la unidad vehicular, la clase entidad E\_Unidad confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Unidad presentar un mensaje al usuario confirmando el resultado de la operación.

## Diagrama de Colaboración – Eliminar

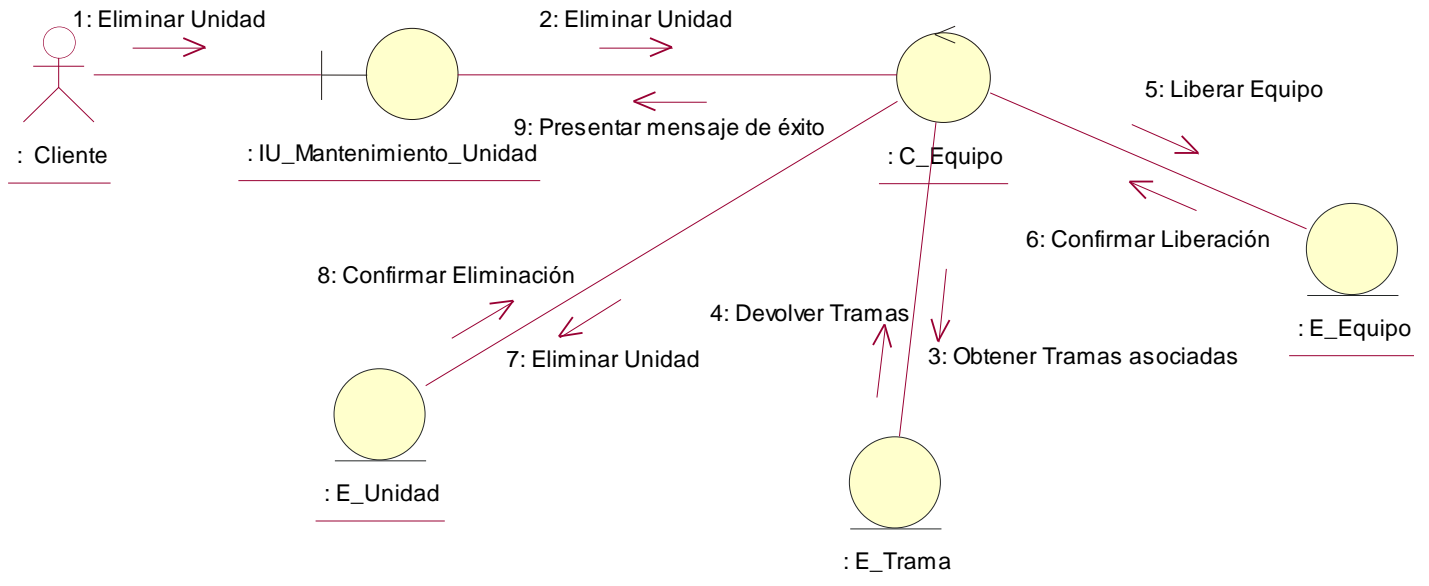


Figura 6.56 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Unidad selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_Unidad que verifica que la unidad no tenga tramas asociadas consultándolo a la clase entidad E\_Trama, la clase E\_Trama devuelve las tramas, en caso de que no hayan registros asociados solicita a la clase entidad E\_Equipo liberar el equipo cambiando su estado a inactivo, la clase E\_Equipo confirma la liberación a la clase controladora, luego la clase controladora solicita a la entidad E\_Unidad eliminar la unidad, la clase entidad E\_Unidad confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Unidades.

#### 4.2.4.4 CUA – Mantenimiento de Flotas

##### Diagrama de Clases

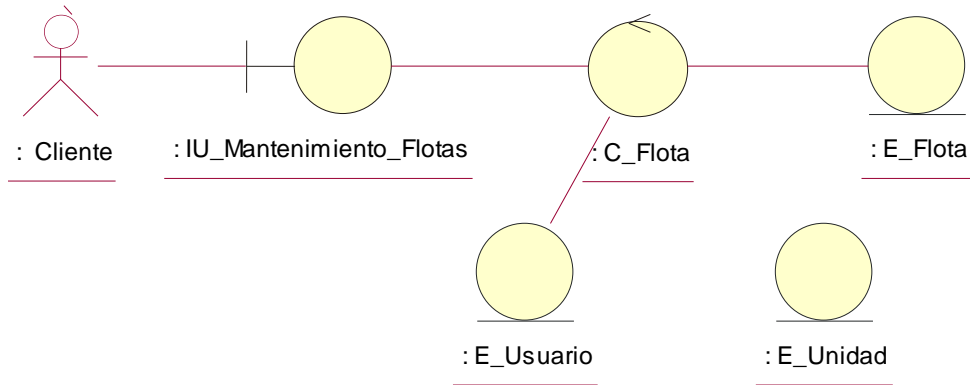


Figura 4.57 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Listar

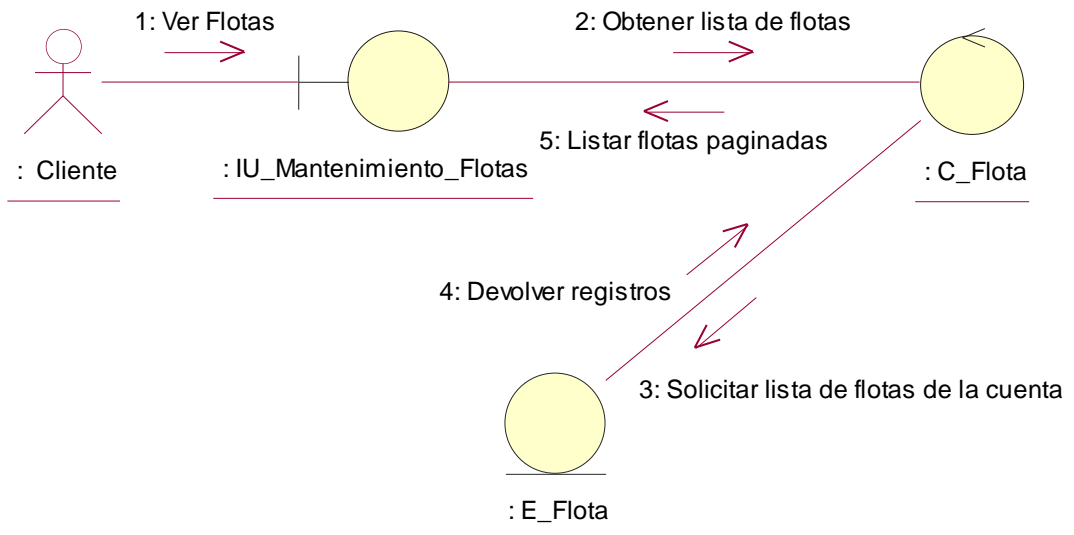


Figura 4.58 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario **IU\_Mantenimiento\_Flotas** solicita ver la lista de flotas de unidades vehiculares registradas en el sistema, la interfaz de usuario **IU\_Mantenimiento\_Flotas** solicita a la clase controladora **C\_Flota** la lista de unidades, la clase controladora se comunica con la clase entidad **E\_Flota** quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de flotas pertenecientes a la cuenta cliente del usuario a la clase **IU\_Mantenimiento\_Flota** de forma paginada.

## Diagrama de Colaboración – Registrar

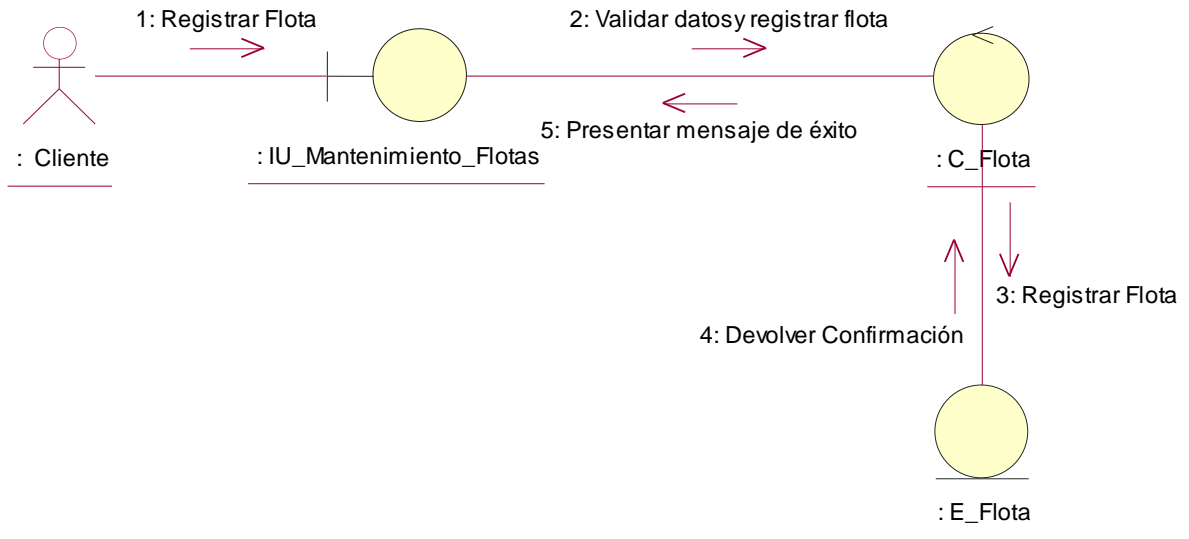


Figura 4.59 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Flota ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Flota, esta verifica que el nombre sea único en la cuenta cliente solicitando dicha confirmación a la clase entidad E\_Flota, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Flota nuevamente y registra la flota vehicular, la clase entidad E\_Flota confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Flota presentar un mensaje al usuario confirmando el resultado de la operación.

## Diagrama de Colaboración – Eliminar

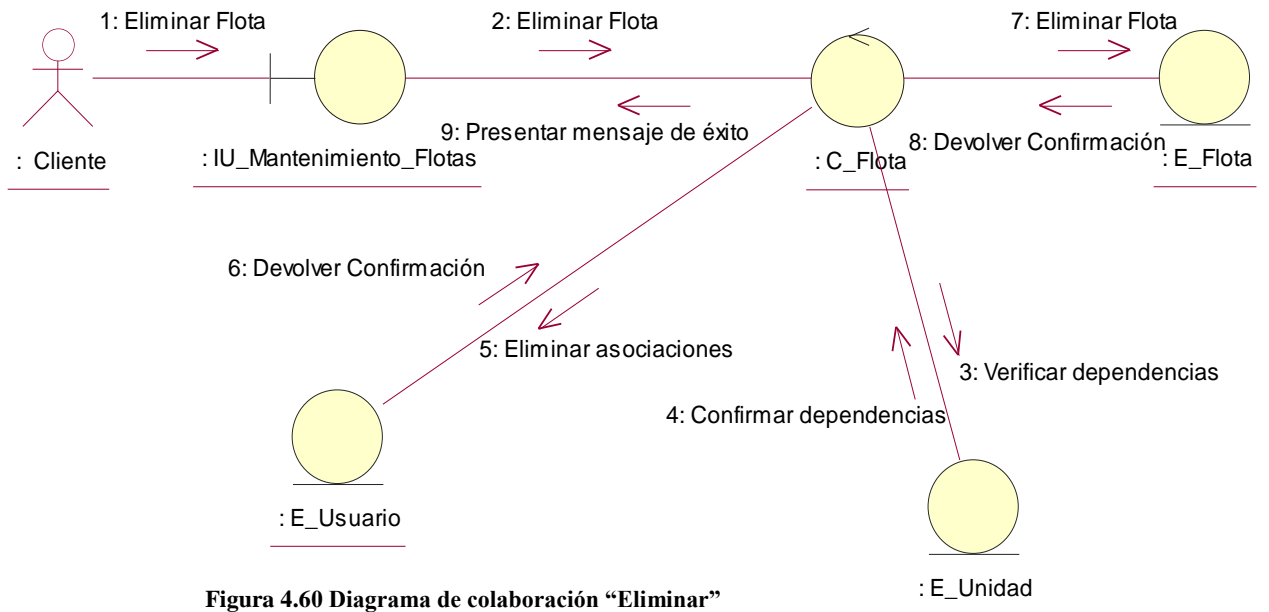


Figura 4.60 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario `IU_Mantenimiento_Flotas` selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora `C_Flota` que verifica que la flota no tenga unidades asociadas consultándolo a la clase entidad `E_Unidad`, la clase `E_Unidad` devuelve las unidades asociadas a la flota, en caso de que no hayan registros asociados solicita a la clase entidad `E_Usuario` eliminar todas las asociaciones con la flota a eliminar, la clase `E_Usuario` confirma la eliminación a la clase controladora, luego la clase controladora solicita a la entidad `E_Flota` eliminar la flota seleccionada, la clase entidad `E_Flota` confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase `IU_Mantenimiento_Flotas`.

#### 4.2.4.5 CUA – Mantenimiento de Conductores

##### Diagrama de Clases

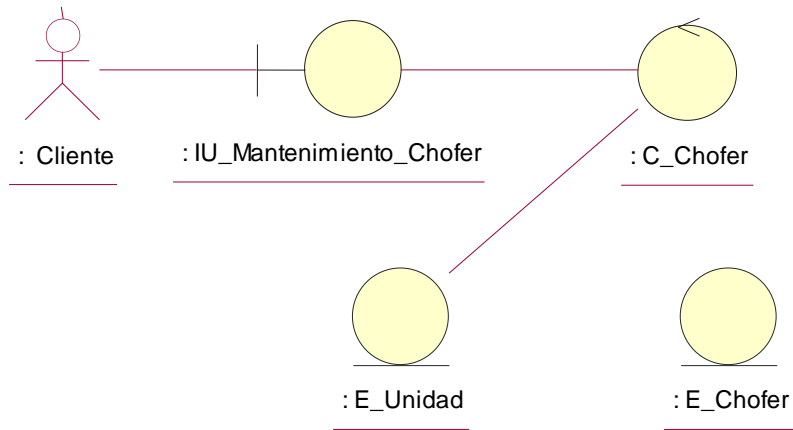


Figura 4.61 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Listar

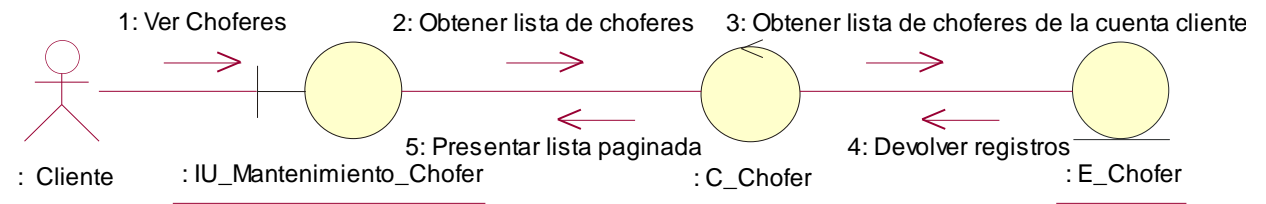
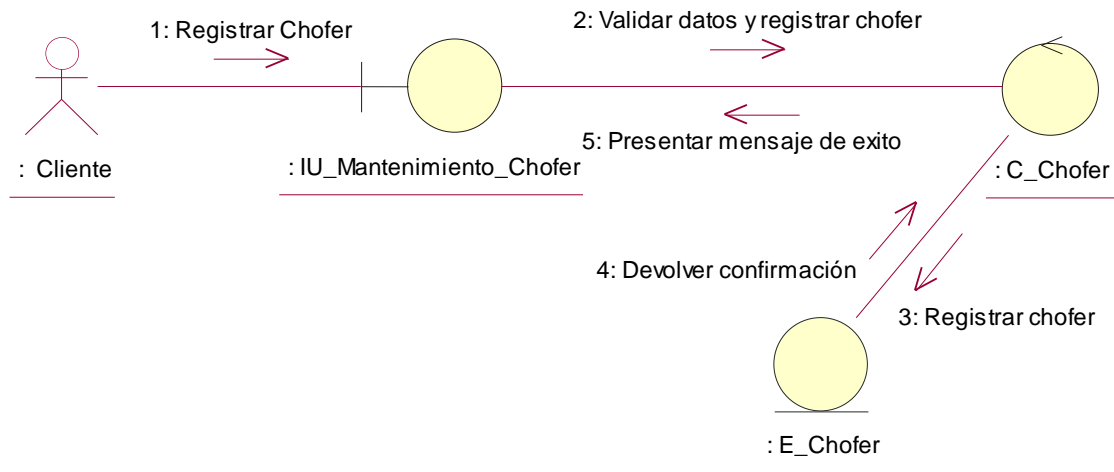


Figura 4.62 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Chofer solicita ver la lista de choferes de la cuenta cliente registrados en el sistema, la interfaz de usuario IU\_Mantenimiento\_Chofer solicita a la clase controladora C\_Chofer la lista de choferes de la cuenta cliente, la clase controladora se comunica con la clase entidad E\_Chofer quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de choferes a la clase IU\_Mantenimiento\_Chofer de forma paginada.

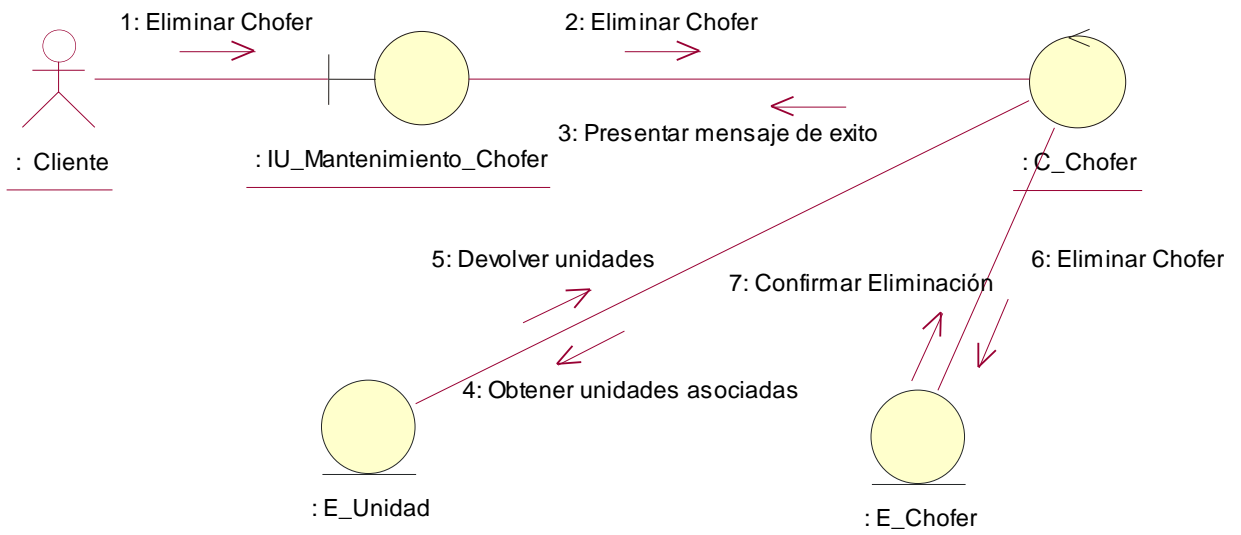
##### Diagrama de Colaboración – Registrar



**Figura 4.63 Diagrama de colaboración “Registrar”**

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Chofer ingresa los datos solicitados y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Chofer, esta verifica que los datos sean válidos, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Chofer y registra al chofer asociándolo a la cuenta cliente, la clase entidad E\_Chofer confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Chofer presentar un mensaje al usuario confirmando el resultado de la operación.

**Diagrama de Colaboración – Eliminar**



**Figura 4.64 Diagrama de colaboración “Eliminar”**

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Chofer selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_Chofer que verifica que el chofer no tenga unidades asociadas consultándolo a la clase entidad E\_Unidad, la clase E\_Unidad devuelve las unidades asociadas al chofer, en caso de que no hayan registros asociados solicita a la entidad E\_Chofer eliminar el chofer seleccionado, la clase entidad E\_Chofer confirma la eliminación a la clase controladora, quien solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Chofer.

#### 4.2.4.6 CUA – Mantenimiento de Geocercas

##### Diagrama de Clases

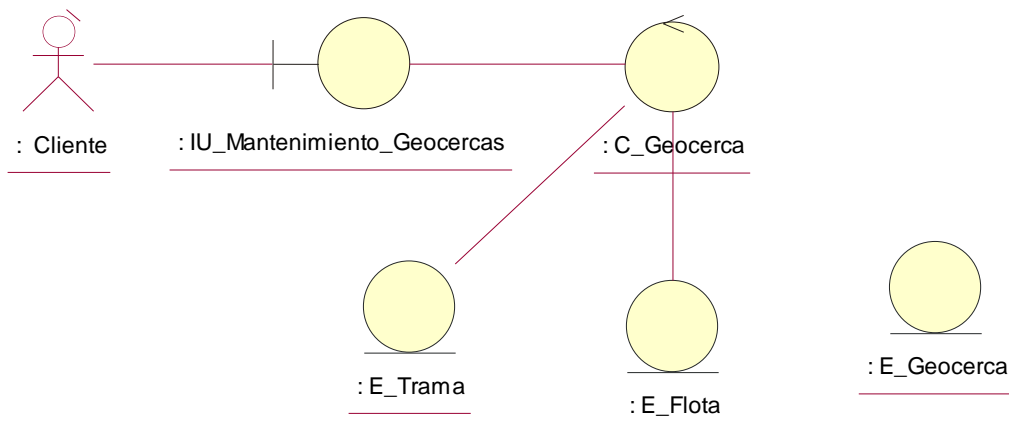


Figura 4.65 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Listar

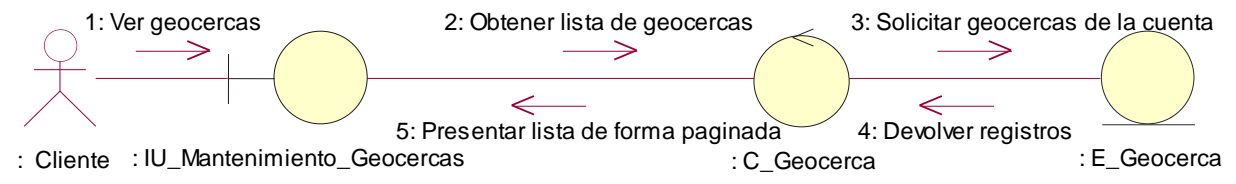


Figura 4.66 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Geocercas solicita ver la lista de geocercas de la cuenta cliente registradas en el sistema, la interfaz de usuario IU\_Mantenimiento\_Geocercas solicita a la clase controladora C\_Geocerca la lista de geocercas de la cuenta cliente, la clase controladora se comunica con la clase entidad E\_Geocerca que devuelve los registros a la clase controladora, la clase controladora devuelve la lista de geocercas a la clase IU\_Mantenimiento\_Geocercas de forma paginada.



## Diagrama de Colaboración – Registrar

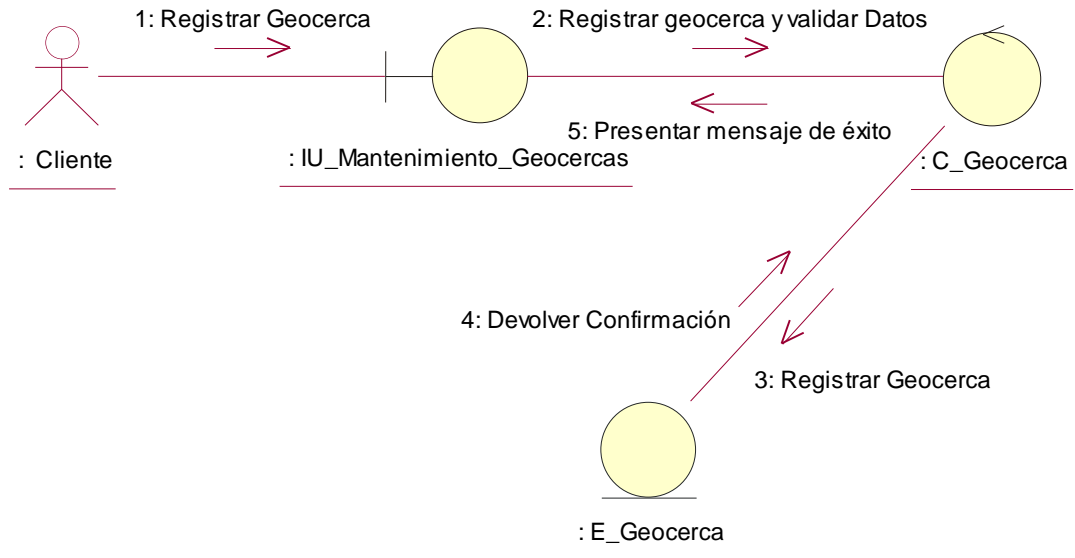


Figura 4.67 Diagrama de colaboración "Registrar"

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Geocercas ingresa los datos solicitados y selecciona la opción "Registrar", el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Geocerca, esta verifica que los datos sean válidos, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Geocerca y registra a la geocerca asociándola a la cuenta cliente, la clase entidad E\_Geocerca confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Geocercas presentar un mensaje al usuario confirmando el resultado de la operación.

## Diagrama de Colaboración – Eliminar

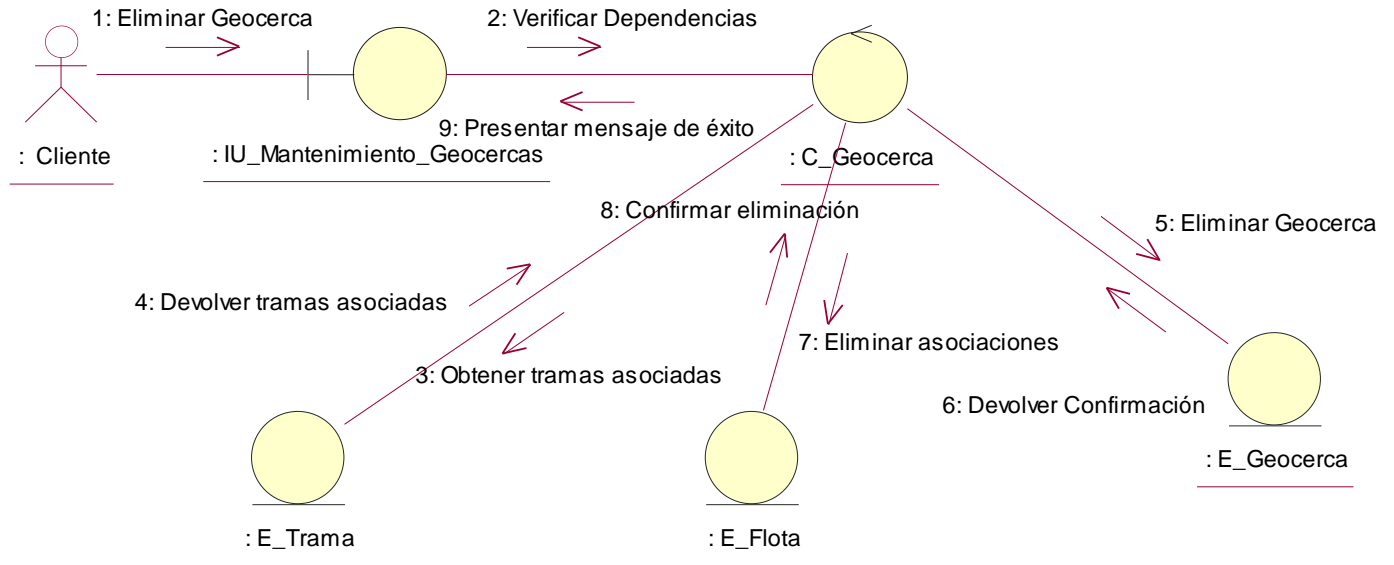


Figura 4.68 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Geocercas selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_Geocerca que verifica que la geocerca no tenga tramas asociadas consultándolo a la clase entidad E\_Trama, la clase E\_Trama devuelve las tramas asociadas a la geocerca, en caso de que no hayan registros asociados solicita a la entidad E\_Geocerca eliminar a la geocerca seleccionada, la clase entidad E\_Geocerca confirma la eliminación a la clase controladora, la clase controladora se comunica con la clase E\_Flota solicitándole eliminar las flotas asociadas, esta le confirma a la clase controladora la eliminación de las asociaciones, la clase controladora solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_Geocercas.

## Diagrama de Colaboración – Registrar Polígono

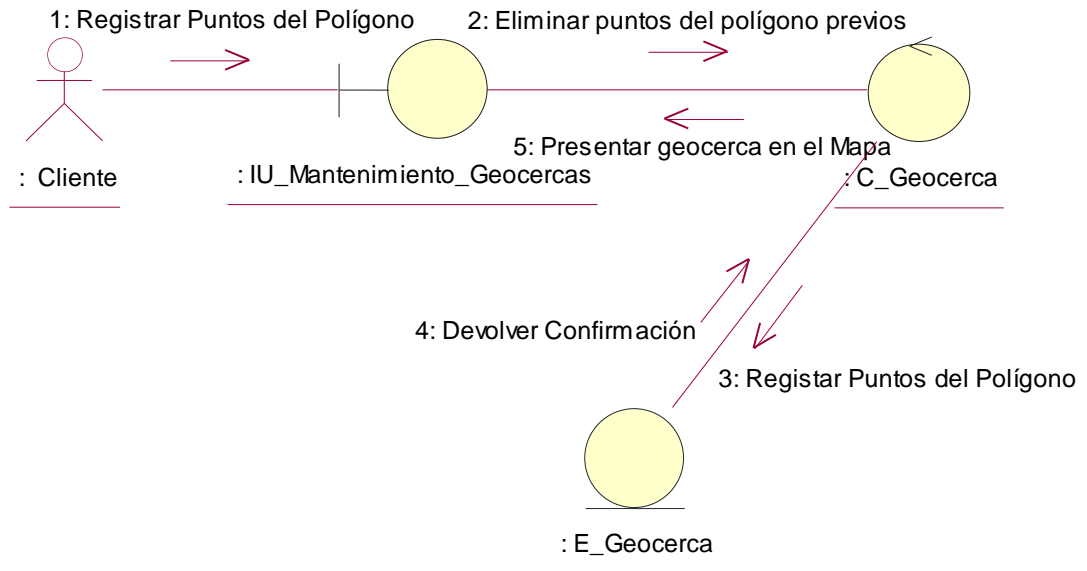


Figura 4.69 Diagrama de colaboración “Registrar Polígono”

**Descripción del diagrama de colaboración – Registrar Polígono:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Geocercas ingresa las coordenadas del polígono y selecciona la opción “Actualizar Coordenadas”, la clase interfaz de usuario invoca a clase controladora C\_Geocerca que elimina cualquier coordenada previa que haya tenido la geocerca solicitándolo a la clase entidad E\_Geocerca, una vez que las coordenadas hayan sido eliminadas, solicita nuevamente a la clase E\_Geocerca registrar las nuevas coordenadas del polígono asociándolo a la geocerca, la clase entidad E\_Geocerca confirma el registro a la clase controladora C\_Geocerca, quien solicita mostrar la confirmación de la operación a la clase interfaz de usuario IU\_Mantenimiento\_Geocercas.

### 4.2.4.7 CUA – Asociar Geocerca a Flotas

#### Diagrama de Clases

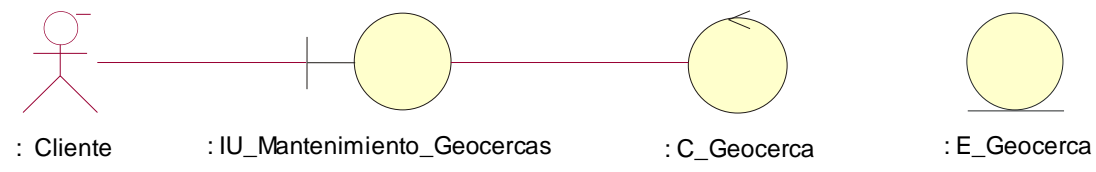


Figura 4.70 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Asociar Geocerca a Flotas

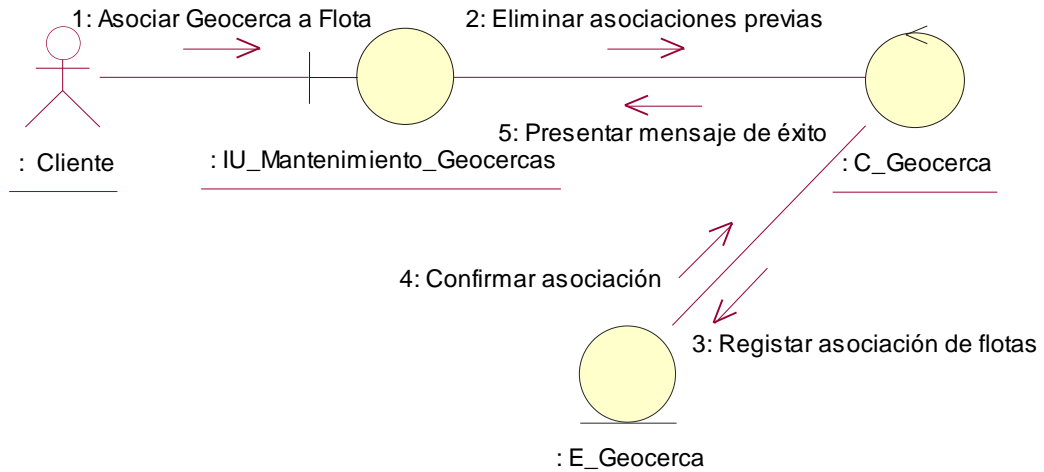


Figura 4.71 Diagrama de colaboración “Asociar Geocerca a Flotas”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Geocercas selecciona las flotas que serán afectadas por la geocerca y selecciona la opción “Registrar”, la clase interfaz de usuario solicita a la controladora C\_Geocerca eliminar cualquier asociación previa con la flota comunicando dicho requerimiento a la clase entidad E\_Geocerca, posteriormente la clase controladora nuevamente solicita a la entidad E\_Geocerca registrar la nueva asociación de flotas, esta clase confirma la asociación a la controladora, la clase controladora solicita a la interfaz de usuario UI\_Mantenimiento\_Geocercas mostrar un aviso de confirmación de la operación al usuario.

### 4.2.4.8 CUA – Mantenimiento de Puntos de Interés

#### Diagrama de Clases

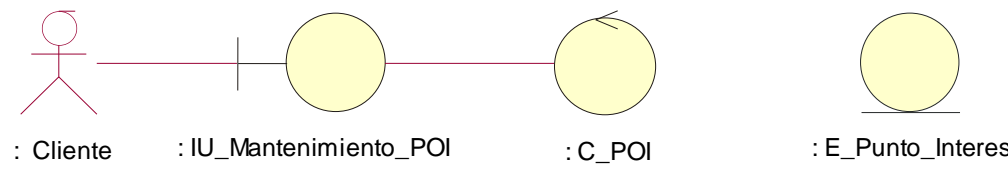


Figura 4.72 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Listar

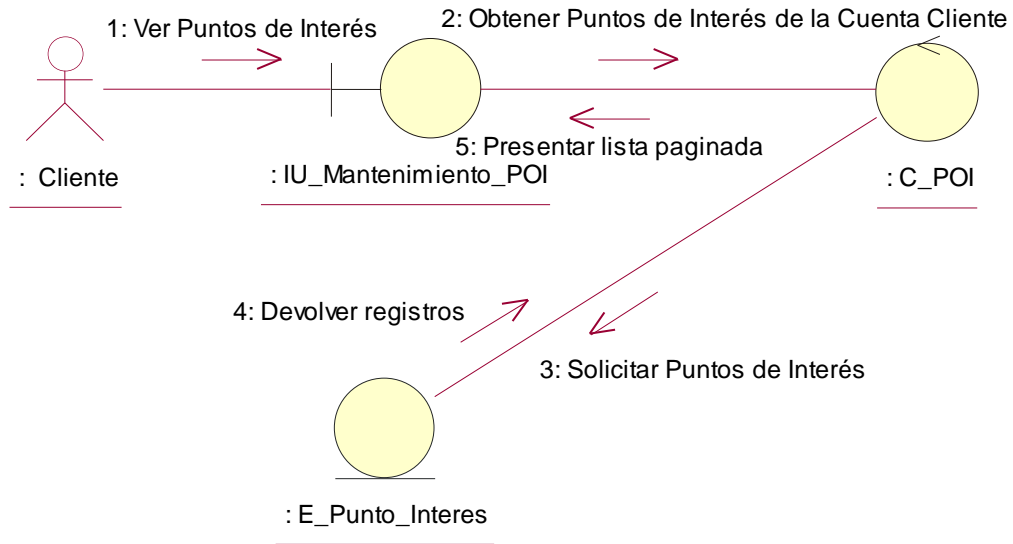


Figura 4.73 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_POI solicita ver la lista de puntos de interés de la cuenta cliente registrados en el sistema, esta se conecta a la clase controladora C\_POI, la clase controladora se comunica con la clase entidad E\_Punto\_Interes que devuelve los registros a la clase controladora y la clase controladora devuelve la lista de puntos de interés a la clase IU\_Mantenimiento\_POI de forma paginada.

### Diagrama de Colaboración – Registrar

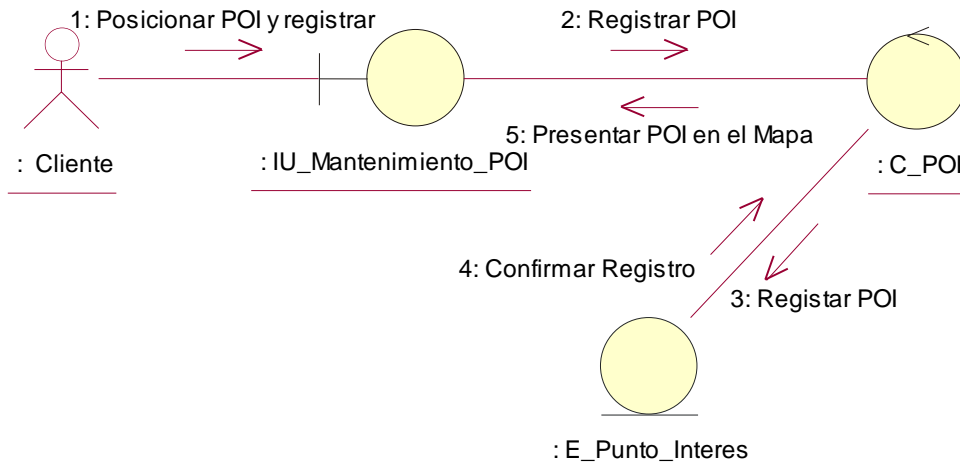
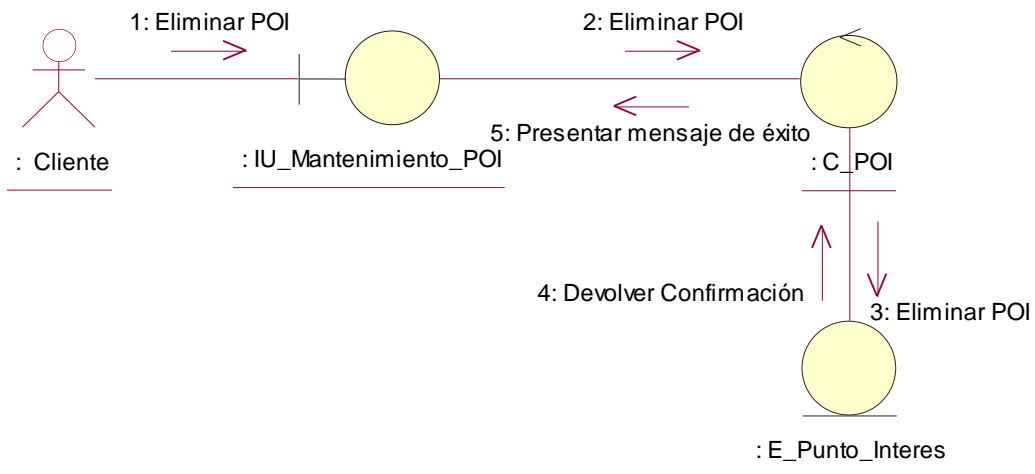


Figura 4.74 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_POI ingresa los datos solicitados del punto de interés, lo posiciona en el mapa y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_POI, esta invoca a la clase entidad E\_Punto\_Interés y registra a el punto de interés asociándolo a la cuenta cliente, la clase entidad E\_Punto\_Interés confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_POI presentar un mensaje al usuario confirmando el resultado de la operación.

**Diagrama de Colaboración – Eliminar**



**Figura 4.75 Diagrama de colaboración “Eliminar”**

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_POI selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_POI a fin de eliminar el punto de interés seleccionado, la clase controladora solicita a la clase entidad E\_Punto\_Interés eliminar el punto de interés, esta confirma a la clase controladora la eliminación del punto de interés, la clase controladora solicita presentar la confirmación al usuario a través de un mensaje a la clase IU\_Mantenimiento\_POI.

## 4.2.5 Realización de casos de uso de análisis – Paquete Reportes

### 4.2.5.1 CUA – Generar Reporte de Recorrido

#### Diagrama de Clases

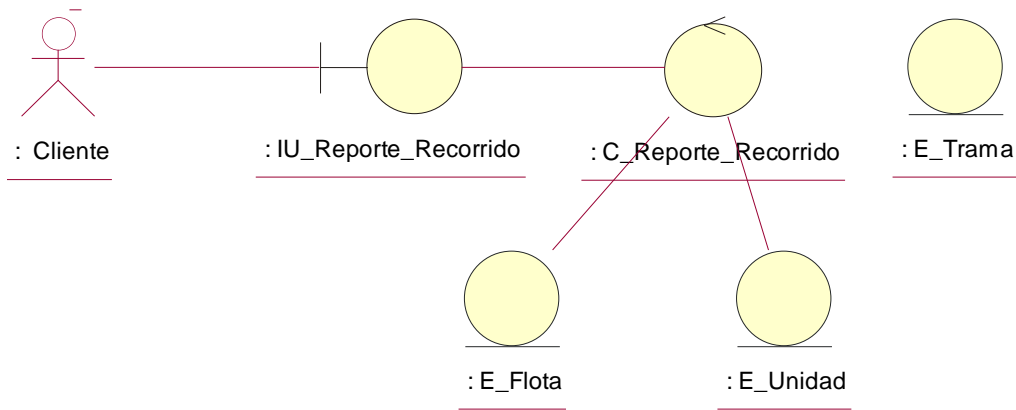


Figura 4.76 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Generar Formulario

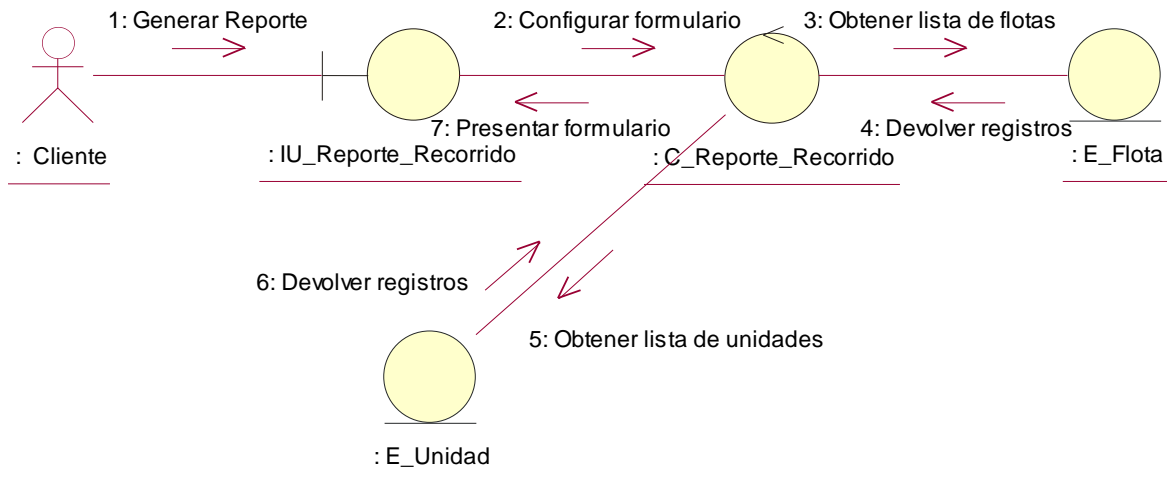


Figura 4.77 Diagrama de colaboración “Generar Formulario”

**Descripción del diagrama de colaboración – Generar Formulario:** El usuario a través del menú de reportes, selecciona la opción “Recorrido”, el sistema a través de la interfaz de usuario IU\_Reporte\_Recorrido solicita a la clase controladora C\_Reporte\_Recorrido configurar el formulario, para lo cual solicita a la clase entidad E\_Flota la lista de flotas asociadas a la cuenta cliente del usuario, una vez que la lista de flotas es entregada, la clase controladora solicita a la clase entidad E\_Unidad la lista de unidades asociadas a las flotas, cuando la clase controladora tiene la lista de unidades, la clase controladora solicita a la interfaz de usuario IU\_Reporte\_Recorrido presentar el formulario al usuario.

## Diagrama de Colaboración – Generar Reporte

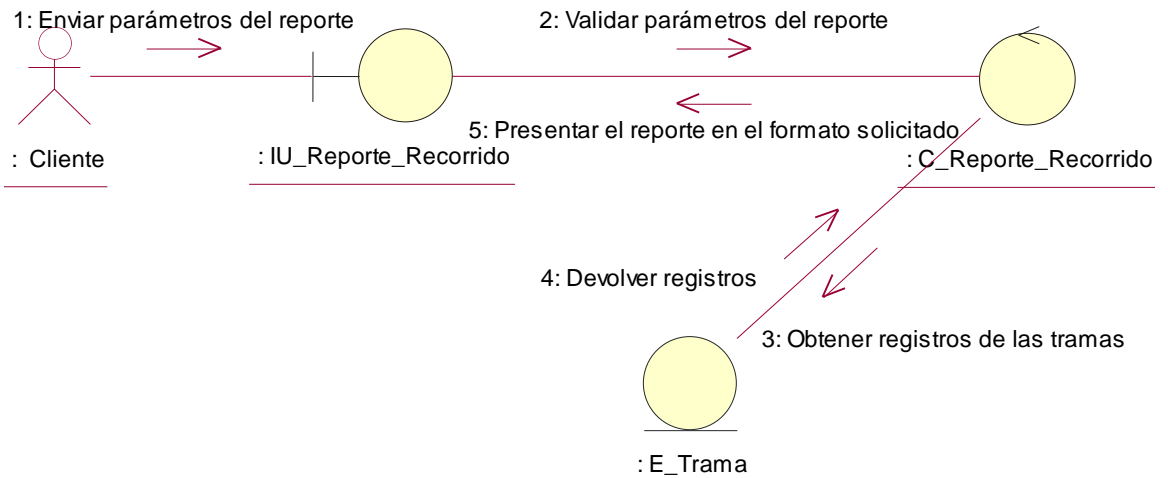


Figura 4.78 Diagrama de colaboración “Generar Reporte”

**Descripción del diagrama de colaboración – Generar Reporte:** El usuario a través de la interfaz de usuario IU\_Reporte\_Recorrido selecciona la opción “Siguiete”, la interfaz de usuario solicita a la clase controladora C\_Reporte\_Recorrido validar los parámetros del reporte, la clase controladora una vez que haya validado la consistencia de los datos, solicita a la clase entidad E\_Trama obtener los registros de las tramas relacionadas a las unidades, la clase entidad devuelve los registros, con dichos registros la clase controladora solicita a la interfaz de usuario presentar los registros de forma paginada y con las opciones de descargarlos en diversos formatos.

### 4.2.5.2 CUA – Generar Reporte de Alertas

#### Diagrama de Clases

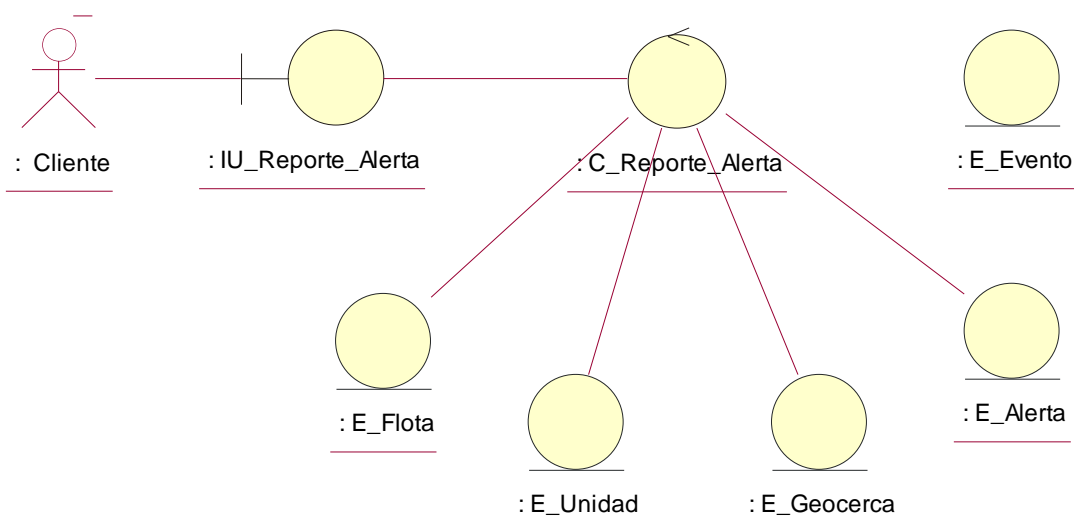


Figura 4.79 Diagrama de clases del caso de uso



## Diagrama de Colaboración – Generar Formulario

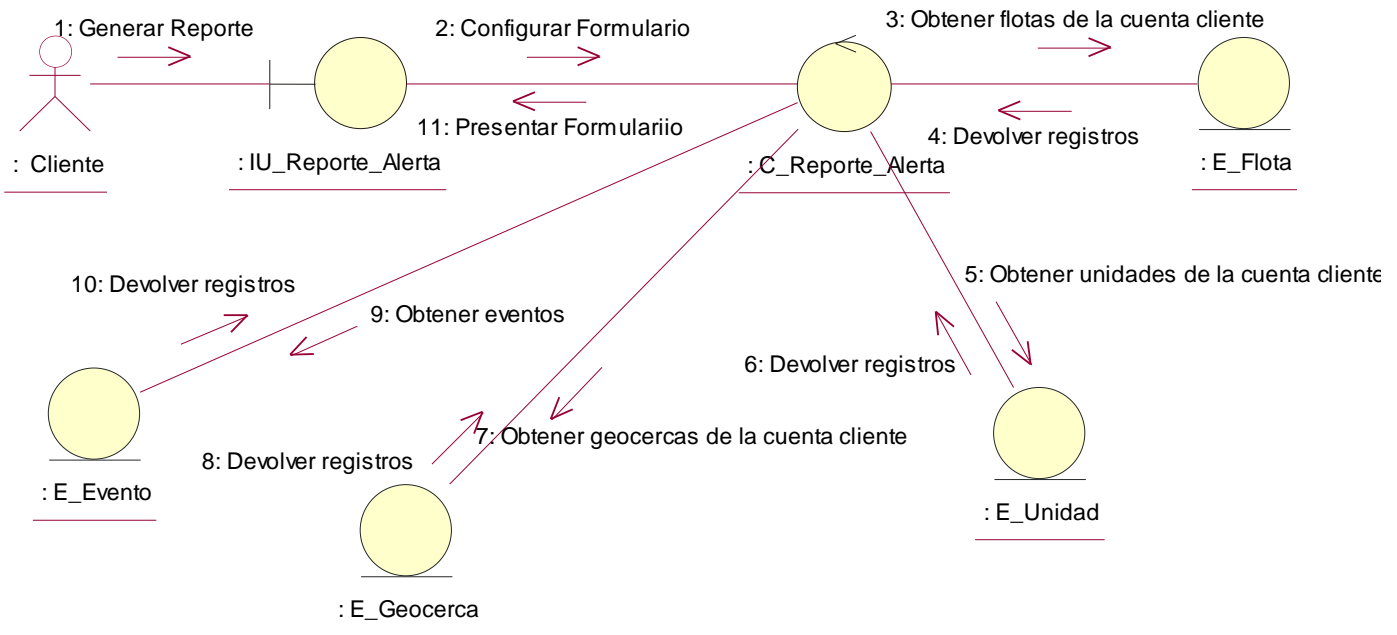


Figura 4.80 Diagrama de colaboración “Generar Formulario”

**Descripción del diagrama de colaboración – Generar Formulario:** El usuario a través del menú de reportes, selecciona la opción “Alertas”, el sistema a través de la interfaz de usuario IU\_Reporte\_Alerta solicita a la clase controladora C\_Reporte\_Alerta configurar el formulario, para lo cual solicita a la clase entidad E\_Flota la lista de flotas asociadas a la cuenta cliente del usuario, una vez que la lista de flotas es entregada, la clase controladora solicita a la clase entidad E\_Unidad la lista de unidades asociadas a las flotas, cuando la clase controladora tiene la lista de unidades, así mismo se solicita a la entidad E\_Geocerca la lista de geocercas asociadas a la cuenta cliente, junto con la clase entidad E\_Evento solicitando los eventos de tipo alerta del sistema, una vez que la clase controladora tiene todos datos agrupados, la clase controladora solicita a la interfaz de usuario IU\_Reporte\_Alerta presentar el formulario al usuario.

## Diagrama de Colaboración – Generar Reporte

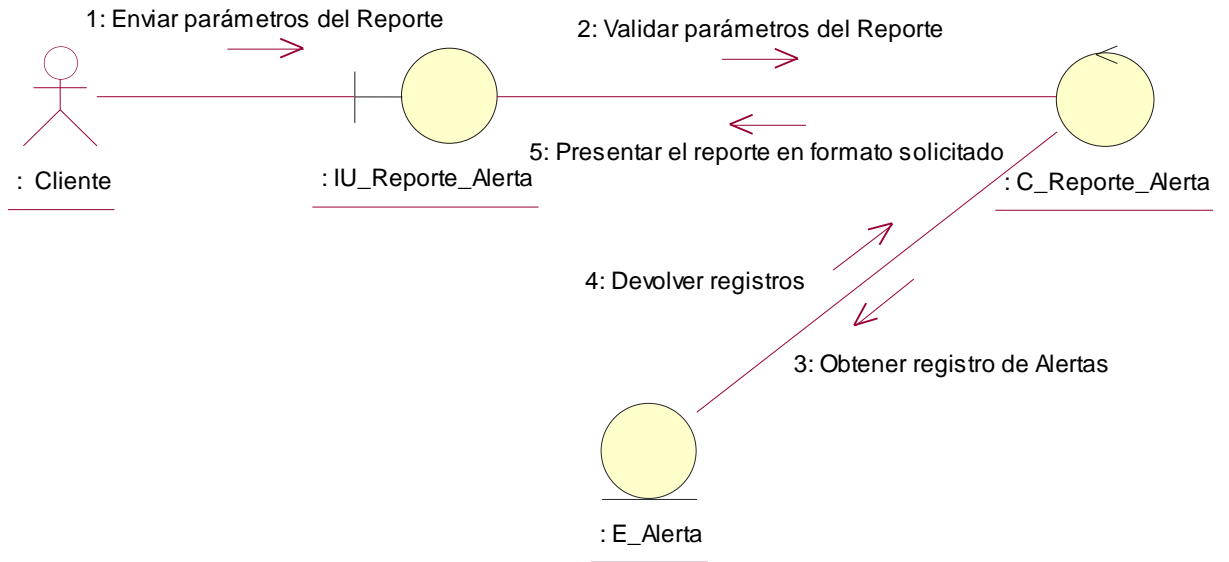


Figura 4.81 Diagrama de colaboración “Generar Reporte”

**Descripción del diagrama de colaboración – Generar Reporte:** El usuario a través de la interfaz de usuario IU\_Reporte\_Alerta selecciona la opción “Siguiete”, la interfaz de usuario solicita a la clase controladora C\_Reporte\_Alerta validar los parámetros del reporte, la clase controladora una vez que haya validado la consistencia de los datos, solicita a la clase entidad E\_Alerta obtener los registros de las alertas relacionadas a los parámetros del reporte, la clase entidad devuelve los registros, con dichos registros la clase controladora solicita a la interfaz de usuario presentar los registros de forma paginada y con las opciones de descargarlos en diversos formatos.

### 4.2.5.3 CUA – Generar Reporte de Comandos

#### Diagrama de Clases

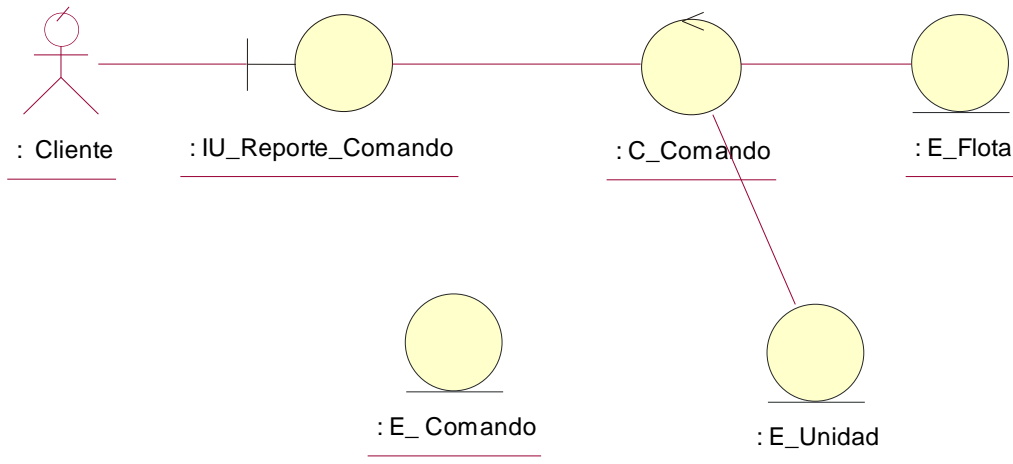


Figura 4.82 Diagrama de clases del caso de uso

#### Diagrama de Colaboración – Generar Formulario

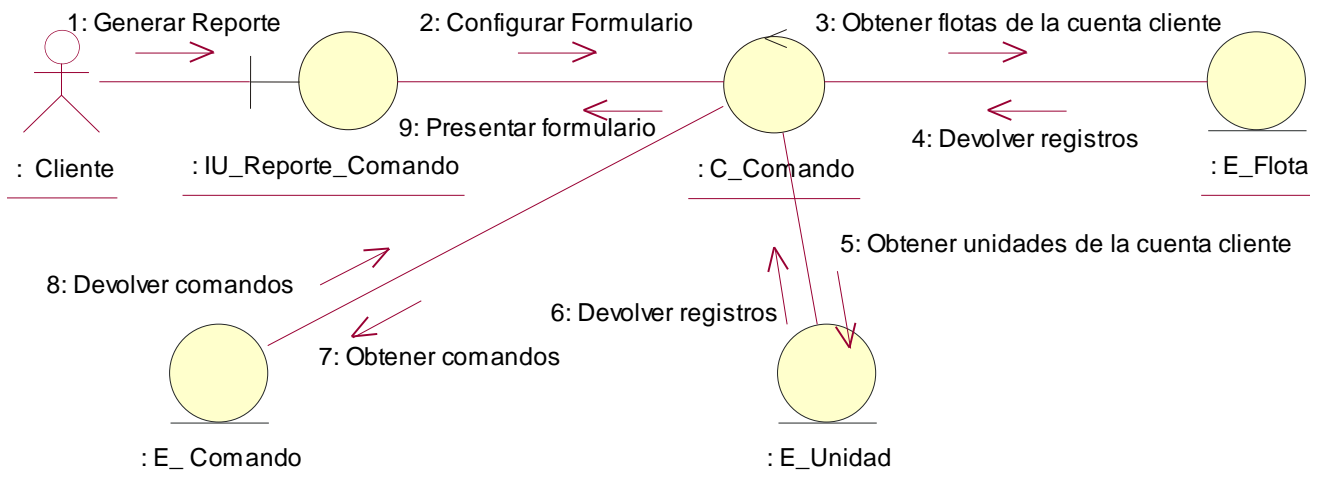


Figura 4.83 Diagrama de colaboración “Generar Formulario”

**Descripción del diagrama de colaboración – Generar Formulario:** El usuario a través del menú de reportes, selecciona la opción “Comandos”, el sistema a través de la interfaz de usuario IU\_Reporte\_Comando solicita a la clase controladora C\_Reporte\_Comando configurar el formulario, para lo cual solicita a la clase entidad E\_Flota la lista de flotas asociadas a la cuenta cliente del usuario, una vez que la lista de flotas es entregada, la clase controladora solicita a la clase entidad E\_Unidad la lista de unidades asociadas a las flotas, cuando la clase controladora

tiene la lista de unidades, así mismo se solicita a la entidad E\_Comando la lista de comandos del sistema, una vez que la clase controladora tiene todos datos agrupados, la clase controladora solicita a la interfaz de usuario IU\_Reporte\_Comando presentar el formulario al usuario.

### Diagrama de Colaboración – Generar Reporte

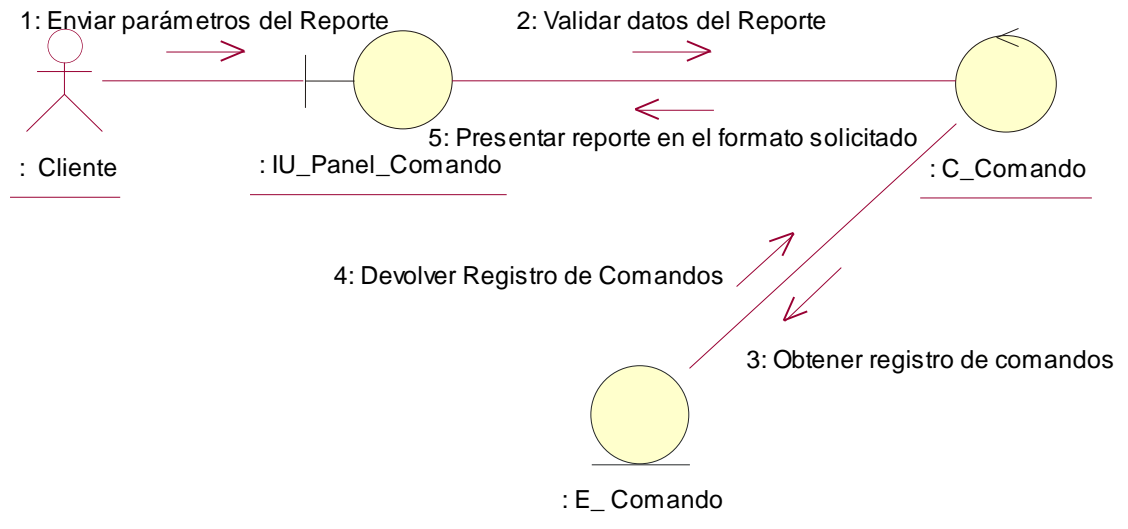


Figura 4.84 Diagrama de colaboración “Generar Reporte”

**Descripción del diagrama de colaboración – Generar Reporte:** El usuario a través de la interfaz de usuario IU\_Reporte\_Comando selecciona la opción “Siguiete”, la interfaz de usuario solicita a la clase controladora C\_Reporte\_Comando validar los parámetros del reporte, la clase controladora una vez que haya validado la consistencia de los datos, solicita a la clase entidad E\_Comando obtener los registros de los comandos relacionadas a los parámetros del reporte, la clase entidad devuelve los registros, con dichos registros la clase controladora solicita a la interfaz de usuario presentar los registros de forma paginada y con las opciones de descargarlos en diversos formatos.

## 4.2.6 Realización de casos de uso de análisis – Paquete Seguridad

### 4.2.6.1 CUA – Actualizar cuenta

#### Diagrama de Clases

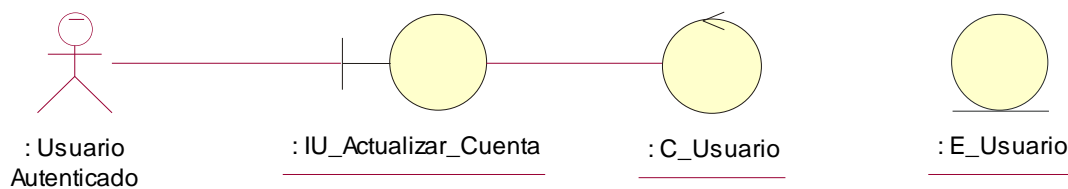


Figura 4.85 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Actualizar datos de ingreso al sistema

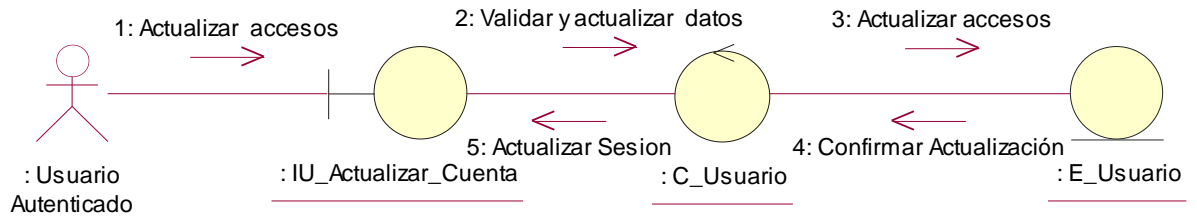


Figura 4.86 Diagrama de colaboración “Actualizar datos de ingreso al sistema”

**Descripción del diagrama de colaboración – Actualizar datos de ingreso al sistema:** El usuario a través del menú de seguridad, selecciona la opción “Actualizar datos de ingreso”, el sistema a través de la interfaz de usuario IU\_Actualizar\_Cuenta presenta el formulario de datos de ingreso al sistema, el usuario llena los datos solicitados y selecciona la opción actualizar, la interfaz de usuario solicita a la controladora C\_Usuario que valide los datos de ingreso, como contraseña y su confirmación o el correo electrónico nuevo, la controladora valida los datos y solicita a la clase entidad E\_Usuario que actualice la contraseña o el correo electrónico del usuario que sirve como nombre de usuario, la clase entidad confirma la actualización de los datos y la clase controladora solicita a la interfaz IU\_Actualizar\_Cuenta actualizar la sesión del usuario a fin de fijar los nuevos parámetros de acceso.

### Diagrama de Colaboración – Actualizar datos personales

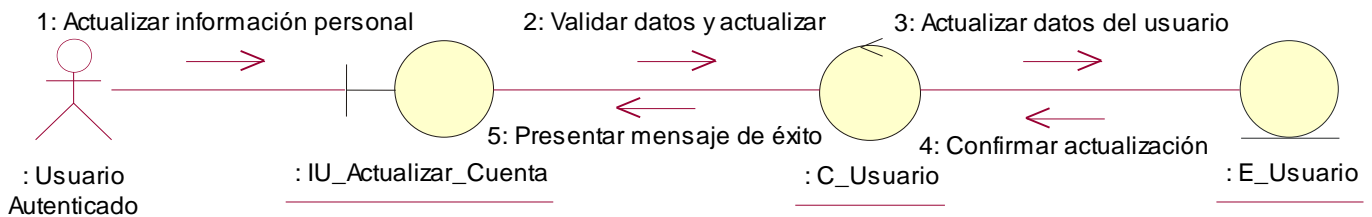


Figura 4.87 Diagrama de colaboración “Actualizar datos personales”

**Descripción del diagrama de colaboración – Actualizar datos personales:** El usuario a través del menú de seguridad, selecciona la opción “Actualizar datos personales”, el sistema a través de la interfaz de usuario IU\_Actualizar\_Cuenta presenta el formulario de datos personales del usuario, el usuario llena los datos solicitados y selecciona la opción actualizar, la interfaz de usuario solicita a la controladora C\_Usuario que valide los datos personales del usuario y solicita a la clase entidad E\_Usuario que actualice los datos personales del usuario, la clase

entidad confirma la actualización de los datos y la clase controladora solicita a la interfaz IU\_Actualizar\_Cuenta presentar un mensaje de confirmación de la operación al usuario.

### Diagrama de Colaboración – Actualizar configuración del Mapa

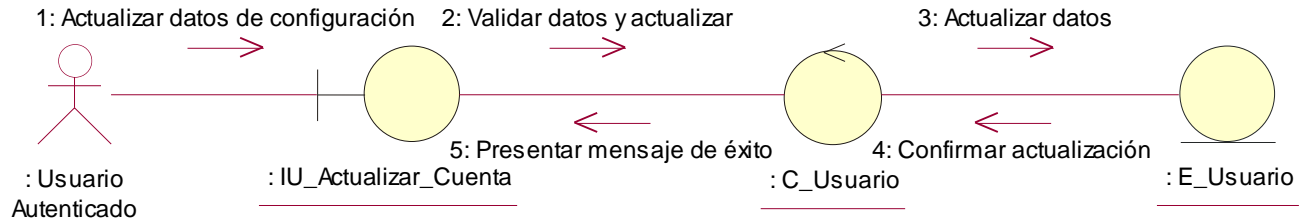


Figura 4.88 Diagrama de colaboración “Actualizar configuración del Mapa”

**Descripción del diagrama de colaboración – Actualizar configuración del Mapa:** El usuario a través del menú de seguridad, selecciona la opción “Actualizar configuración del Mapa”, el sistema a través de la interfaz de usuario IU\_Actualizar\_Cuenta presenta el formulario de configuración, el usuario llena los datos solicitados y selecciona la opción actualizar, la interfaz de usuario solicita a la controladora C\_Usuario que valide los datos de configuración y solicita a la clase entidad E\_Usuario que actualice los datos de configuración del usuario, la clase entidad confirma la actualización de los datos y la clase controladora solicita a la interfaz IU\_Actualizar\_Cuenta presentar un mensaje de confirmación de la operación al usuario.

#### 4.2.6.2 CUA – Ingresar al sistema

##### Diagrama de Clases

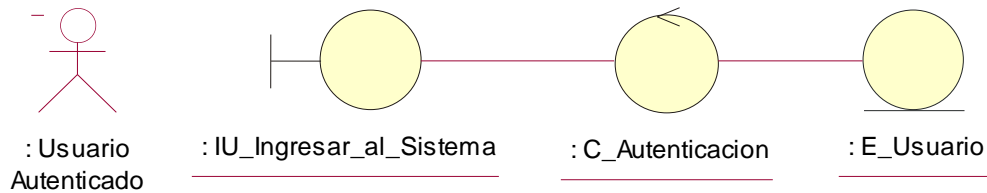


Figura 4.89 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Ingresar al Sistema

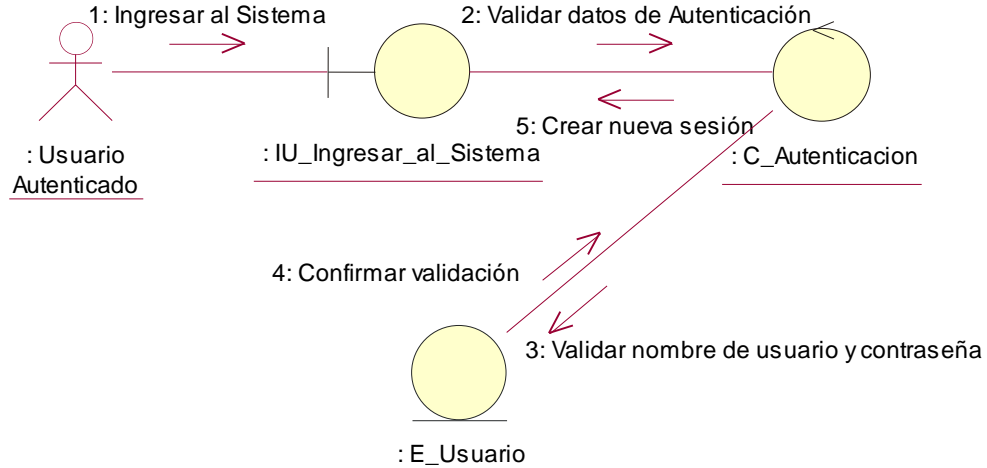


Figura 4.90 Diagrama de colaboración “Ingresar al Sistema”

**Descripción del diagrama de colaboración – Ingresar al Sistema:** El usuario a través de la interfaz de usuario IU\_Ingresar\_al\_Sistema ingresa su correo electrónico, su contraseña y selecciona la opción “Ingresar al Sistema”, la interfaz de usuario solicita a la clase controladora C\_Autenticación que valide los datos de ingreso, a fin de verificar que la contraseña corresponda al correo electrónico ingresado, la clase controladora solicita a la entidad E\_Usuario que valide si existe un usuario en el sistema que posea dicho correo electrónico, si existe el usuario con dicho correo electrónico verifica que su contraseña es igual a la contraseña ingresada, si el usuario con dichas credenciales existe, la clase controladora solicita a la interfaz de usuario crear una nueva sesión con dichas credenciales, en caso de que las credenciales no coincidan, la clase controladora solicita a la interfaz de usuario IU\_Ingresar\_al\_Sistema presentar un mensaje de error al usuario.

### 4.2.6.3 CUA – Recuperar contraseña

#### Diagrama de Clases

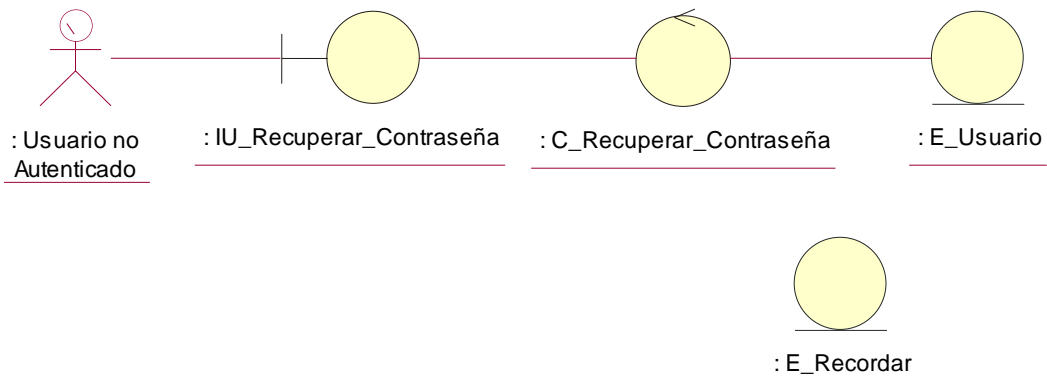


Figura 4.91 Diagrama de clases del caso de uso

## Diagrama de Colaboración – Enviar Instrucciones

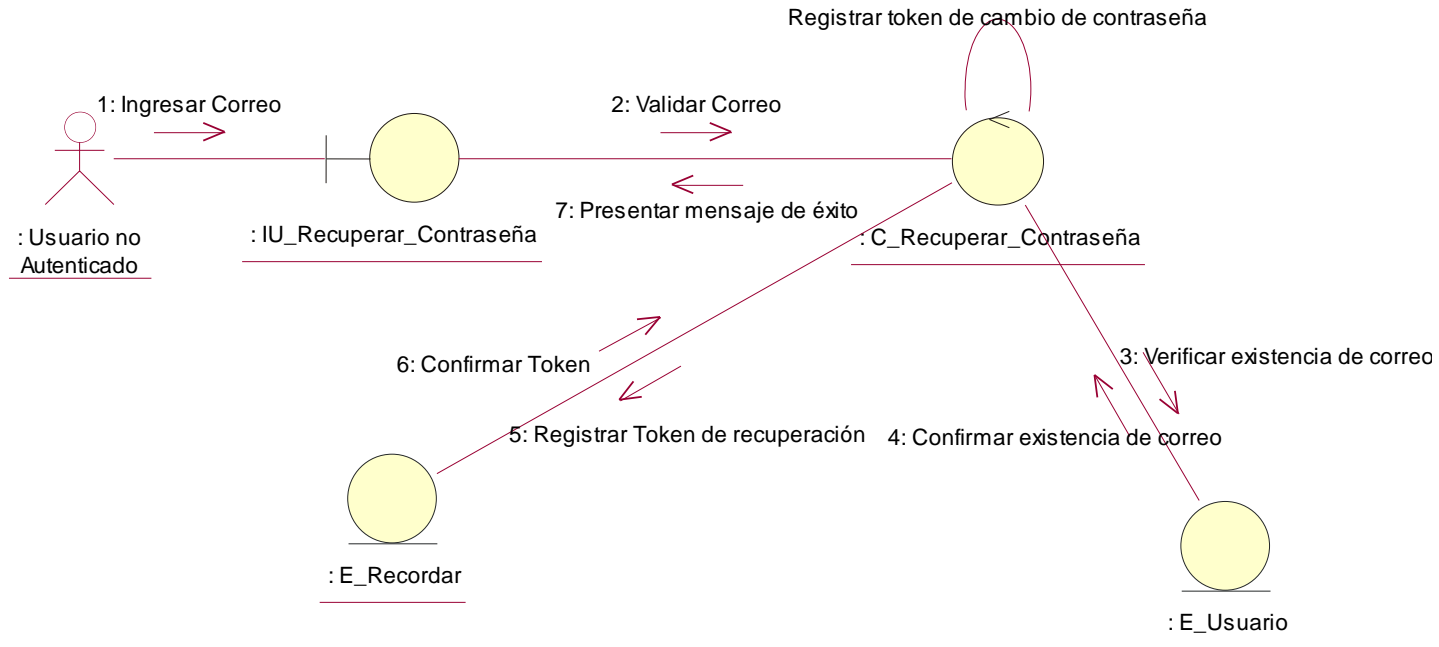


Figura 4.92 Diagrama de colaboración “Enviar Instrucciones”

**Descripción del diagrama de colaboración – Enviar Instrucciones:** El usuario a través de la interfaz de usuario `IU_Recuperar_Contraseña` ingresa su correo electrónico y selecciona la opción “Enviar Instrucciones”, la clase interfaz de usuario solicita a la clase controladora `C_Recuperar_Contraseña` validar el correo electrónico ingresado, para eso la clase controladora solicita a la clase entidad `E_Usuario` si existe algún usuario en el sistema que posea dicho correo electrónico como nombre de usuario, en caso de no exista la clase controladora solicita la interfaz de usuario que presente un mensaje al usuario indicando que el correo electrónico no está registrado, en caso de existir un usuario con dicho correo electrónico, la clase controladora `C_Recuperar_Contraseña` se comunica con la clase entidad `E_Recordar` a fin de registrar un token de recuperación por la solicitud de recuperación de contraseña, la clase entidad confirma a la clase controladora el registro del token, la clase controladora envía un correo electrónico a la cuenta de correo electrónico del usuario junto con un enlace de validación que contiene el token y presenta un mensaje indicando que las instrucciones fueron enviadas a la cuenta de correo electrónico del usuario.



## Diagrama de Colaboración – Restaurar Contraseña

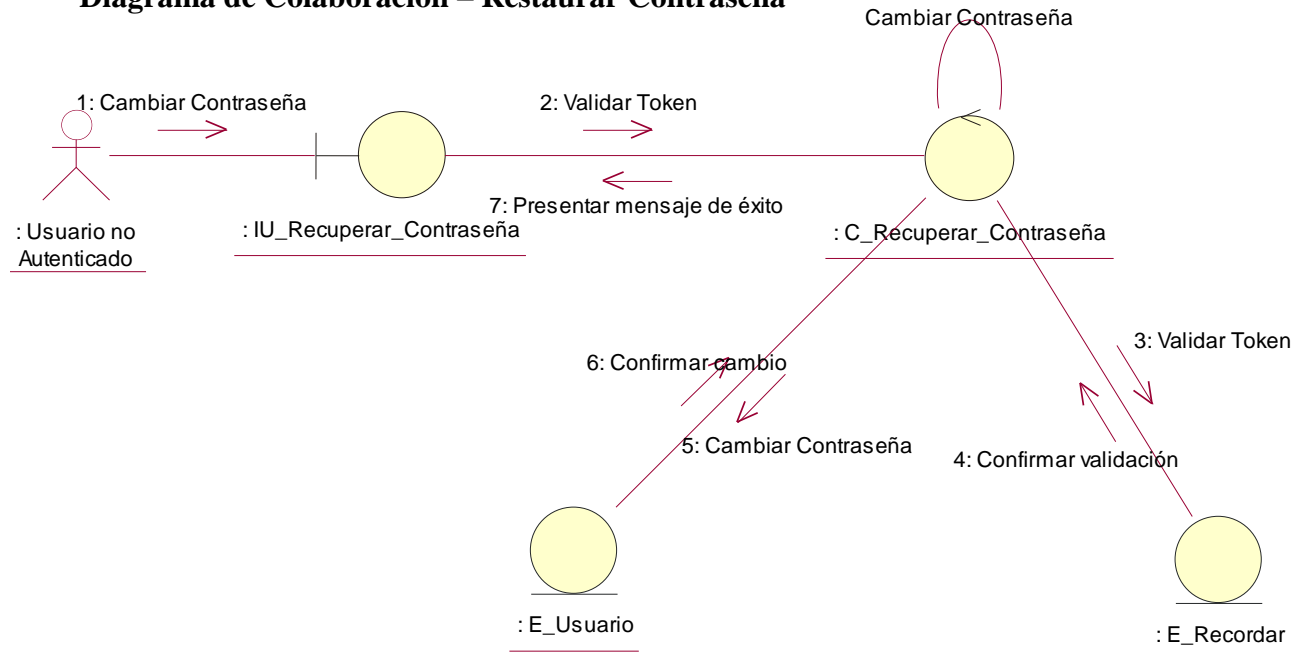


Figura 4.93 Diagrama de colaboración “Restaurar Contraseña”

**Descripción del diagrama de colaboración – Restaurar Contraseña:** El usuario dentro del detalle de su correo electrónico hace clic en el enlace de recuperación de contraseña que contiene el token de recuperación, el enlace le presenta la interfaz de usuario `IU_Recuperar_Contraseña` que solicita a clase controladora `C_Recuperar_Contraseña` validar el token, la clase controladora solicita a la entidad `E_Recordar` si existe algún registro con dicho token, de ser así solicita cambiar su estado de generado a utilizado, posteriormente la clase controladora solicita a la clase entidad `E_Usuario` actualizar la contraseña del usuario con la contraseña ingresa, una vez que la clase `E_Usuario` confirma la actualización de la contraseña, la clase controladora solicita a la interfaz de usuario presentar un mensaje al usuario confirmando la operación

#### 4.2.6.4 CUA – Salir del sistema

##### Diagrama de Clases

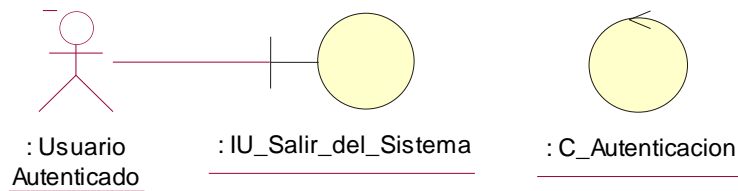


Figura 4.94 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Salir del Sistema

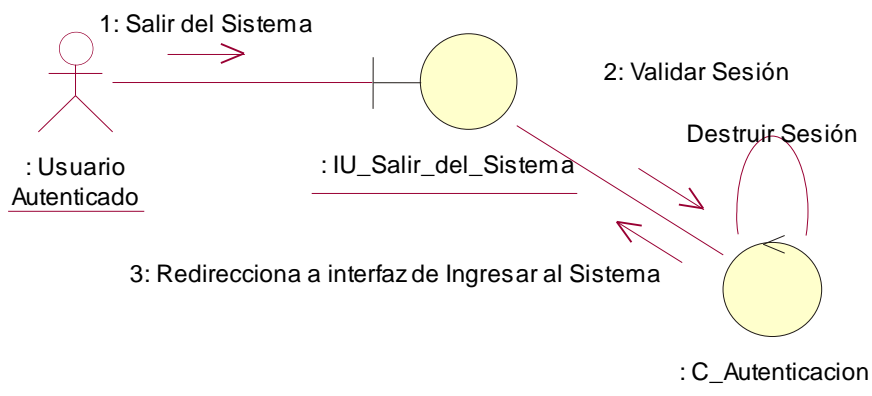


Figura 4.95 Diagrama de colaboración “Salir del Sistema”

**Descripción del diagrama de colaboración – Salir del Sistema:** El usuario a través de la interfaz de usuario IU\_Salir\_del\_Sistema, selecciona la opción “Salir del Sistema”, la interfaz solicita a la clase controladora C\_Autenticacion validar si la sesión se encuentra activa, de ser así se solicita destruir la sesión del usuario, cuando la clase controladora destruye la sesión, se le solicita a la interfaz de usuario IU\_Salir\_del\_Sistema direccionar a la interfaz de Ingresar al Sistema.

#### 4.2.6.5 CUA – Mantenimiento de Usuarios

##### Diagrama de Clases



Figura 4.96 Diagrama de clases del caso de uso

### Diagrama de Colaboración – Listar

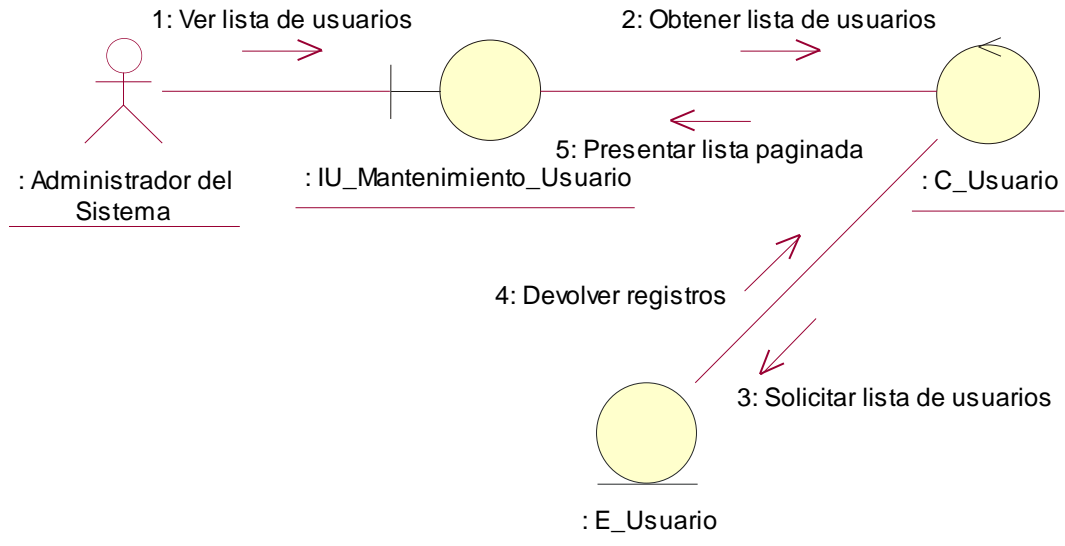


Figura 4.97 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Usuario solicita ver la lista de usuarios registrados en el sistema, la interfaz de usuario solicita a la clase controladora C\_Usuario la lista de usuarios, la clase controladora se comunica con la clase entidad E\_Usuario quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de usuarios a la clase IU\_Mantenimiento\_Usuario de forma paginada.

## Diagrama de Colaboración – Registrar

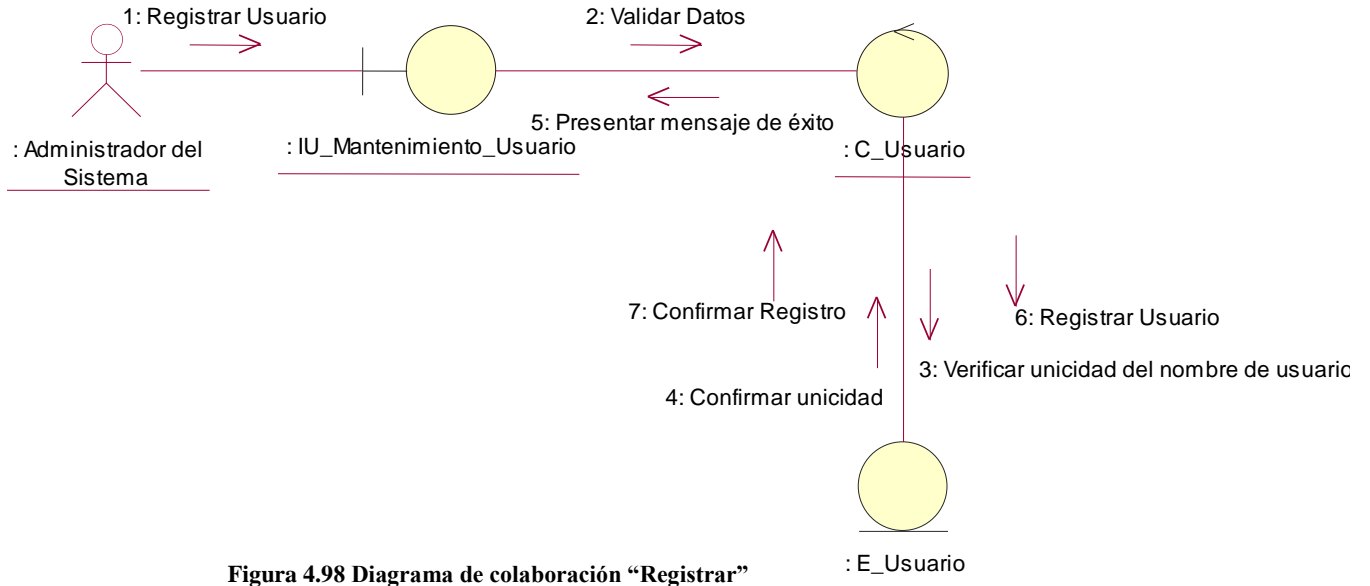


Figura 4.98 Diagrama de colaboración "Registrar"

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Usuario ingresa los datos solicitados y selecciona la opción "Registrar", el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Usuario, esta verifica que los datos sean válidos y que el nombre de usuario sea único consultándolo con la clase entidad E\_Usuario, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Usuario nuevamente y registra al usuario, la clase entidad E\_Usuario confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Usuario presentar un mensaje confirmando el resultado de la operación.

## Diagrama de Colaboración – Eliminar

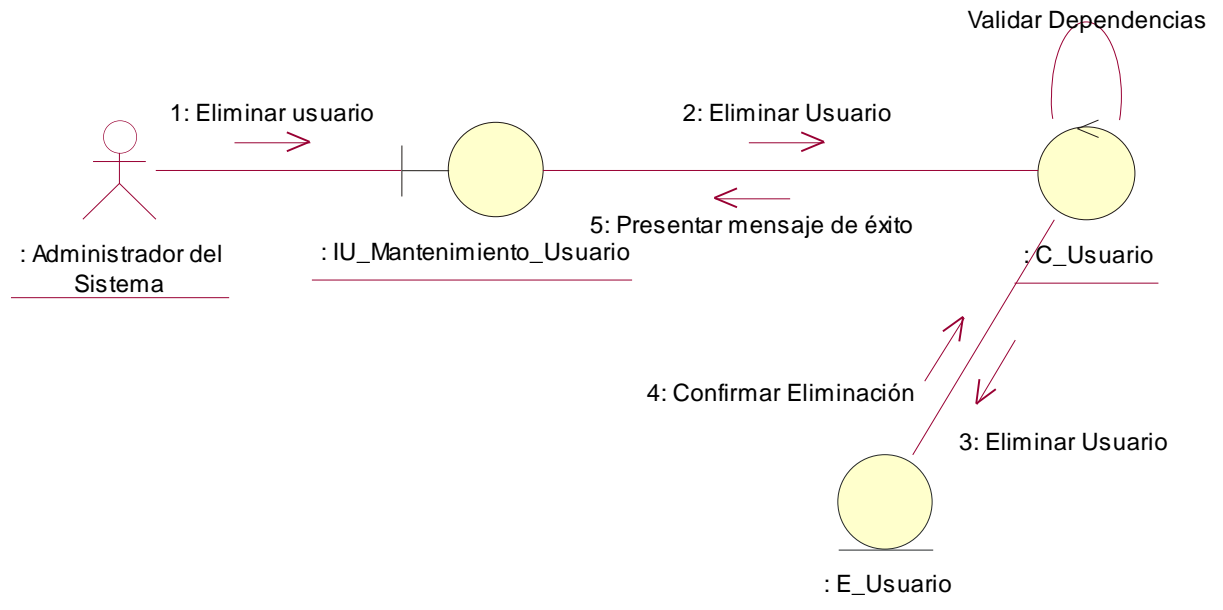


Figura 4.99 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Usuario selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_Usuario que verifica que el usuario no tenga registros asociados, en caso de que no hayan registros asociados, la clase controladora solicita a la entidad E\_Usuario eliminar el usuario seleccionado, la clase entidad E\_Usuario confirma la eliminación a la clase controladora, quien solicita presentar la confirmación a través de un mensaje a la clase IU\_Mantenimiento\_Usuario.

## Diagrama de Colaboración – Asociar a Flotas

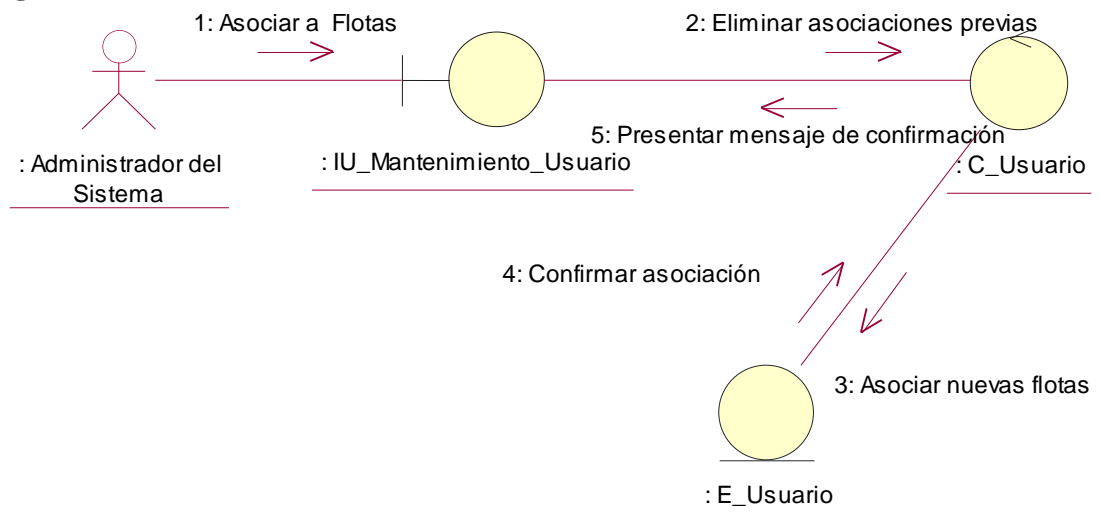


Figura 4.100 Diagrama de colaboración “Asociar a Flotas”

**Descripción del diagrama de colaboración – Asociar a Flotas:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Usuario selecciona las flotas que estarán asociadas al usuario a fin de poder consultar unidades en el mapa y selecciona la opción “Registrar”, la interfaz de usuario solicita a la clase controladora C\_Usuario realizar la asociación, para esto la clase controladora elimina toda relación previa del usuario con alguna flota del sistema solicitándole dicha operación a la clase entidad E\_Usuario, posteriormente, la clase controladora indica a la clase E\_Usuario registrar la nueva asociación, la clase entidad confirma la operación a la clase controladora, la clase controlada C\_Usuario solicita a la interfaz de usuario presentar un mensaje confirmando la operación.

#### 4.2.6.6 CUA – Mantenimiento de Roles

##### Diagrama de Clases

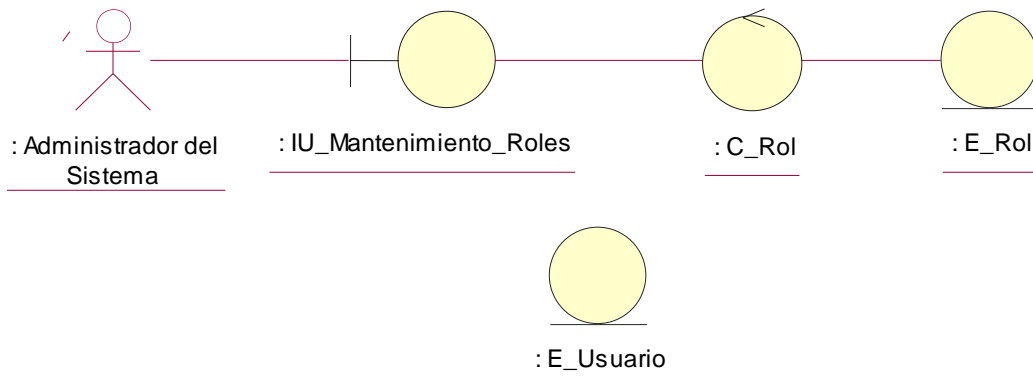


Figura 4.101 Diagrama de clases del caso de uso

##### Diagrama de Colaboración – Listar

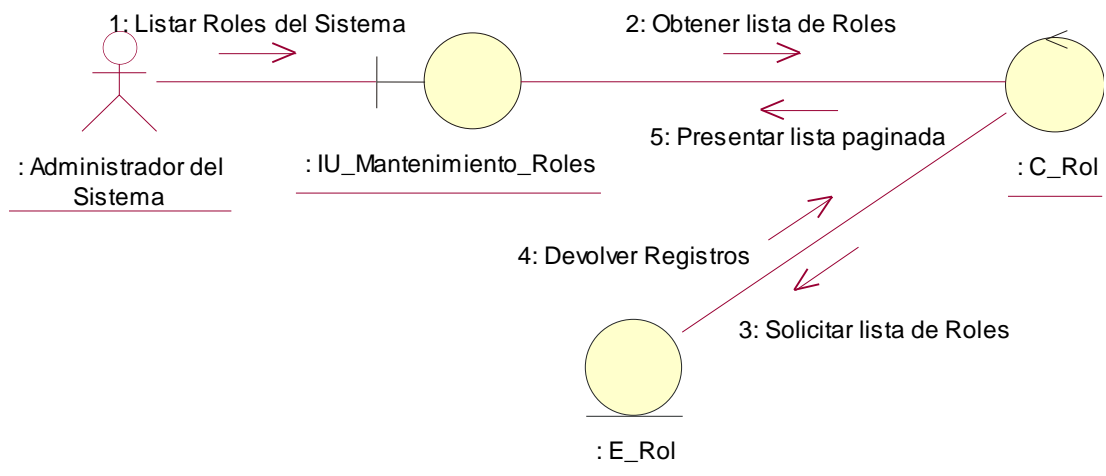


Figura 4.102 Diagrama de colaboración “Listar”

**Descripción del diagrama de colaboración – Listar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Roles solicita ver la lista de roles registrados en el sistema, la interfaz de usuario solicita a la clase controladora C\_Rol la lista de roles, la clase controladora se comunica con la clase entidad E\_Rol quien devuelve los registros a la clase controladora, la clase controladora devuelve la lista de roles a la clase IU\_Mantenimiento\_Roles de forma paginada.

**Diagrama de Colaboración – Registrar**

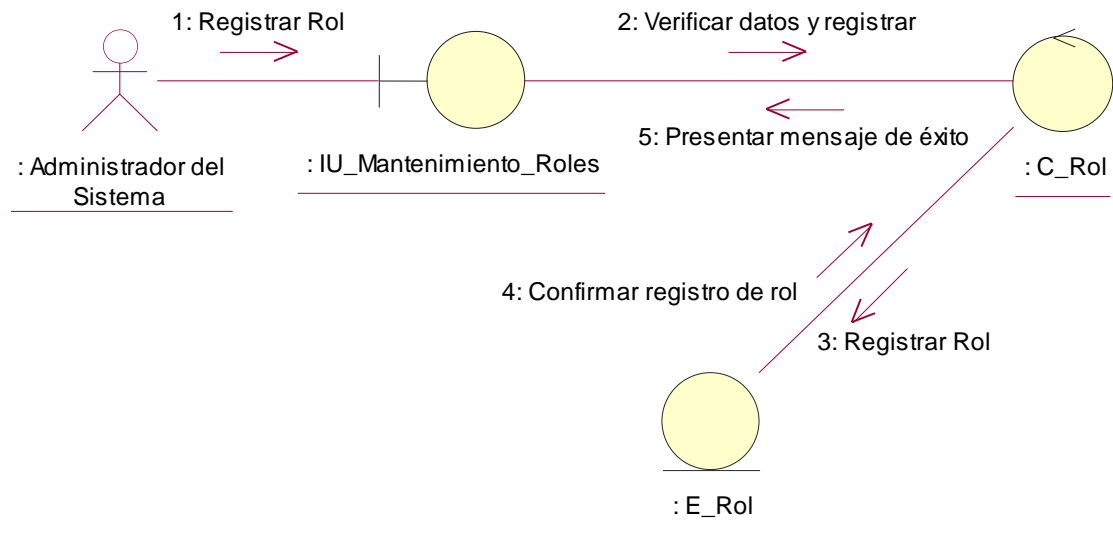


Figura 4.103 Diagrama de colaboración “Registrar”

**Descripción del diagrama de colaboración – Registrar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Roles ingresa el nombre del rol y selecciona la opción “Registrar”, el sistema a través de la interfaz de usuario, invoca a la clase controladora C\_Rol, esta verifica que el nombre del rol sea único consultándolo con la clase entidad E\_Rol, una vez que los datos hayan sido validados, invoca a la clase entidad E\_Rol nuevamente y registra al rol, la clase entidad E\_Rol confirma el registro y la clase controladora solicita a la clase IU\_Mantenimiento\_Roles presentar un mensaje confirmando el resultado de la operación.

## Diagrama de Colaboración – Eliminar

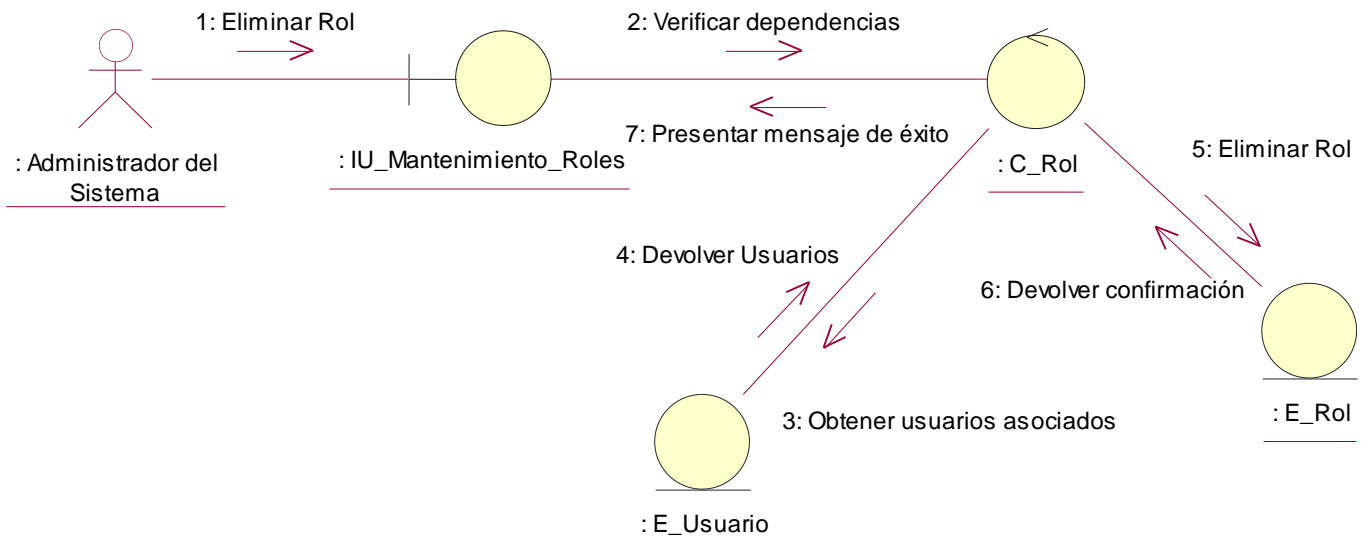


Figura 4.104 Diagrama de colaboración “Eliminar”

**Descripción del diagrama de colaboración – Eliminar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Roles selecciona la opción eliminar, la clase interfaz de usuario invoca a clase controladora C\_Rol que verifica que el rol no tenga registros asociados consultando si existen usuario con el rol a eliminar a la clase E\_Usuario, la clase E\_Usuario devuelve la lista de usuario con dicho rol, en caso de que no hayan registros asociados, la clase controladora solicita a la entidad E\_Rol eliminar el rol seleccionado, la clase entidad E\_Rol confirma la eliminación a la clase controladora, quien solicita presentar la confirmación a través de un mensaje a la clase IU\_Mantenimiento\_Roles.

### 4.2.6.7 CUA – Asociar Permisos a Roles

#### Diagrama de Clases

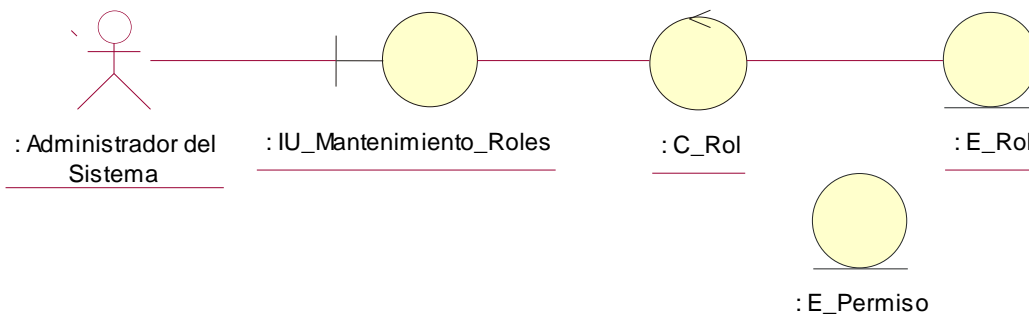


Figura 4.105 Diagrama de clases del caso de uso



## Diagrama de Colaboración – Asociar

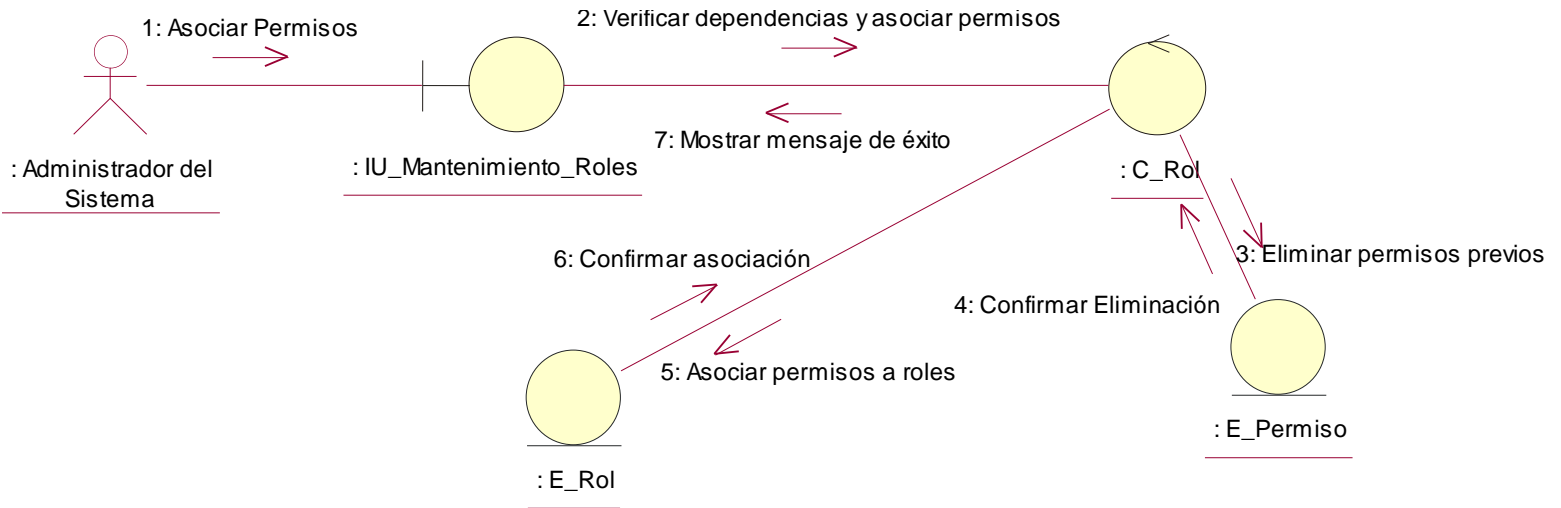


Figura 4.106 Diagrama de colaboración “Asociar”

**Descripción del diagrama de colaboración – Asociar:** El usuario a través de la interfaz de usuario IU\_Mantenimiento\_Roles escoge los permisos que estarán asignados al rol, y selecciona la opción “Establecer Permisos”, la interfaz de usuario solicita a la clase controladora C\_Rol eliminar cualquier permiso previo asignado al rol, solicitando a la clase entidad E\_Permission eliminar cualquier asociación entre el rol y algún permiso, posteriormente, la clase C\_Rol solicita a la clase E\_Rol asociar los nuevos permisos seleccionados, la clase entidad E\_Rol confirma a la clase controladora que la asociación ya fue realizada, posteriormente la clase controladora solicita a la clase interfaz de usuario presentar un mensaje confirmando la operación.

## 4.2.7 Diagrama de clases entidades

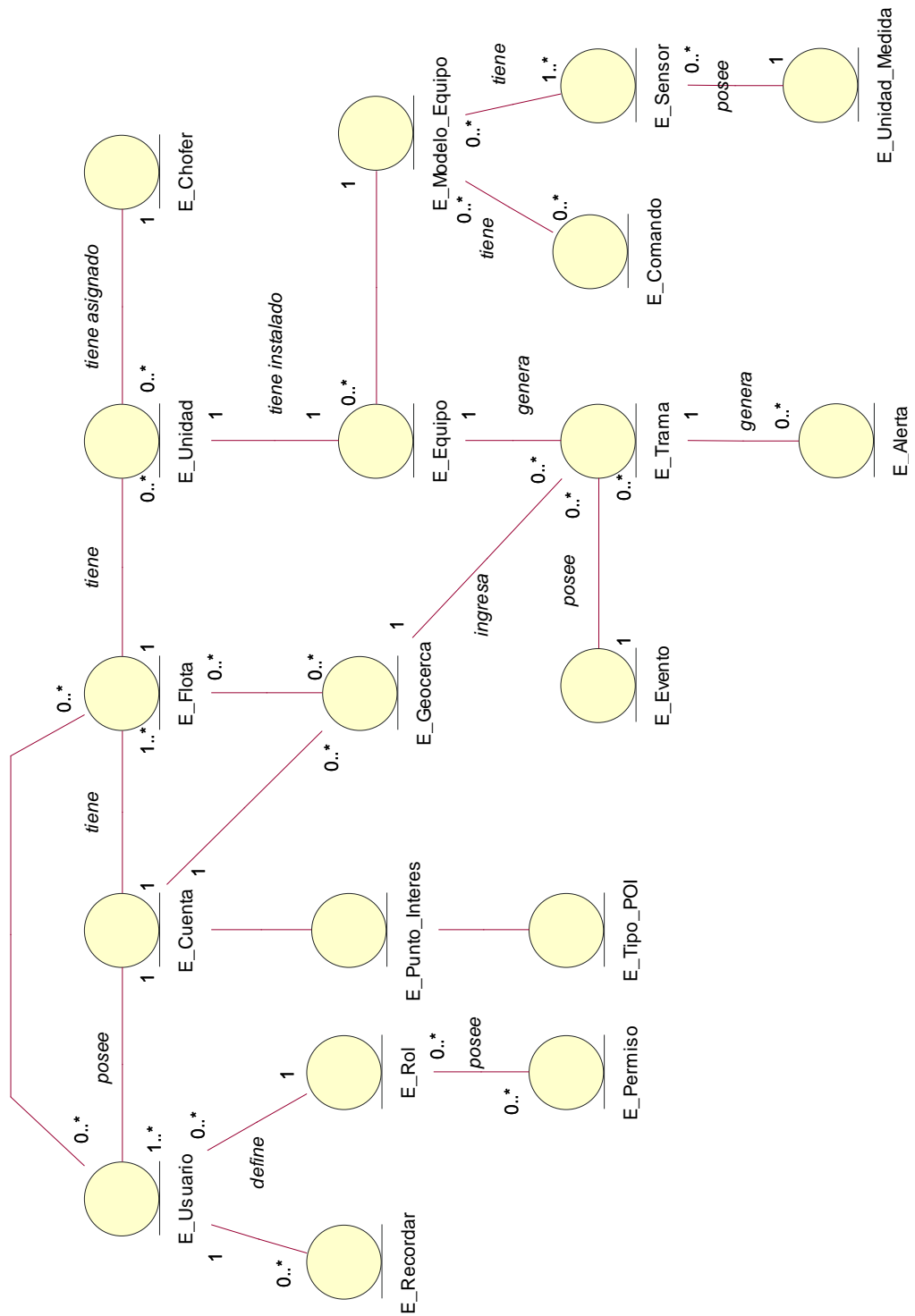


Figura 4.107 Diagrama de Clases Entidades

### 4.3 Diseño del sistema

#### 4.3.1 Acerca del diseño del sistema

El diseño de sistema describe la interacción real de los diversos componentes que conforman el sistema, al igual que la implementación de un edificio empieza por los planos y demás detalles para que al final se dé inicio a la construcción, de igual manera empezaremos a describir uno a uno los artefactos que serán el fundamento a la construcción del sistema, entre estos tenemos la arquitectura del sistema, los diagramas de clases y los diagramas de secuencia de los casos de uso, el diagrama de clases de acceso a datos, el diseño físico de la base de datos y el diagrama de componentes del sistema, todos estos artefactos serán herramientas útiles al momento de implementar el sistema de manera objetiva y puntual.

#### 4.3.2 Diseño de la Arquitectura del Sistema

##### Diagrama de distribución de la Arquitectura

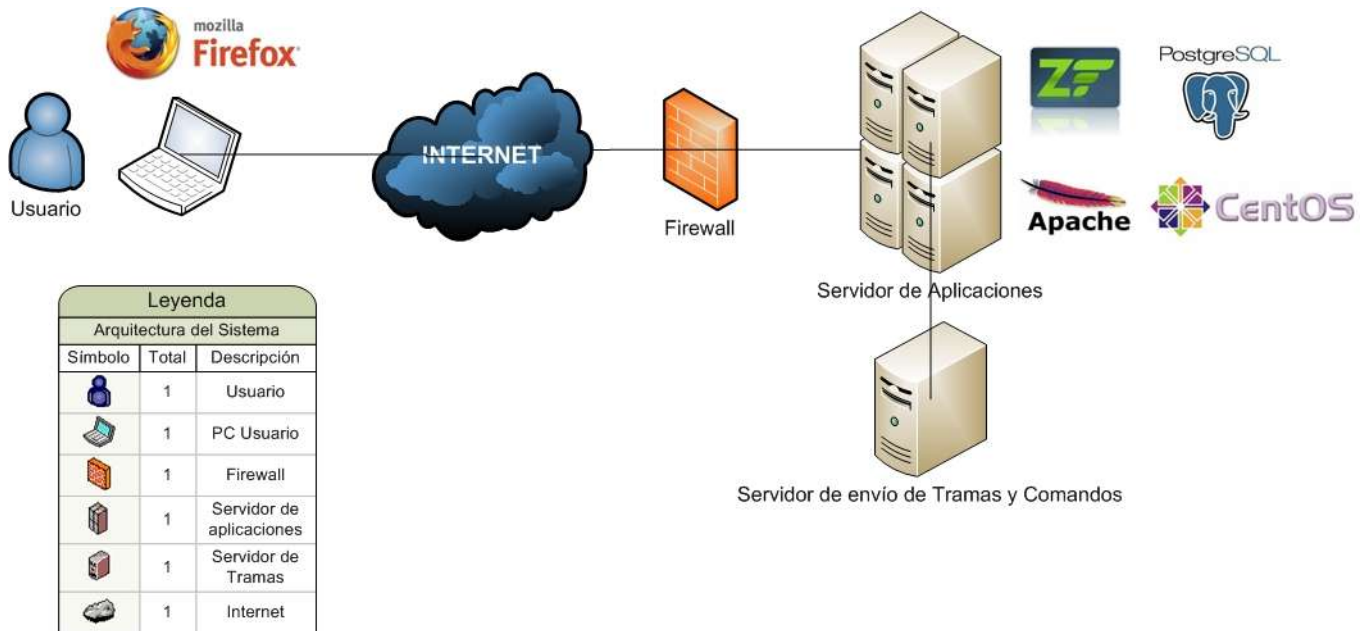


Figura 4.108 Diagrama de Distribución

#### Especificación de los Nodos

<u>Servidor de Aplicaciones</u>	
<b>Especificación del hardware</b>	
<b>Nombre:</b>	HP ProLiant DL360 G7 Intel Xeon E5649 (6 núcleos, 2,53 GHz, 80 W) (633776-421)
<b>Procesador:</b>	Intel® Xeon® E5649 (6 núcleos, 2,53 GHz, 12 MB L3, 80 W)

<b>Memoria:</b>	6 GB
<b>Controlador de Almacenamiento:</b>	(1) Smart Array P410i/256 MB
<b>Disco Duro</b>	2 discos de 500 GB y 2 discos de 140 GB
<b>Especificación del Software</b>	
<b>Sistema Operativo:</b>	Centos 5.6 64 bits
<b>Servidor Web:</b>	Apache 2.0
<b>Base de Datos:</b>	PostgreSQL 8
<b>Lenguaje de Programación:</b>	PHP 5 – Zend Framework

<b><u>Servidor de envío de tramas y comandos</u></b>	
<b>Especificación del hardware</b>	
<b>Nombre:</b>	Servidor HP ProLiant ML330 G6
<b>Procesador:</b>	Procesador Intel® Xeon® E5506
<b>Memoria:</b>	1x2GB DDR3(U-Dimms)
<b>Disco Duro</b>	2 discos de 250GB HP SATA LFF
<b>Especificación del Software</b>	
<b>Sistema Operativo:</b>	Ubuntu
<b>Base de Datos:</b>	MySQL 5
<b>Lenguaje de Programación:</b>	Java

## Diagrama de capas del sistema

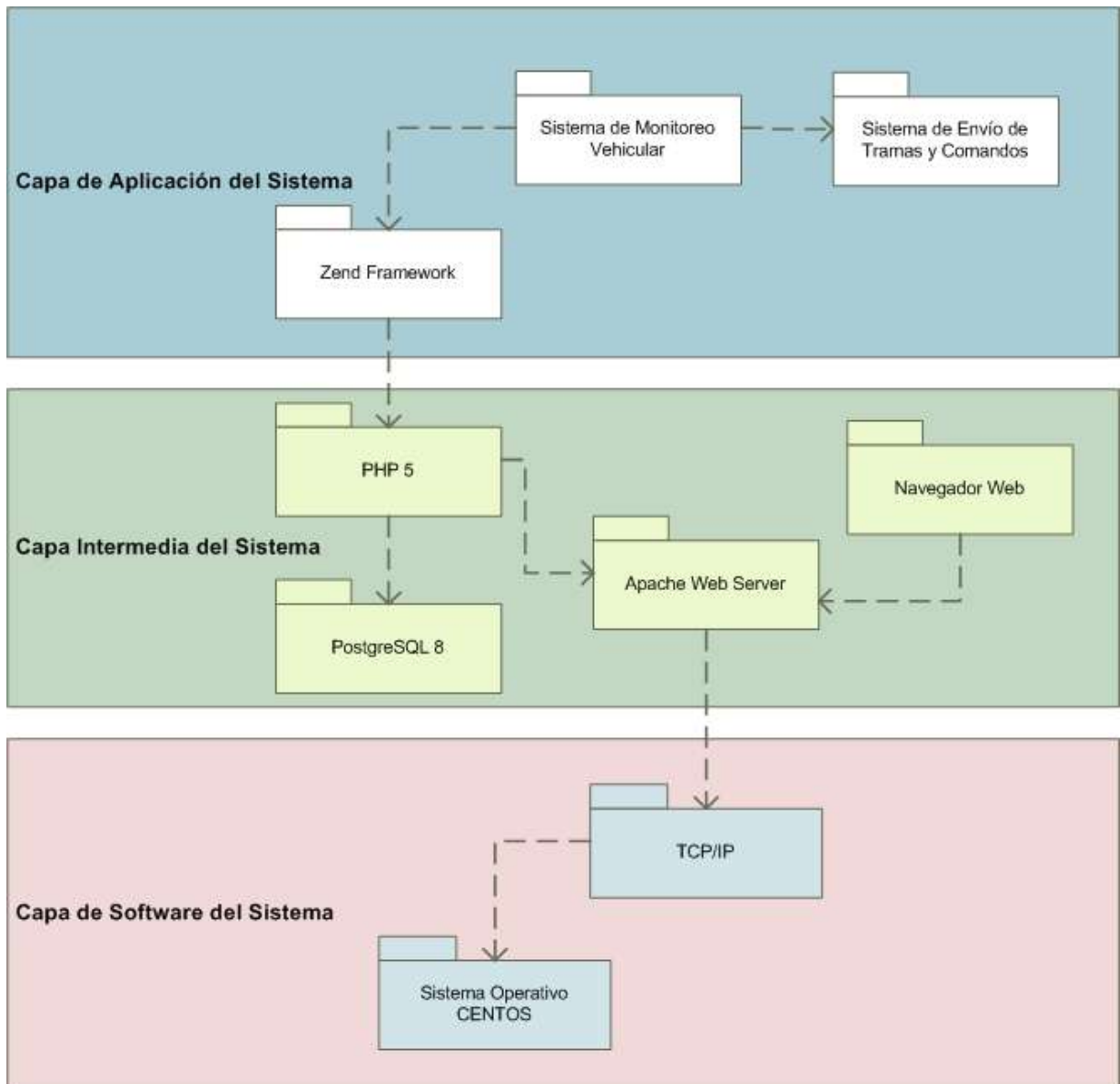


Figura 4.109 Diagrama de Capas del Sistema

### 4.3.3 Realización de casos de uso de diseño

En la realización de casos de uso de análisis se listaron los casos de uso del sistema junto a su respectiva realización en casos de uso de análisis, si bien en el análisis del sistema se describió una propuesta lógica que intentaba asegurar que una posible implementación era factible, en la realización de los casos de uso del diseño

mostraremos la realización de manera objetiva ilustrando el comportamiento interno del sistema y sus componentes.

A continuación se mostrarán la realización de casos de uso agrupados por paquetes:

Paquete	Caso de Uso de Análisis	Caso de Uso de Diseño
Consulta	CUA_Ver Mapa	CUD_Ver Mapa
	CUA_Ver Unidades	CUD_Ver Unidades
	CUA_Seleccionar Cliente en Mapa	CUD_Seleccionar Cliente en Mapa
	CUA_Ver Detalle de Posición	CUD_Ver Detalle de Posición
	CUA_Ver Recorrido	CUD_Ver Recorrido
	CUA_Ver Alertas	CUD_Ver Alertas
	CUA_Atender Alertas	CUD_Atender Alertas
	CUA_Ver Comandos	CUD_Ver Comandos
	CUA_Enviar Comandos	CUD_Enviar Comandos
	CUA_Ver Punto de Interes	CUD_Ver Punto de Interes
	CUA_Buscar Unidad	CUD_Buscar Unidad
CUA_Buscar Punto de Interes	CUD_Buscar Punto de Interes	

Paquete	Caso de Uso de Análisis	Caso de Uso de Diseño
Configuración	CUA_Mantenimiento de Modelo de Equipos	CUD_Mantenimiento de Modelo de Equipos
	CUA_Mantenimiento de Eventos	CUD_Mantenimiento de Eventos
	CUA_Mantenimiento de Comandos	CUD_Mantenimiento de Comandos
	CUA_Asociar Comando a Modelo de Equipo	CUD_Asociar Comando a Modelo de Equipo
	CUA_Mantenimiento de Tipo de Punto de Interés	CUD_Mantenimiento de Tipo de Punto de Interés

Paquete	Caso de Uso de Análisis	Caso de Uso de Diseño
Administración	CUA_Asociar Geocerca a Flota	CUD_Asociar Geocerca a Flota
	CUA_Mantenimiento de Clientes	CUD_Mantenimiento de Clientes
	CUA_Mantenimiento de Conductores	CUD_Mantenimiento de Conductores
	CUA_Mantenimiento de Equipos de Localización	CUD_Mantenimiento de Equipos de Localización
	CUA_Mantenimiento de Flotas	CUD_Mantenimiento de Flotas
	CUA_Mantenimiento de Geocercas	CUD_Mantenimiento de Geocercas
	CUA_Mantenimiento de Puntos de Interés	CUD_Mantenimiento de Puntos de Interés

Paquete	Caso de Uso de Análisis	Caso de Uso de Diseño
Reportes	CUA_Generar Reportes de Alertas	CUD_Generar Reportes de Alertas
	CUA_Generar Reporte de Comandos	CUD_Generar Reporte de Comandos
	CUA_Generar Reporte de Recorrido	CUD_Generar Reporte de Recorrido

Paquete	Caso de Uso de Análisis	Caso de Uso de Diseño
Seguridad	CUA_Actualizar Cuenta	CUD_Actualizar Cuenta
	CUA_Ingresar al Sistema	CUD_Ingresar al Sistema
	CUA_Mantenimiento de Permisos	CUD_Mantenimiento de Permisos
	CUA_Mantenimiento de Roles	CUD_Mantenimiento de Roles
	CUA_Mantenimiento de Usuarios	CUD_Mantenimiento de Usuarios
	CUA_Recuperar Contraseña	CUD_Recuperar Contraseña
	CUA_Salir del Sistema	CUD_Salir del Sistema

#### 4.3.4 Descripción de la plataforma del sistema

##### 4.3.4.1 Plataforma de la aplicación

**Zend Framework:** En la actualidad las soluciones de software están basados en una plataforma, en este ámbito se entiende por plataforma al conjunto de componentes (librerías, clases, scripts, etc.) que implementan las mejores prácticas a fin de dar solución a un problema, Zend Framework es una plataforma basada en PHP, orientada a objetos y que incluye los mejores patrones de software en aplicaciones web, en la presente investigación aplicada, utilizaremos la versión 1 de la plataforma, de tal manera que las clases y demás componentes estarán sujetas a trabajar en armonía con esta plataforma, por lo tanto, si en el análisis los casos de uso estaban agrupados por paquetes, en esta ocasión, las clases estarán agrupadas por subsistemas que especifica la plataforma.

##### 4.3.4.2 Diagrama de paquetes del sistema

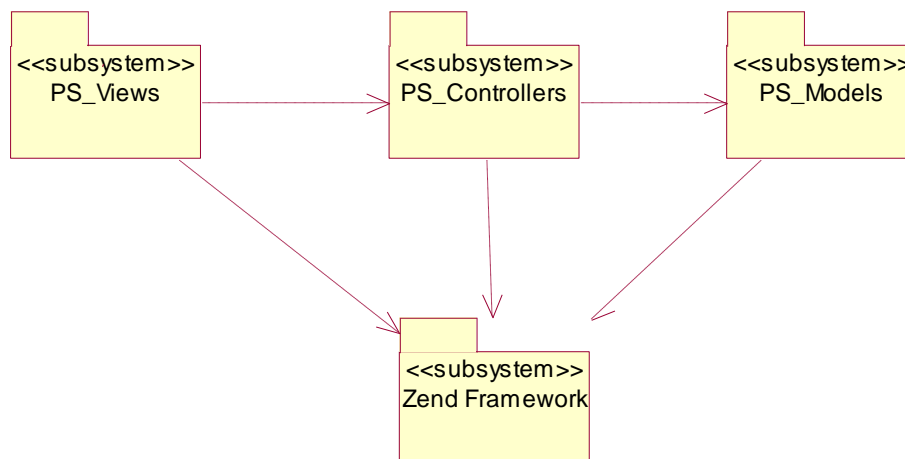


Figura 4.110 Diagrama de paquetes del sistema

En la realización de los casos de uso describiremos 3 tipos de clases: clases de tipo vista, clases tipo controladora, y clases de acceso a datos, estas clases tendrán atributos y métodos, no obstante Zend Framework nos invita a agrupar las clases en subsistemas, entre ellos tenemos:

**PS\_Views:** Agrupará clases de tipo vista estas clases implementarán lenguaje PHP y lenguaje javascript, las clases con PHP estarán alojadas en el servidor de aplicaciones y las clases que contengan javascript serán descargadas por el navegador web, la descripción de cómo estas serán ubicadas, estará descrito en el diagrama de componentes.

**PS\_Controllers:** Agrupará clases de tipo controlador en el lenguaje PHP, estas clases se comunicarán directamente con las clases de tipo vista y las clases de acceso a datos.

**PS\_Models:** Agrupará clases de tipo acceso a datos, como su nombre lo indica, estas clases se conectarán con la base de datos y ejecutarán las consultas que las controladoras le exijan.

**Zend Framework:** Es el conjunto de clases que servirán de plataforma a los 3 subsistemas anteriormente mencionados, por ejemplo las clases de vista que implementan código PHP, heredarán de la clase VIEW, las clases de tipo controlador heredarán de la clase CONTROLLER, y las clases de tipo acceso a datos heredarán de la clase MODEL, de esta forma las clases implementadas gozarán de todas las ventajas que ofrece la plataforma Zend Framework.



### 4.3.5 Realización de casos de uso de diseño – Paquete Consulta

#### 4.3.5.1 Caso de Uso – Ver mapa

##### Diagrama de Clases

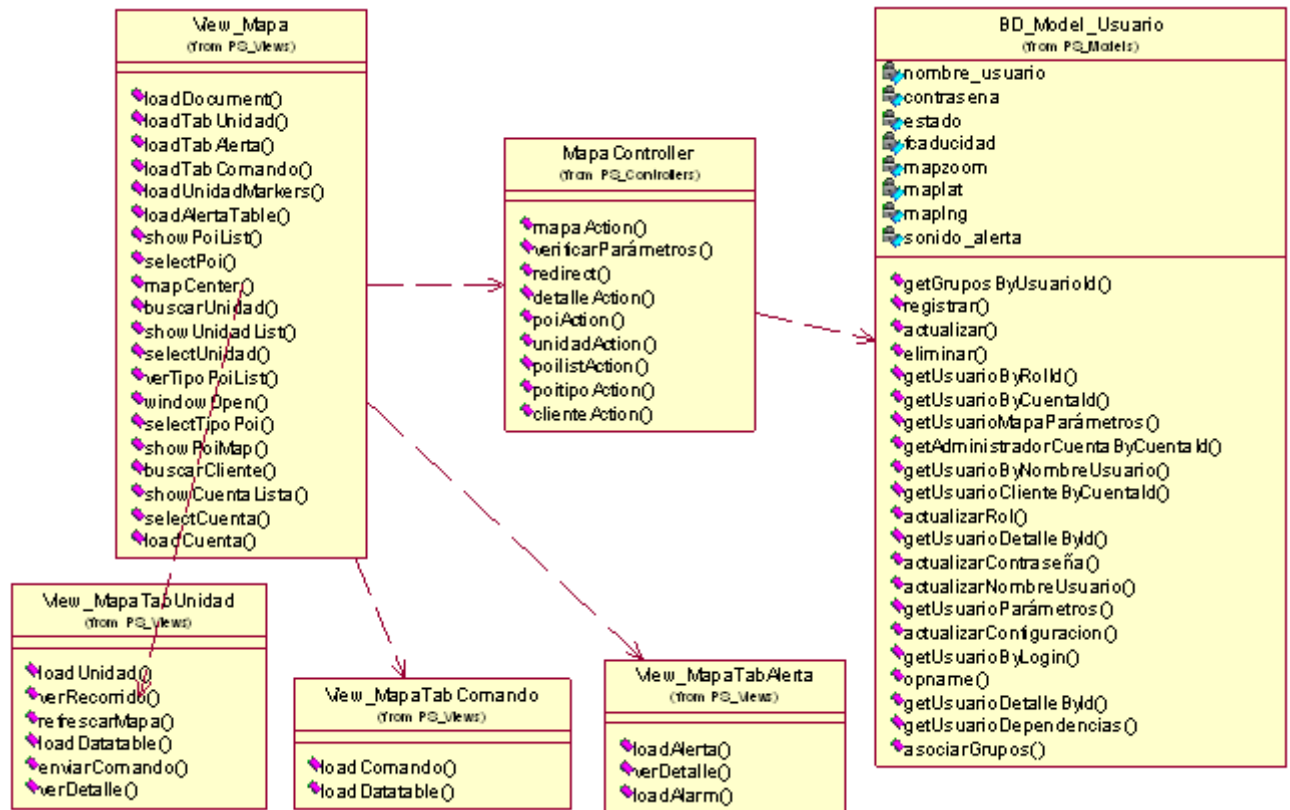


Figura 4.111 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Mapa

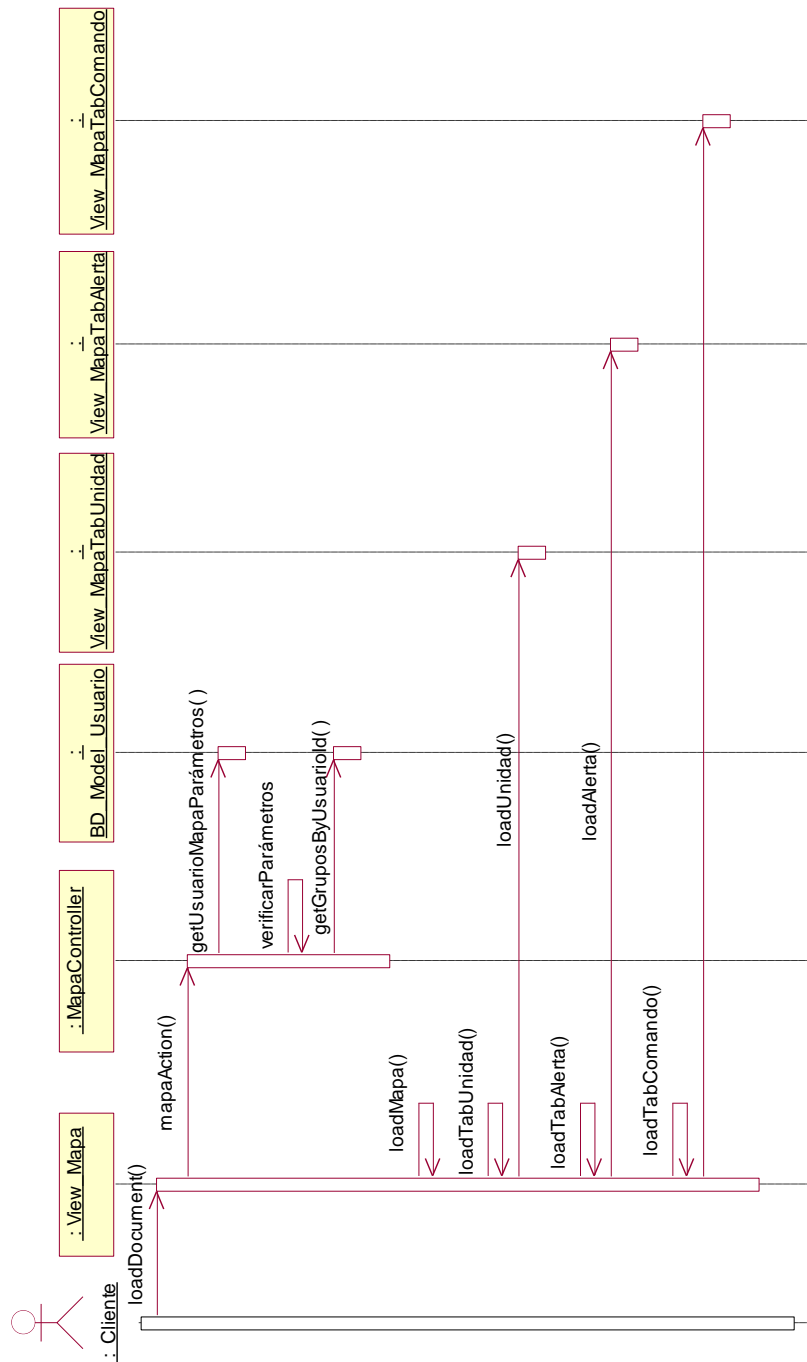


Figura 4.112 Diagrama de secuencia “Ver Mapa”

**Descripción del diagrama de secuencia – Ver Mapa:** El usuario al seleccionar la opción “Mapa” del menú de Consulta, invoca a la clase View\_Map cargando la

función loadDocument() que ejecuta la carga de la página, cuando la página esta cargando, la clase View\_Mapa hace una llamada a la clase MapaController que ejecuta la función mapaAction, la cual obtiene los parámetros del usuario a través de la clase BD\_Model\_Usuario, luego verifica los parámetros, tomando en cuenta de que el usuario tenga flotas asignadas, en caso de que el usuario sea administrador del sistema, las flotas asignadas serán nulas y podrá cargar todas las flotas de cualquier cuenta cliente, una vez que la página este cargada, se ejecutará la función javascript loadMapa, la cual invocará a la función loadTabUnidad de la clase View\_MapaTabUnidad y cargará las unidades en el panel de unidades y mostrará las última posición de cada unidad en el mapa, llamará la función loadTabAlerta de la clase View\_MapaTabAlerta que carga las alertas de las unidades de las flotas en el panel de alertas y llama a la función loadTabComando de la clase View\_MapaTabComando, que se encarga de cargar los comandos enviados a las unidades de las flotas en el panel de comandos.

### 4.3.5.2 Caso de Uso – Ver Unidades

#### Diagrama de Clases

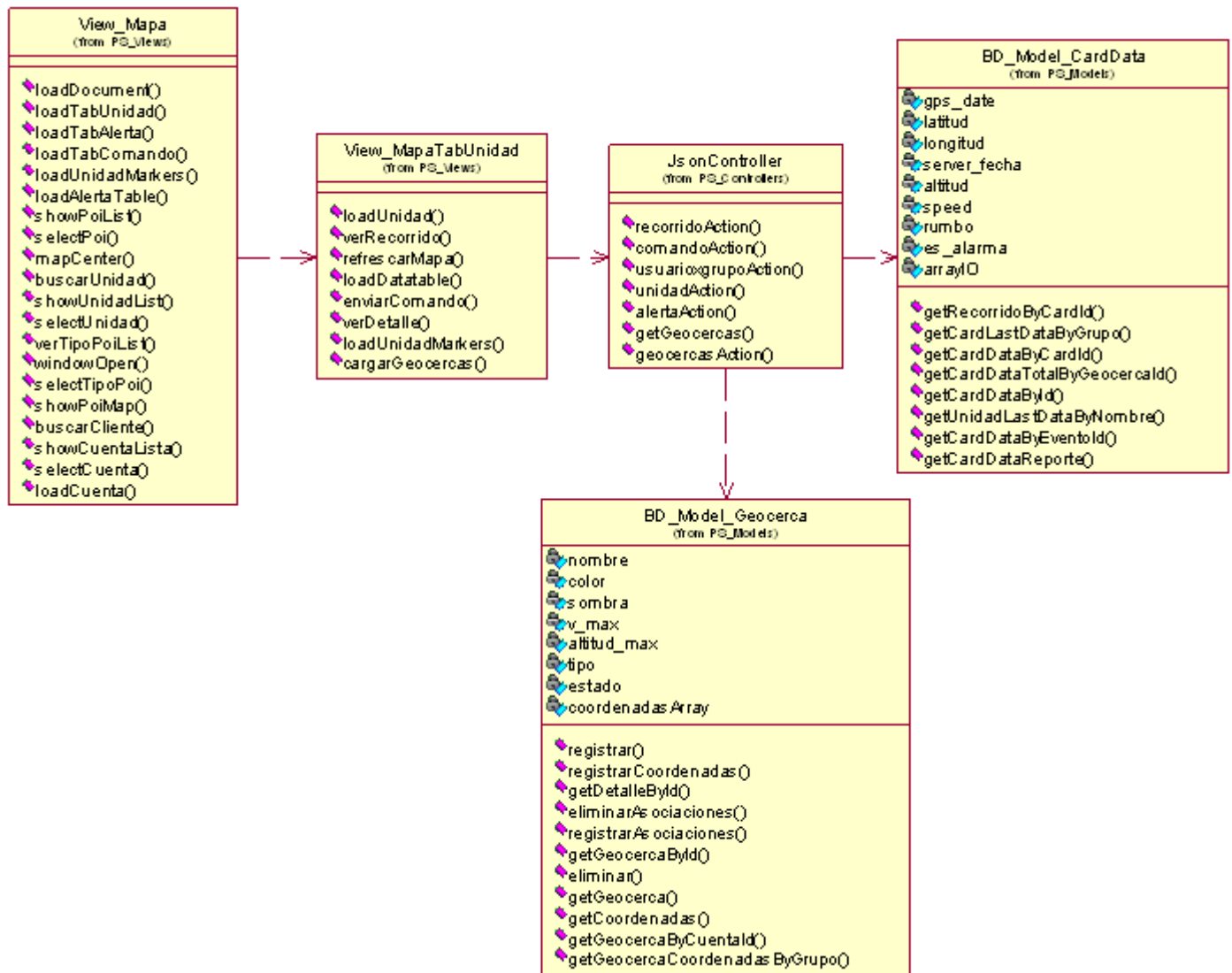


Figura 4.113 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Unidades

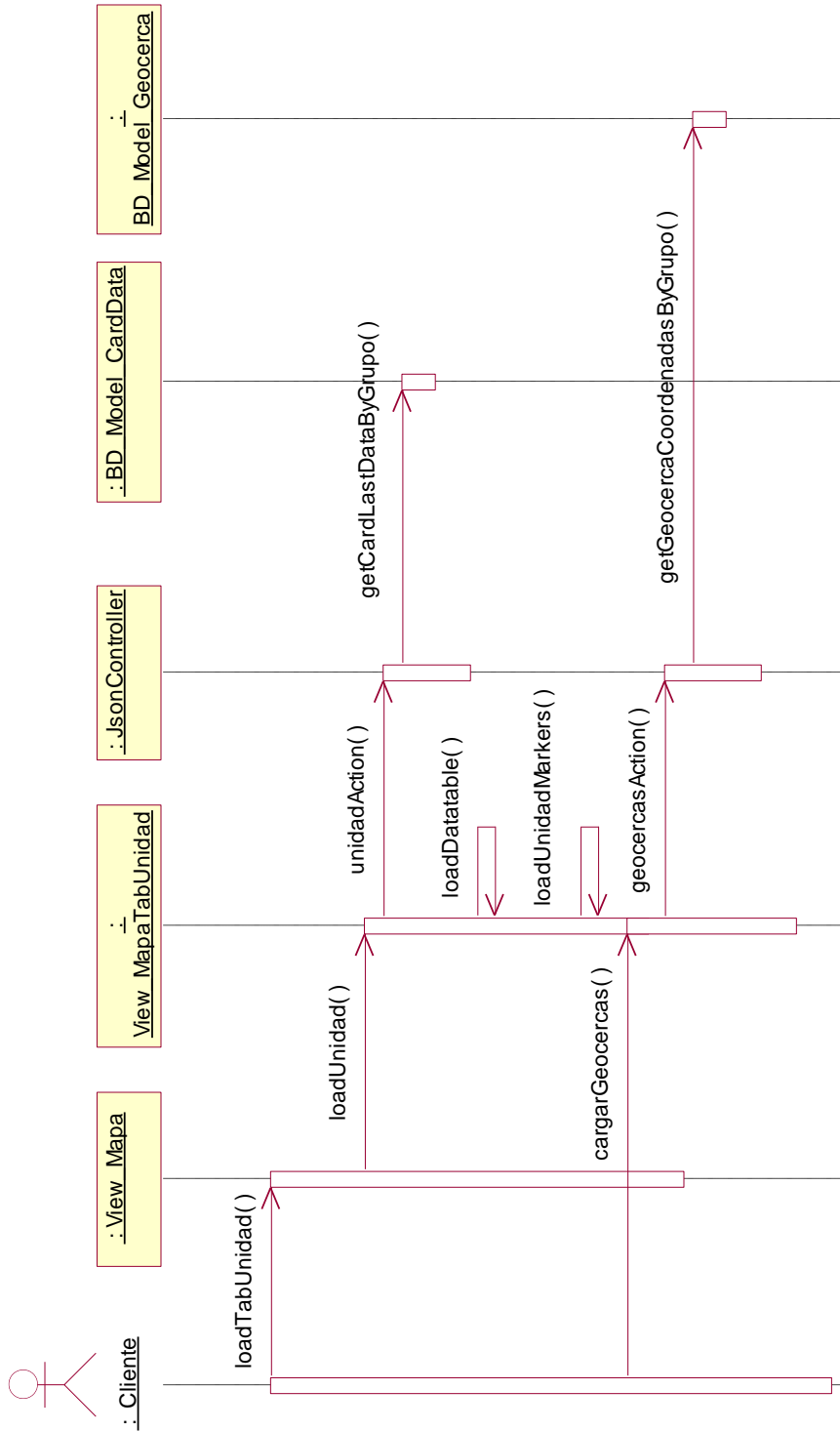
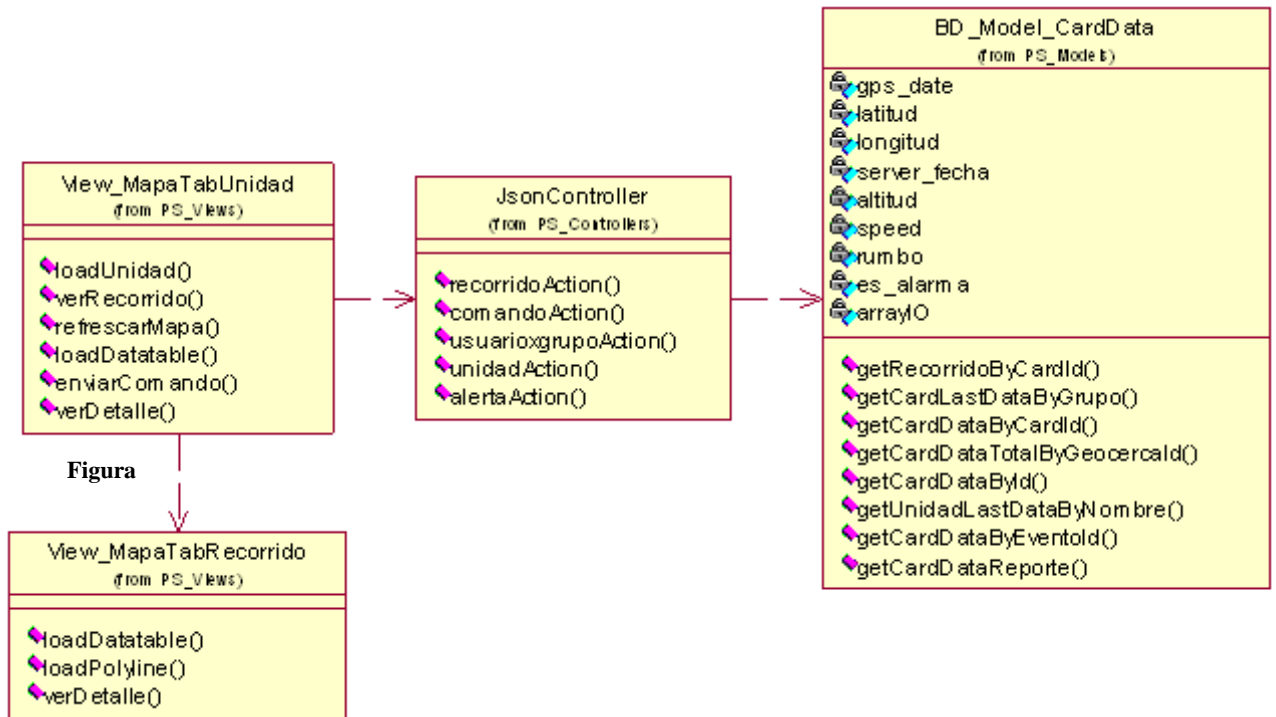


Figura 4.114 Diagrama de secuencia “Ver Unidades”

**Descripción del diagrama de secuencia – Ver Unidades:** El usuario al seleccionar la opción “Mapa” del menú de Consulta, invoca a la clase View\_Mapa, una vez que la página de ver mapa está cargada, la clase View\_Mapa, invoca a la clase javascript View\_MapaTabUnidad, la cual deberá obtener la última posición de cada unidad vehicular junto con sus datos básicos, para eso invoca a la clase JsonController a través de una llamada Ajax ejecutando la función unidadAction, la cual obtiene de la clase BD\_Model\_CardData las últimas tramas emitidas en función al array de grupos de unidades enviados como parámetro, una vez que la clase JsonController tiene el conjunto de registros, prepara una descarga tipo json la cual la función loadUnidad de la clase View\_MapaTabUnidad pueda leer, una vez que se tienen los datos de las unidades, carga los datos en el panel a través de la función loadDatatable y carga las posiciones en el mapa a través de la función loadUnidadMarkers, esta función también invoca a la librería de clusterización llamada MapCluster, que se encarga de agrupar las unidades cercanas en el mapa en íconos circulares en función a su nivel de zoom, cuando se hace clic en un marcador del mapa, este presenta en una ventana de información la misma información que existe en la tabla de unidades ubicada en el panel. Si el en la ventana de información hace clic en la opción “Geocerca”, este evento ejecuta la función cargarGeocercas de la clase View\_MapaTabUnidad, esta función hace una llamada tipo ajax a la función geocercasAction de la clase JsonController, la cual solicita la clase BD\_Model\_Geocerca a través de la función getGeocercaCoordenadasByGrupo las coordenadas de las geocercas de las flotas que el usuario tiene asignado, una vez que la función geocercasAction tiene los registros, ejecuta la descarga de un archivo tipo json, el cual la función cargarGeocercas utiliza y dibuja la geocerca en el mapa web como un objeto tipo polígono.

### 4.3.5.3 Caso de Uso – Ver Recorrido

#### Diagrama de Clases



4.115 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Recorrido

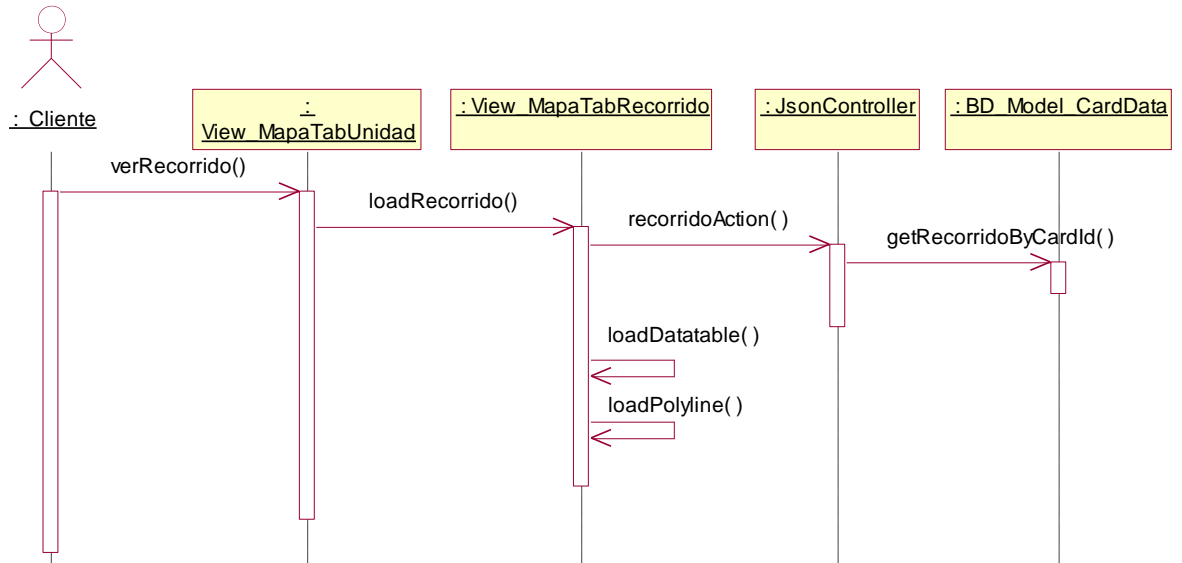


Figura 4.116 Diagrama de secuencia “Ver Recorrido”

**Descripción del diagrama de secuencia – Ver Recorrido:** Cuando el usuario en el panel de la unidad manejado por la clase `View_MapasTabUnidad` (clase en javascript), selecciona la opción “Ver Ruta”, ya sea en el panel de la unidad o en la ventana de información cuando hace clic en un marcador (ya que es la misma ventana de información en ambos lados), se ejecuta la función `verRecorrido` de la clase `View_MapasTabUnidad`, esta función realiza una instancia de la clase `View_MapasTabRecorrido` (clase en javascript) ejecutando la función `loadRecorrido` (que se encarga de hacer una llamada tipo ajax a la clase `JsonController` ejecutando la función `recorridoAction` que recibe como parámetros el Id del Card (Equipo de Localización), dentro de la función `recorridoAction`, se hace una llamada a la función `BD_Model_CardData`, que realiza una consulta a la base de datos y obtiene el recorrido de las últimas 30 posiciones de aquel equipo de localización o Card, una vez que la función `recorridoAction` tiene los registros, genera un archivo tipo Json con los datos solicitados, la cual la clase `loadRecorrido` puede leer, luego la función crea un array de recorridos de equipos a fin de manejar recorridos simultáneos posteriormente dibuja el recorrido en el mapa a través de la función `loadPolyline`, los datos del panel son idénticos a los datos del recorrido dibujado en el mapa, así mismo la venta de información que se muestra al hacer clic en un marcador de recorrido.



### 4.3.5.4 Caso de Uso – Ver Alertas

#### Diagrama de Clases

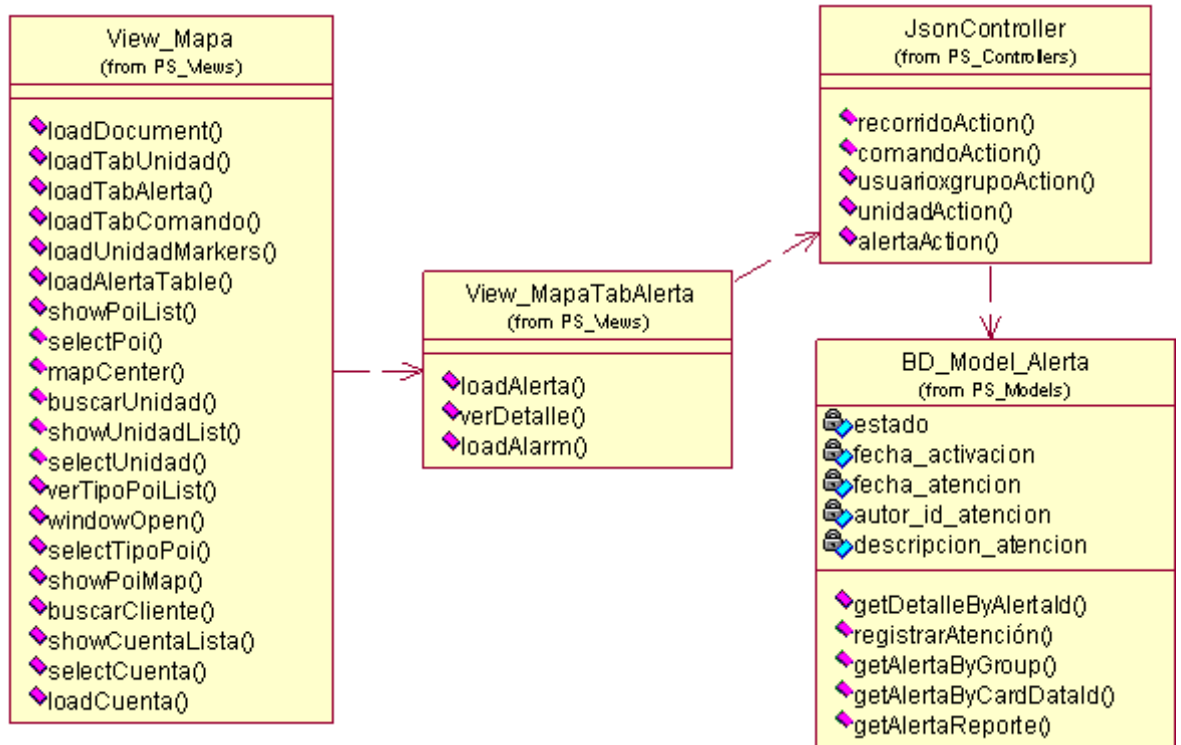


Figura 4.117 Diagrama de clases del caso de uso

#### Diagrama de Secuencia –Ver Alertas

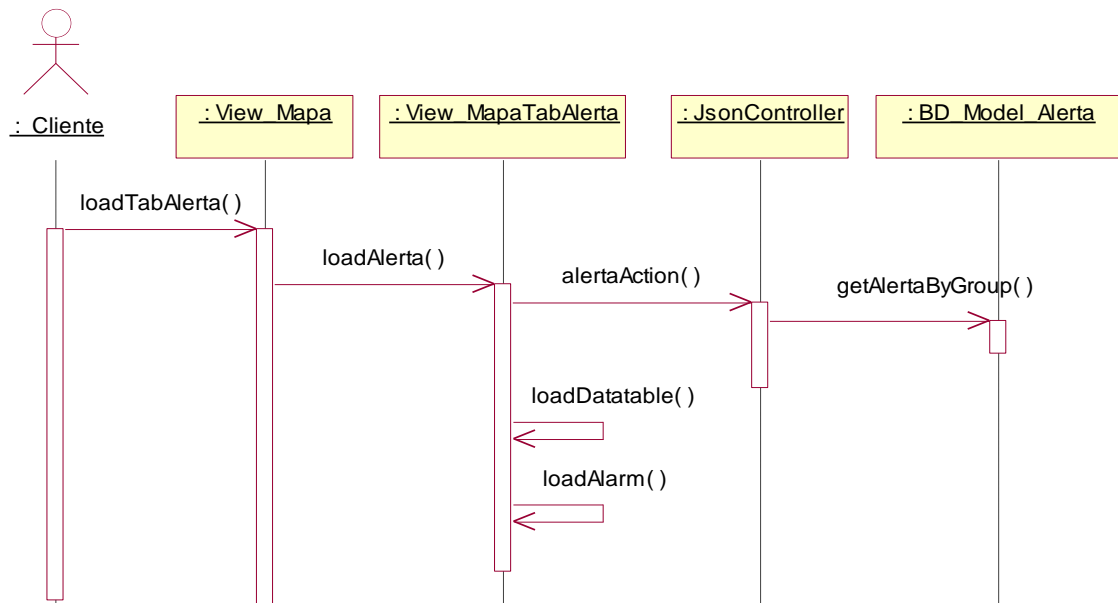


Figura 4.118 Diagrama de secuencia “Ver Alertas”

**Descripción del diagrama de secuencia – Ver Alertas:** El usuario al seleccionar la opción “Mapa” del menú de Consulta, invoca a la clase View\_Mapa, una vez que la página de ver mapa es cargada, la clase View\_Mapa, invoca a la clase javascript View\_MapaTabAlerta, la cual carga todas las alertas de las unidades de las flotas que el usuario tiene acceso a monitorear, para eso la función loadAlerta de la clase View\_MapaTabAlerta, hace una llamada tipo ajax a la función alertaAction de la clase JsonController la cual invoca a la case BD\_Model\_Alerta que ejecuta la función getAlertaByGroup la cual obtiene todas las alertas del array de grupos que fueron enviados como parámetros, una vez que la función alertaAction tiene los registros, permite la descarga de un archivo tipo json con los datos solicitados, una vez que la función loadAlerta de la clase View\_MapaTabAlerta tiene los registros carga los datos en el panel de alertas a través de la función loadDatatable, así mismo si existen alertas sin se ejecuta la función loadAlarma que ejecuta la opción play de un archivo MDI que tiene como función ejecutar una alarma sonora a fin de atender las alertas pendientes.

#### 4.3.5.5 Caso de Uso – Ver Comandos

##### Diagrama de Clases

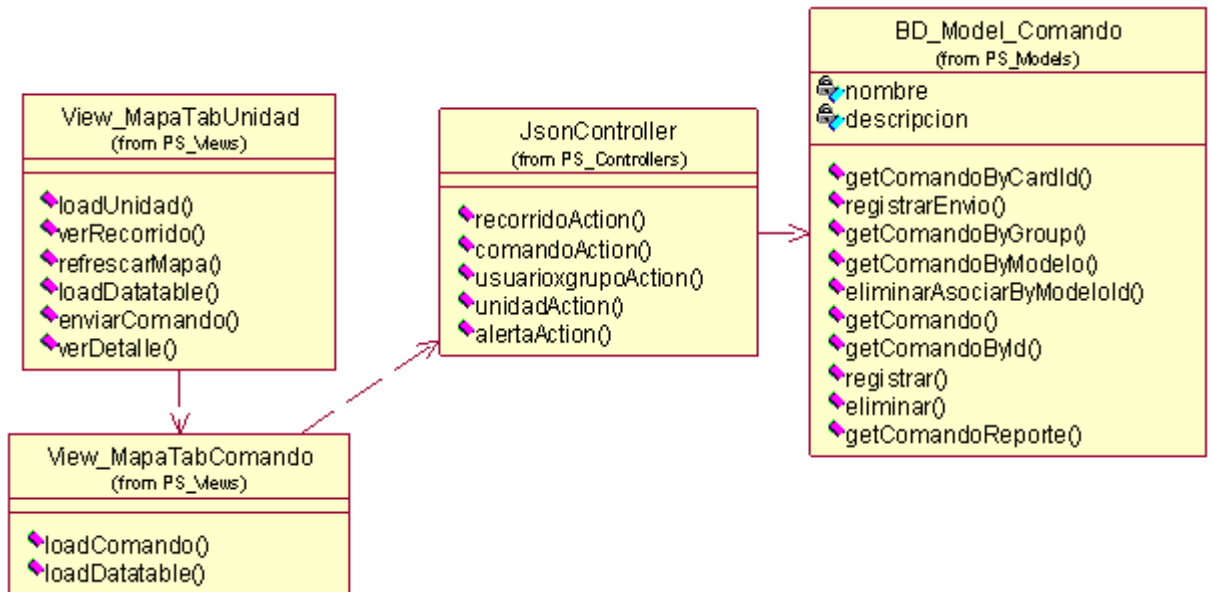


Figura 4.119 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Comandos

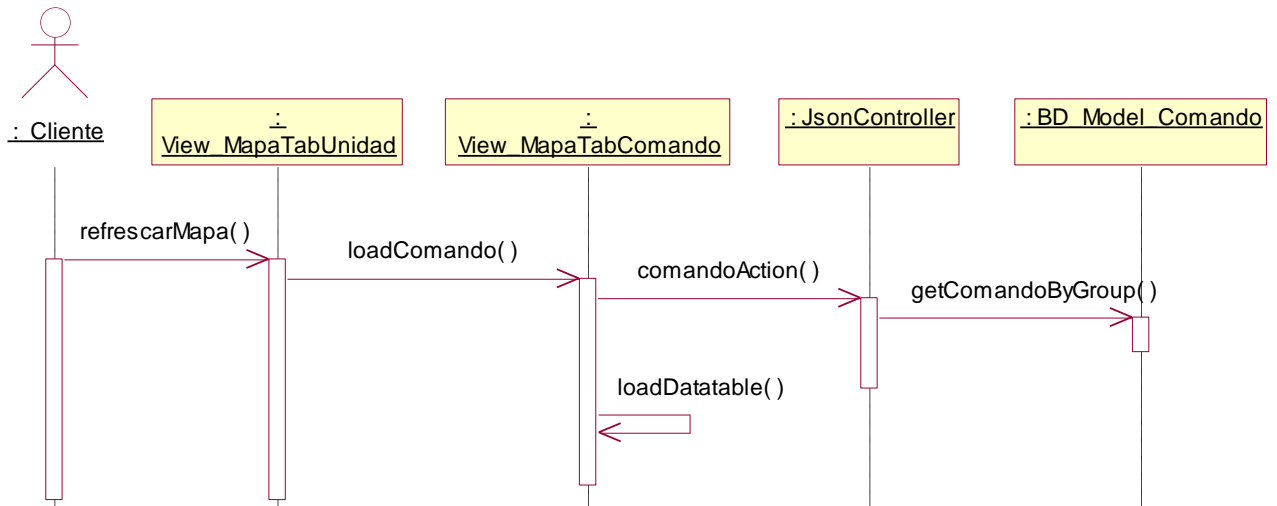


Figura 4.120 Diagrama de secuencia “Ver Comandos”

**Descripción del diagrama de secuencia – Ver Comandos:** El usuario al seleccionar la opción “Mapa” del menú de Consulta, invoca a la clase View\_Mapa, una vez que la página de ver mapa es cargada, la clase View\_Mapa, invoca a la clase javascript View\_MapaTabComando, la cual carga todos los comandos enviados a las unidades de las flotas que el usuario tiene acceso a monitorear, a través de la función loadComando de la clase View\_MapaTabComando, se hace una llamada tipo ajax a la clase controladora JsonController solicitándole un archivo tipo json con los datos de los comandos enviados, para eso la función comandoAction de la clase JsonController hace una llamada a la función getComandoByGroup de la clase BD\_Model\_Comando a fin de obtener los comandos enviados a las unidades de las flotas del usuario, para los identificadores de los grupos de unidades fueron enviados como array, una vez que la función comandoAction posee los registros solicitados, ejecuta la descarga de un archivo tipo Json con los comandos enviados, cuando la función loadComando puede obtener los registros ejecuta la función loadDatatable a fin de cargar los datos recibidos en el panel de comandos.

### 4.3.5.6 Caso de Uso – Enviar comando

#### Diagrama de Clases

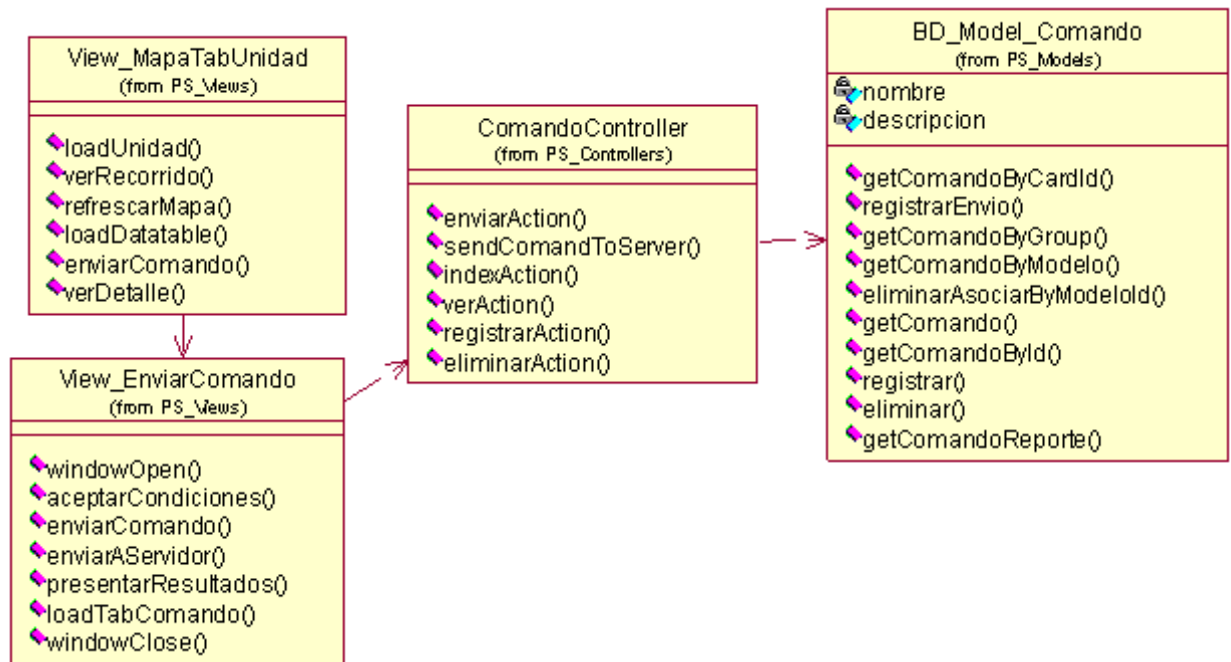


Figura 4.121 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Enviar Comando

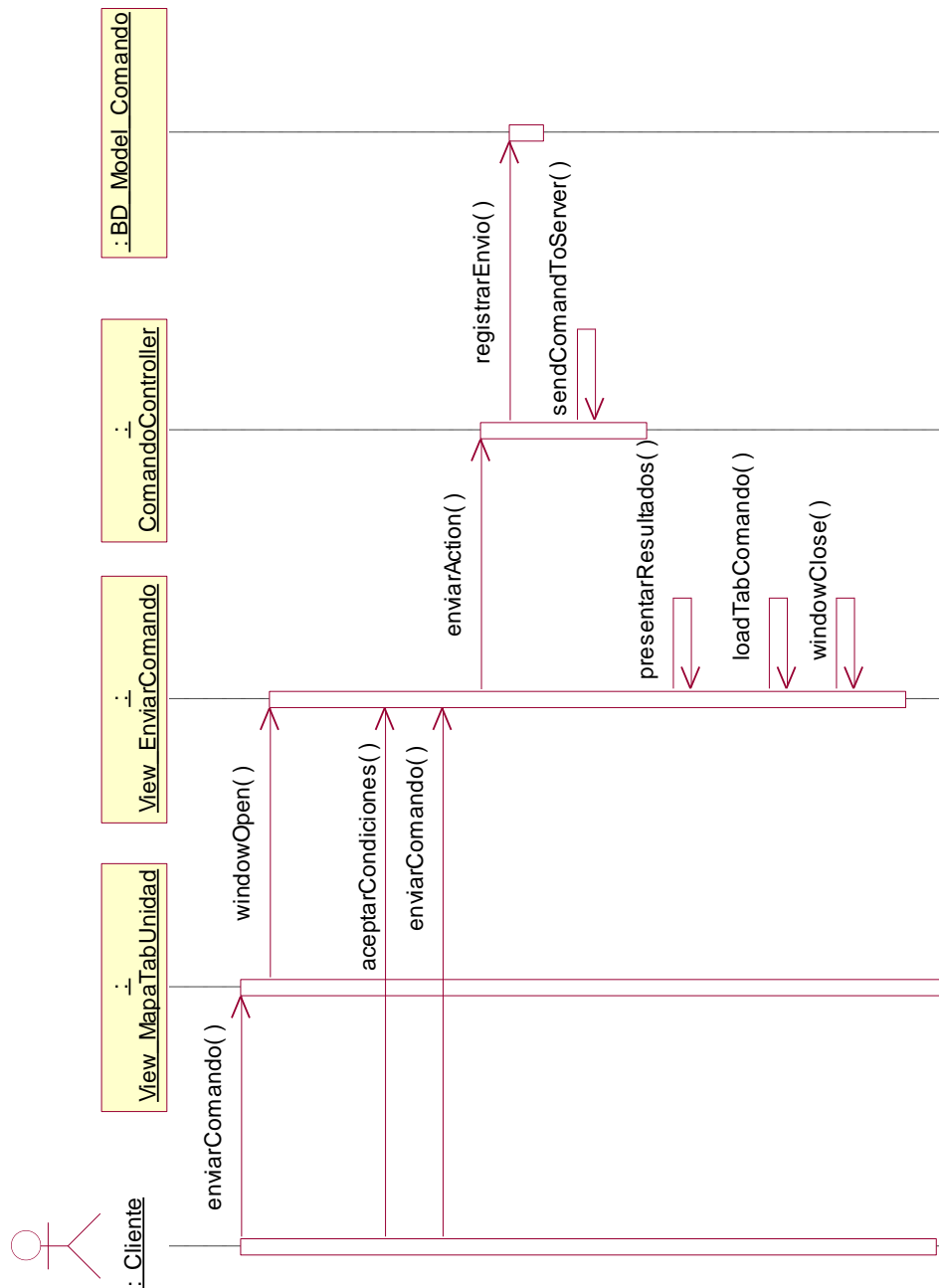


Figura 4.122 Diagrama de secuencia “Enviar Comando”

**Descripción del diagrama de secuencia – Enviar Comando:** El usuario cuando utiliza el panel de unidades o hace clic en algún marcador de unidad en el mapa, tiene la opción “Comando” cuando el usuario hace clic en dicha opción, la clase `View_MapaTabUnidad` ejecuta la función `enviarComando`, el cual invoca la

apertura de la ventana con la función `windowOpen` instanciando a la clase `View_EnviarComando`, esta clase presenta los comandos que el usuario puede enviar al equipo de localización, una vez que el usuario haya seleccionado el comando a enviar, hace clic en el botón “aceptar condiciones” y posteriormente hace clic en el botón “Enviar Comando” que ejecuta la función `enviarComando`, esta función hace un llamado a la clase `ComandoController` y ejecuta la función `enviarAction`, la función `enviarAction` registra el envío a través de la función `registrarEnvio` de la clase `BD_Model_Comando`, posteriormente la función `enviarAction` ejecuta la función `sendComandToServer` que se encarga de hacer una llamada al servidor de “Envío de Tramas y Comandos” donde ejecuta el envío del comando con los parámetros especificados, posteriormente la clase `View_EnviarComando` presenta los resultados del envío del comando, hacer una llamada a la función `loadTabComando` de la clase `View_MapTabComando` y se cierra por sí sola a través de la función `windowClose`, posteriormente el usuario puede visualizar el comando enviado en el panel de comandos enviados.

### 4.3.5.7 Caso de Uso – Ver Detalle de la Posición

#### Diagrama de Clases

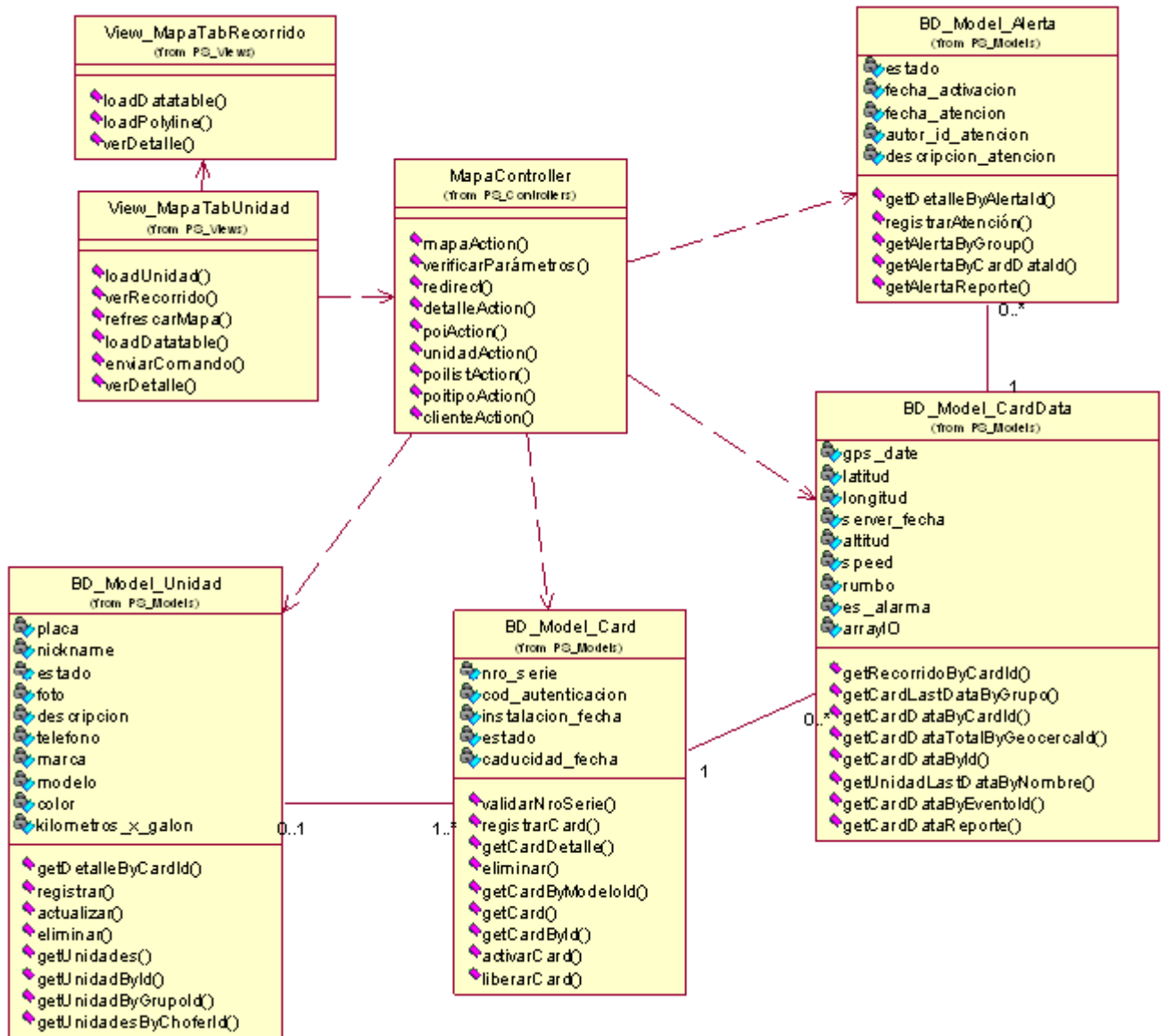


Figura 4.123 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Detalle de la Posición

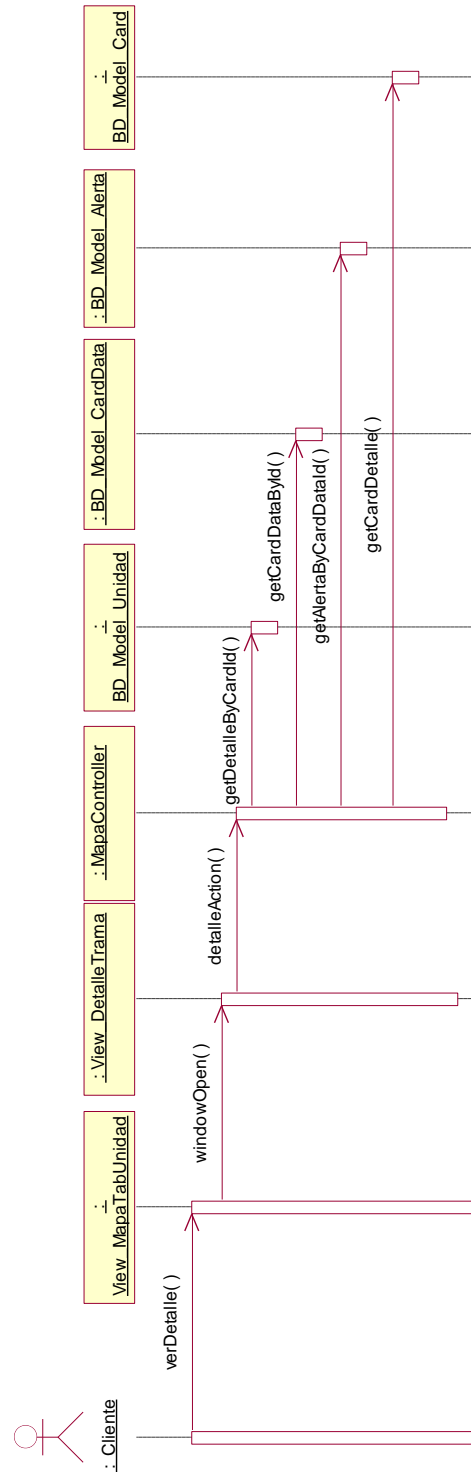


Figura 4.124 Diagrama de secuencia “Ver Detalle de la Posición”



**Descripción del diagrama de secuencia – Ver Detalle de la Posición:** El usuario cuando utiliza el panel de unidades o hace clic en algún marcador de unidad en el mapa o en algún marcador de un recorrido, tiene la opción “Detalle” cuando el usuario hace clic en dicha opción, la clase View\_MapaTabUnidad ejecuta la función verDetalle, el cual invoca la apertura de la ventana con la función windowOpen instanciando a la clase View\_DetalleTrama, esta clase presenta el detalle de la transmisión de la unidad vehicular, al momento de instanciarse a la clase View\_DetalleTrama, ejecuta la función detalleAction de la clase MapaController, esta función obtiene el detalle del equipo de localización mediante la función getDetalleByCardId de la clase BD\_Model\_Unidad, así mismo obtiene el detalle de la trama a través de la función getCardDataById de la clase BD\_ModelCardData, también obtiene el detalle de la alerta (en caso de haberla) con la función getAlertaByCardDataId de la clase BD\_Model\_Alerta, y por último obtiene el detalle del equipo de localización con la función getCardDetalle de la clase BD\_Model\_Card, una vez que tiene todos estos datos recopilados, procede a mostrar la posición en un mapa cargado en la venta recién abierta junto con la información obtenida de las diferentes clases en un tabs separados por objetos divs tipos CSS.

#### 4.3.5.8 Caso de Uso – Atender Alerta

##### Diagrama de Clases

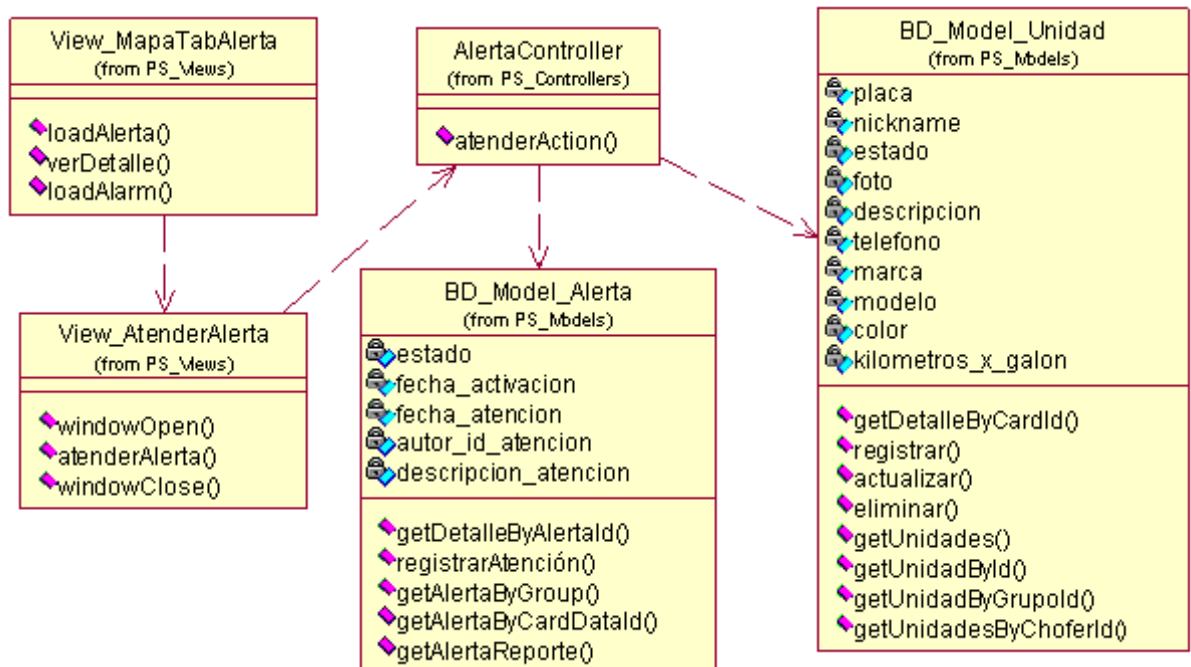


Figura 4.125 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Atender Alerta

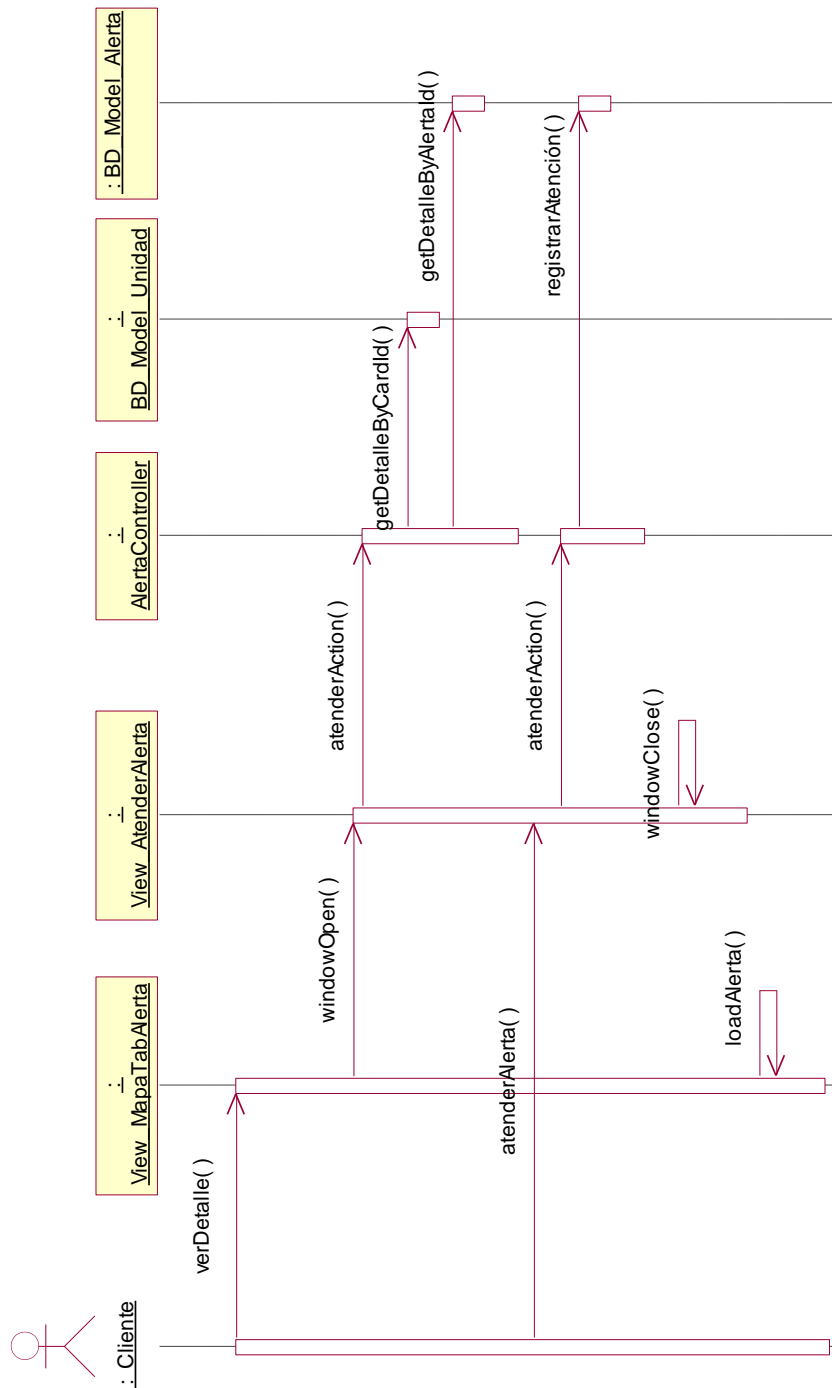


Figura 4.126 Diagrama de secuencia “Atender Alerta”

**Descripción del diagrama de secuencia – Atender Alerta:** El usuario cuando utiliza el panel de unidades, el panel de alertas o hace clic en algún marcador de unidad en el mapa o en algún marcador de un recorrido, tiene la opción “Detalle”

cuando el usuario hace clic en dicha opción, la clase View\_MapaTabUnidad ejecuta la función verDetalle, el cual invoca la apertura de la venta con la función windowOpen instanciando a la clase View\_DetalleTrama, en caso de que la trama tenga alguna alerta, en la carga de la pestaña alerta, se invoca a la función atenderAction de la clase AlertaController, esta función obtiene el detalle de la unidad a través de la función getDetalleByCardId de la clase BD\_Model\_Unidad y obtiene el detalle de la alerta a través de la función getDetalleByAlertaId de la clase BD\_Model\_Alerta, una vez que los datos son presentados, la ventana View\_AtenderAlerta (que es instanciada en la pestaña de alerta de la ventana de detalle de la posición) presenta un formulario donde el usuario puede ingresar la descripción de la atención de la alerta, cuando el usuario ingresa la descripción y hace clic en el botón “Atender Alerta”, se ejecuta la función atenderAlerta de la clase View\_AtenderAlerta que invoca a la función atenderAction de la clase AlertaController, esta clase captura la descripción y los datos del usuario que está atendiendo la alerta y las registra a través de la función registrarAtención de la clase BD\_Model\_Alerta, una vez que la alerta está registrada la ventana que instancia a la clase View\_AtenderAlerta se cierra con la función windowClose y se carga nuevamente las alertas del panel de alertas con la función loadAlerta.

#### 4.3.5.9 Caso de Uso – Buscar punto de interés

##### Diagrama de Clases

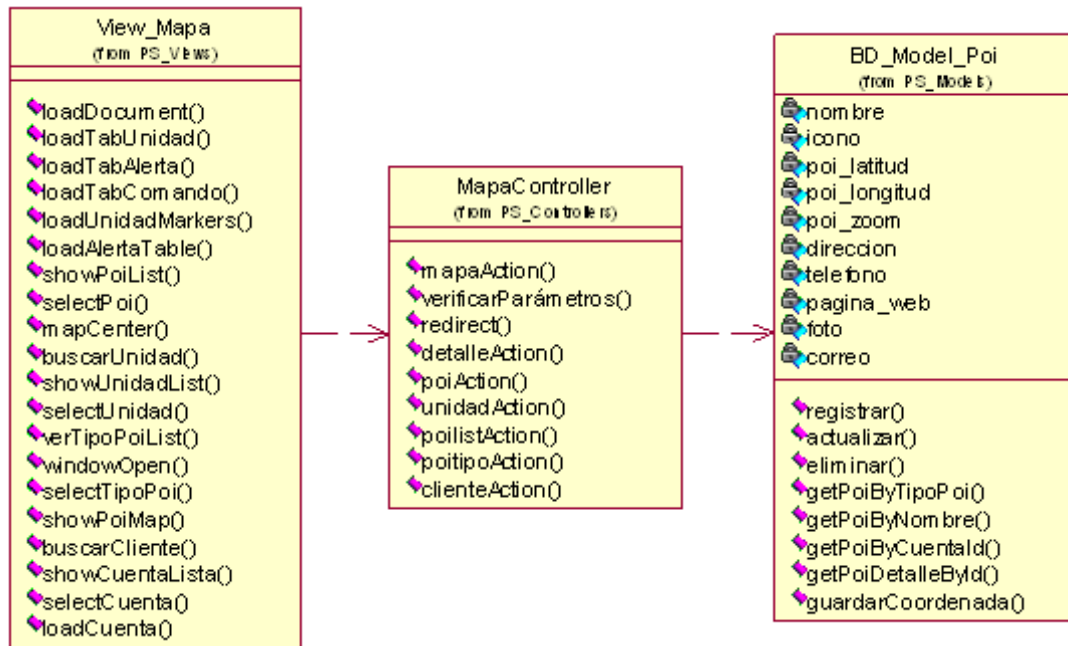


Figura 4.127 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Buscar Punto de Interés

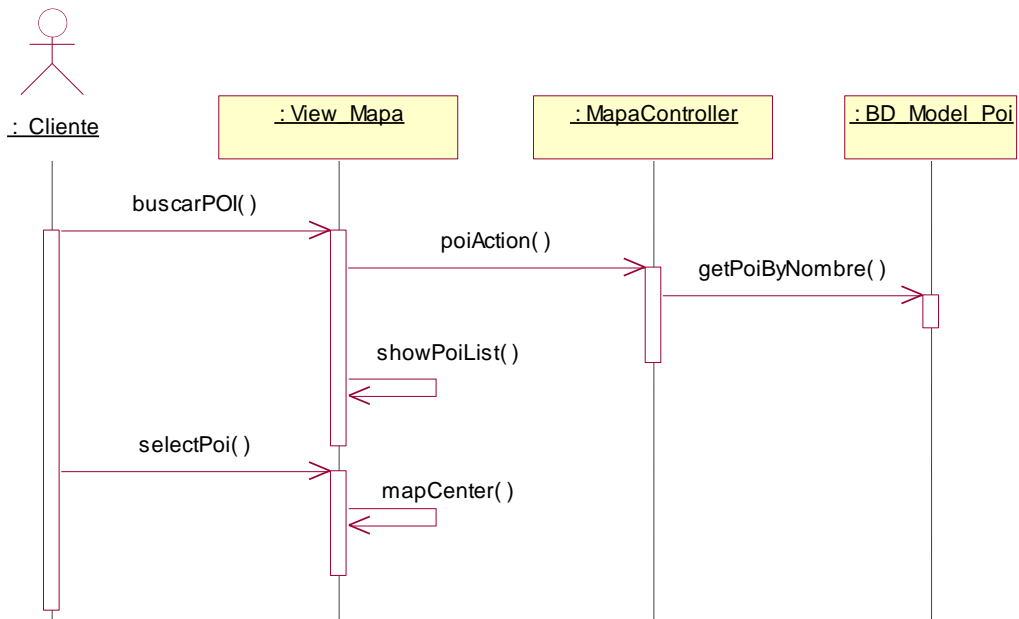


Figura 4.128 Diagrama de secuencia “Buscar Punto de Interés”

**Descripción del diagrama de secuencia – Buscar Punto de Interés:** El usuario en la caja de texto de la búsqueda de POI, escribe el nombre de un punto de interés, este evento ejecuta la función javascript buscarPOI de la clase View\_Mapa, que mediante una llamada ajax invoca a la función poiAction de la clase MapaController solicitándole un archivo tipo json la lista de puntos de interés que coincidan con el nombre ingresado, la función poiAction hace una llamada a la función getPoiByNombre de la clase BD\_Model\_Poi que devuelve la lista de puntos de interés que coinciden con el nombre escrito, cuando la función poiAction posee los registros con los puntos de interés, ejecuta la descarga del archivo tipo json, posteriormente cuando la función buscarPOI posee la lista de registros, los muestra al usuario con una lista tipo `<ul><li>contenido</li></ul>` al usuario. El usuario selecciona un punto de interés de la lista, este evento ejecuta la función selectPoi de la clase View\_Mapa y tomando como parámetros la latitud y longitud del punto de interés llama a la función mapCenter de la misma clase a fin de centrar el mapa en función a la latitud y longitud del punto de interés.

### 4.3.5.10 Caso de Uso – Buscar Unidad

#### Diagrama de Clases

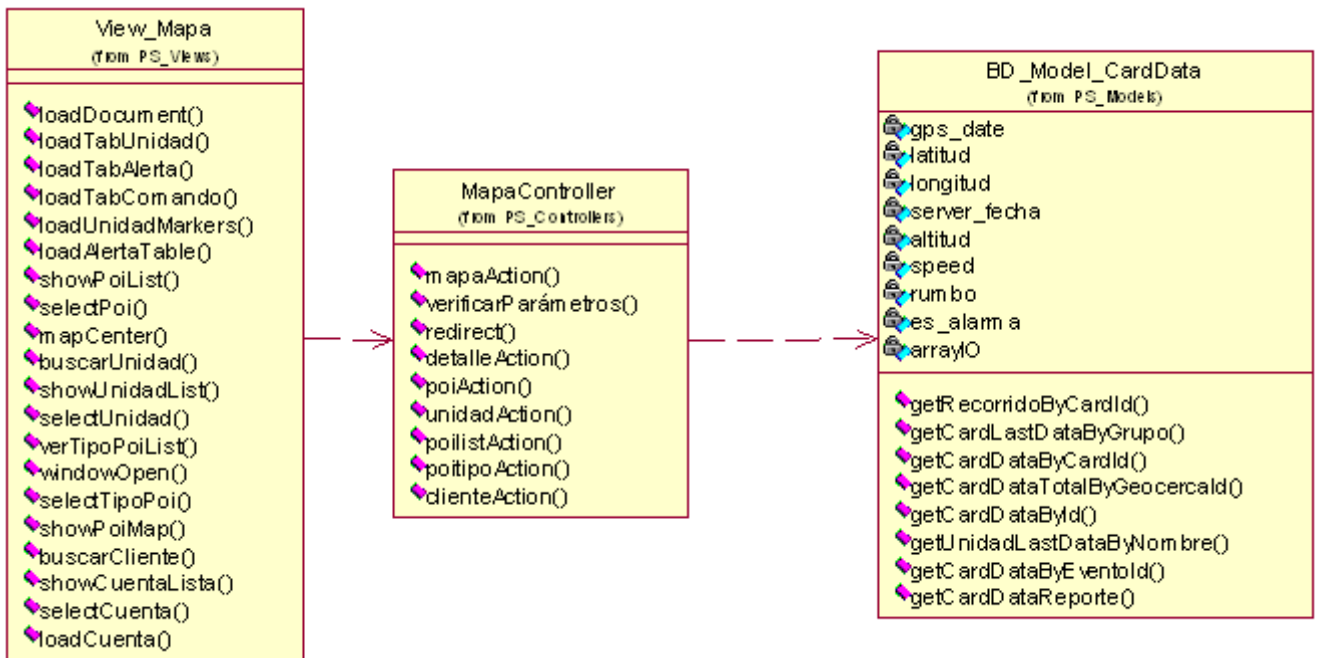


Figura 4.129 Diagrama de clases del caso de uso

#### Diagrama de Secuencia – Buscar Unidad

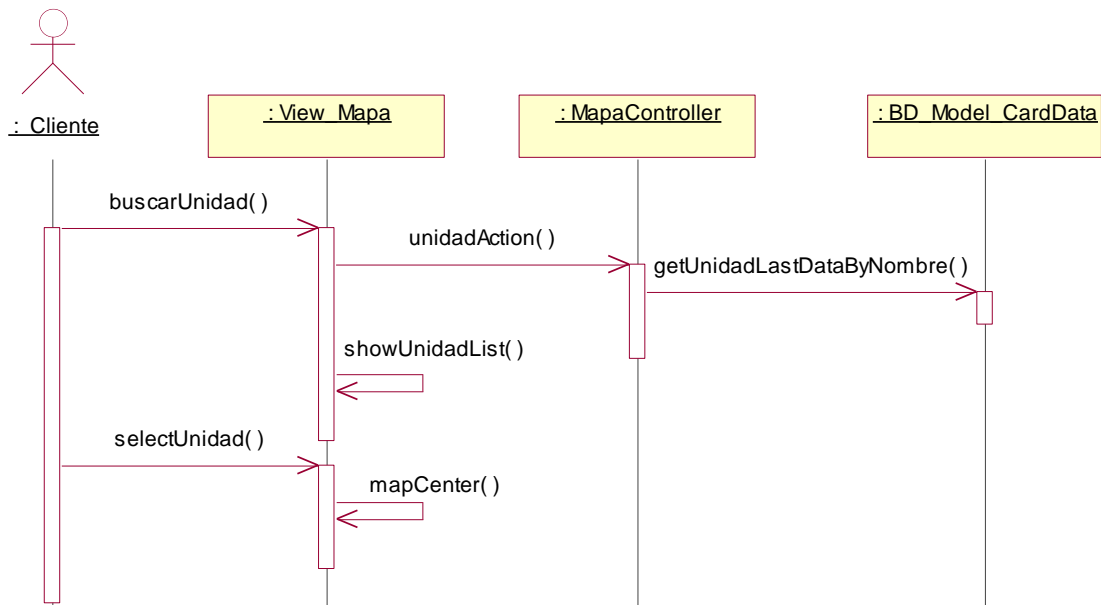


Figura 4.130 Diagrama de secuencia “Buscar Unidad”

**Descripción del diagrama de secuencia – Buscar Unidad:** El usuario en la caja de texto de la búsqueda de unidades vehiculares, escribe la placa o nickname de una unidad vehicular, este evento ejecuta la función javascript buscarUnidad de la clase View\_Mapa, la función hace una llamada ajax a fin de descargar un archivo tipo

json con las unidades que coincidan con el texto ingresado, para esto, se instancia a la función unidadAction de la clase MapaController que invoca a su vez a la función getUnidadLastDataByNombre de la clase BD\_Model\_CardData que se encarga de obtener el detalle de la última posición emitida por unidad en función a la placa o nickname, una vez que la función unidadAction tiene los registros, ejecuta la descarga de un archivo tipo json con los registros obtenidos, cuando la función buscarUnidad posee la lista de unidades que coinciden con el texto, muestra una lista de selección con el siguiente formato CSS <ul><li>contenido</li></ul>, una vez que una unidad es seleccionada, se ejecuta el evento selectUnidad, que envía la latitud y longitud de la unidad vehicular a la función mapCenter que se encarga de centrar el mapa en función a los parámetros recibidos.

#### 4.3.5.11 Caso de Uso – Ver Punto de Interés

##### Diagrama de Clases

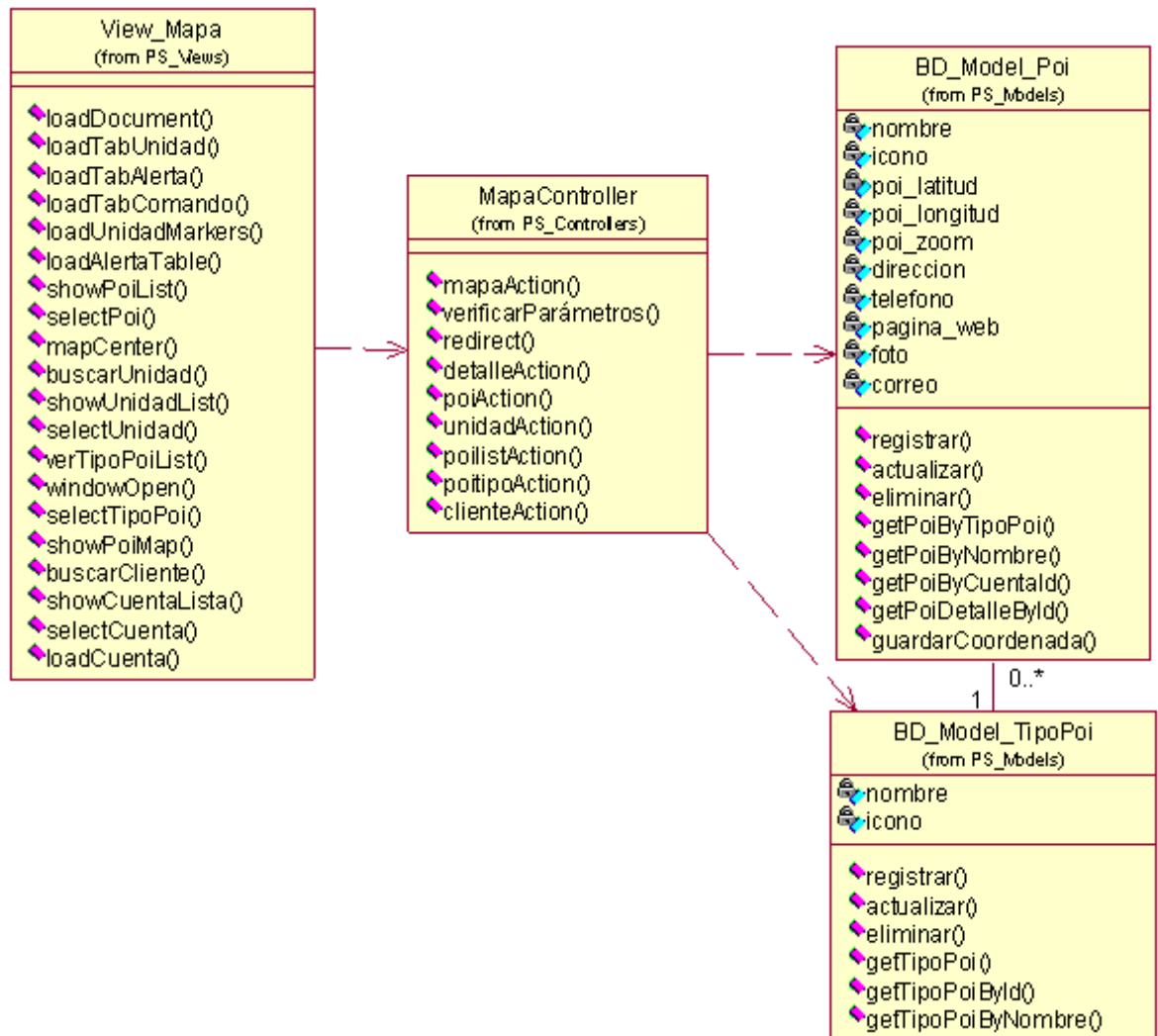


Figura 4.131 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Ver Punto de Interés

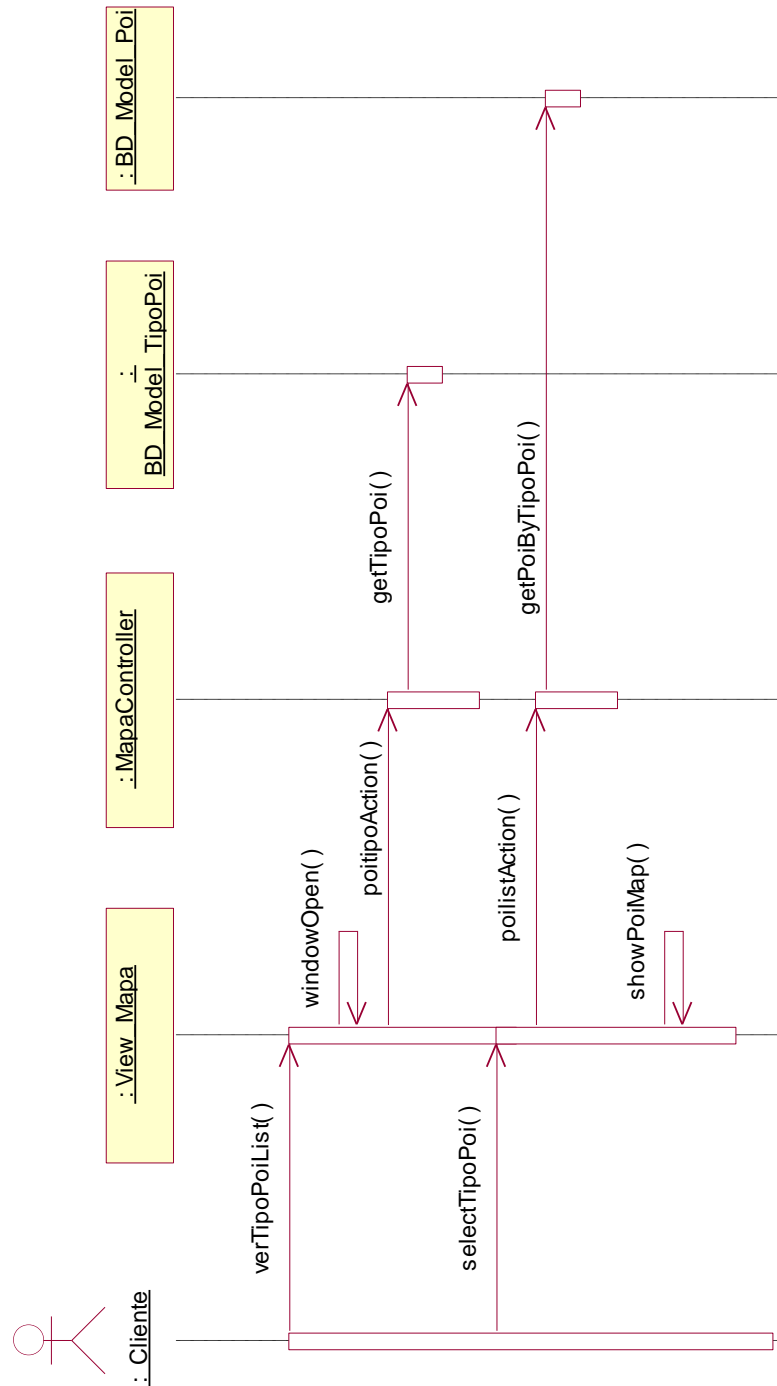


Figura 4.132 Diagrama de secuencia “Ver Punto de Interés”

**Descripción del diagrama de secuencia – Ver Punto de Interés:** El usuario cuando hace clic en el botón “Ver POI”, este evento ejecuta la función verTipoPoiList de la clase View\_Mapa, esta función abre una ventana a través de la

función windowOpen de la clase View\_Mapa que se encarga de listar los tipos de puntos de interés a fin de seleccionarlos, para esto, la función hace una llamada tipo ajax a la función poitipoAction de la clase MapaController que consulta a la clase BD\_Model\_TipoPoi a través de la función getTipoPoi devolviendo todos los tipos de puntos de interés del sistema, una vez que la función poitipoAction tiene los registros, ejecuta la descarga de un archivo tipo json a fin de que la función verTipoPoiList pueda listarlos en la venta de selección, una vez que el usuario seleccione los tipos de puntos de interés y haga clic en el botón “Ver Puntos de Interés”, se ejecuta la función selectTipoPoi de la clase ViewMapa, esta función a través de una llamada ajax solicita descargar un archivo tipo json de la clase MapaController a través de la función poilistAction, la función poilistAction llama a la función getPoiByTipoPoi de la clase BD\_Model\_Poi enviándole el identificador de la cuenta de usuario, esta función devuelve todos los puntos de interés de la cuenta que tenga los tipos de puntos de interés enviados como parámetros, una vez que la función poilistAction tiene los registros, ejecuta la descarga de un archivo tipo json que la función selectTipoPoi puede leer, posteriormente esta función envía como parámetros la información obtenida a la función showPoiMap, que se encarga de mostrar los puntos de interés en el mapa.

#### 4.3.5.12 Caso de Uso – Seleccionar Cliente en el Mapa

##### Diagrama de Clases

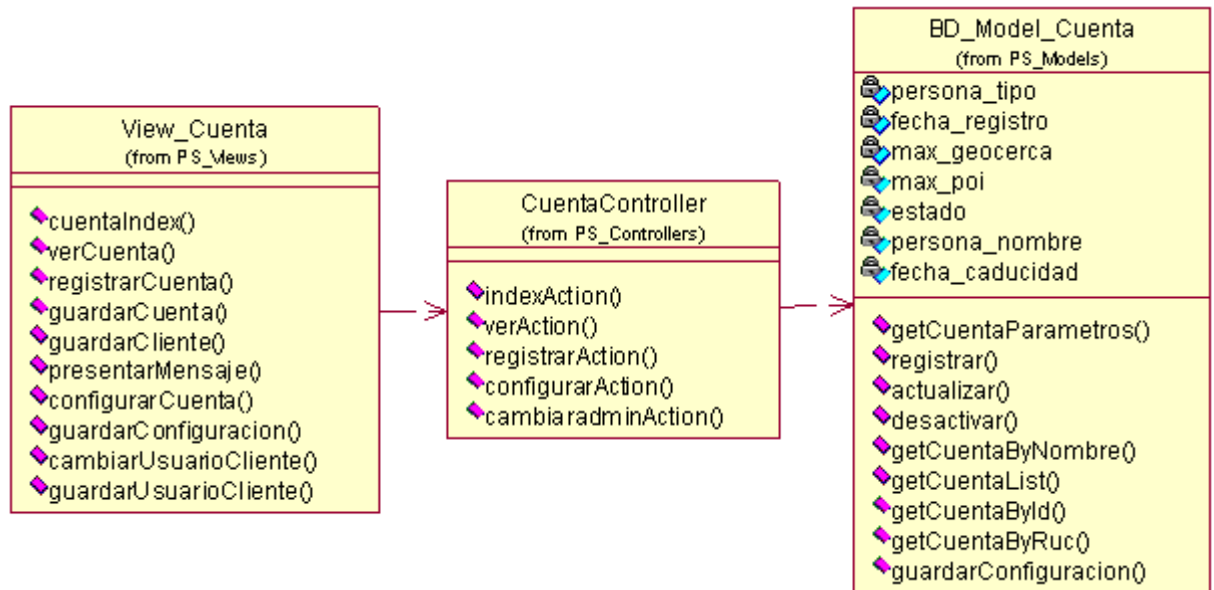


Figura 4.133 Diagrama de clases del caso de uso



## Diagrama de Secuencia – Seleccionar Cliente en el Mapa

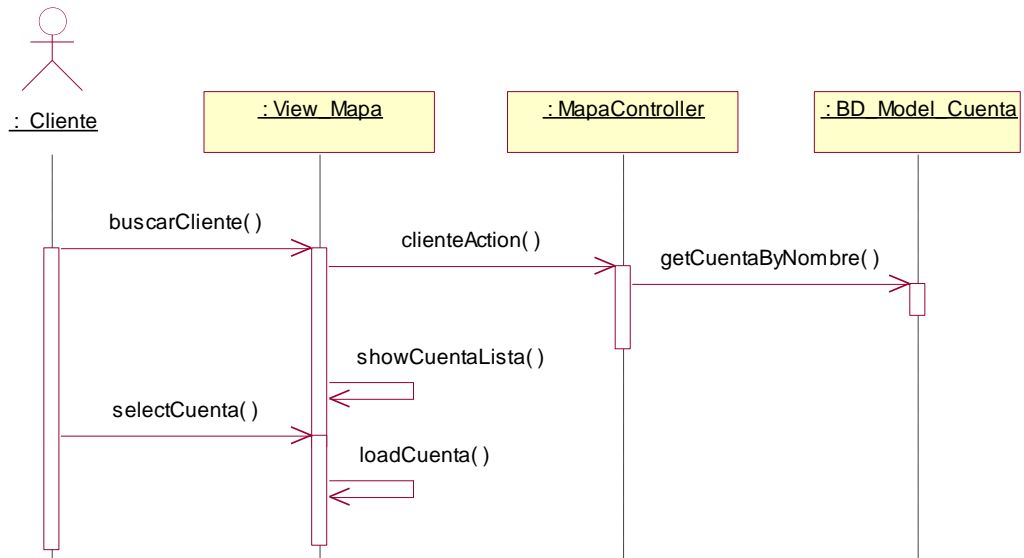


Figura 4.134 Diagrama de secuencia “Seleccionar Cliente en el Mapa”

**Descripción del diagrama de secuencia – Seleccionar Cliente en el Mapa:** El usuario en la caja de texto de cambio de cuentas clientes, escribe el nombre de una cuenta cliente, este evento ejecuta la función buscarCliente de la clase ViewMapa, esta función hace una llamada tipo ajax solicitando la descarga de un archivo tipo json a la función clienteAction de la clase MapaController que consulta a la función getCuentaByNombre de la clase BD\_Model\_Cuenta a fin de obtener todas las cuentas que coincidan con el texto ingresado, una vez que la función clienteAction tiene los registros solicitados, ejecuta la descarga del archivo tipo json, cuando la función buscarCliente tiene el archivo con los registros, ejecuta la función showCuentaLista que se encarga de mostrar en el formato CSS `<ul><li>contenido</li></ul>` la lista de cuentas clientes que coinciden con el texto ingresado, cuando el usuario selecciona un registro de la lista, se ejecuta la función selectCuenta de la clase View\_Mapas que se encarga de reiniciar todos los datos del mapa a través de la función loadCuenta, la cual recarga literalmente todo el contenido del mapa en función a la cuenta seleccionada.

### 4.3.6 Realización de casos de uso de diseño – Paquete Configuración

#### 4.3.6.1 Caso de Uso – Mantenimiento de Modelo de Equipos

##### Diagrama de Clases

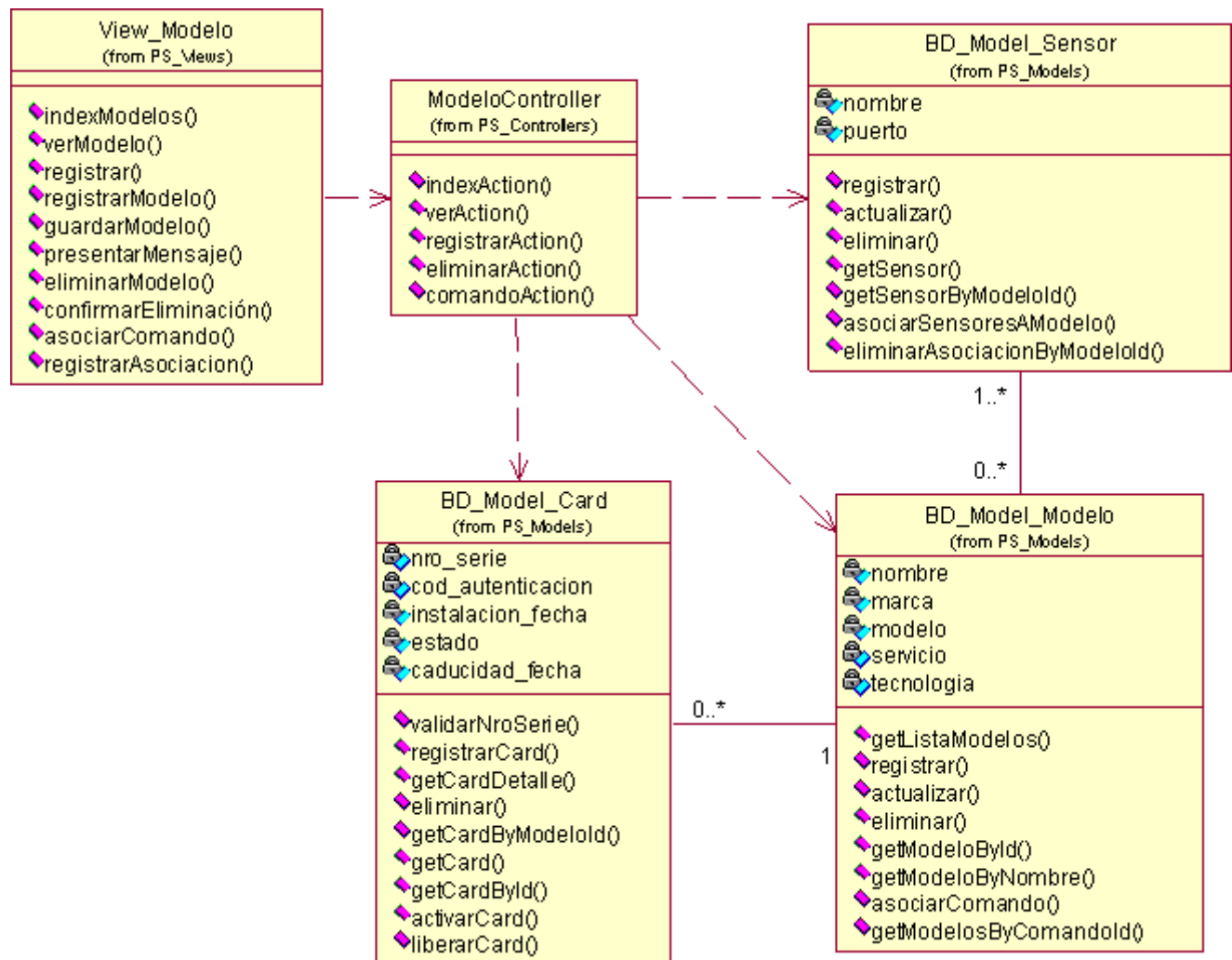


Figura 4.135 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

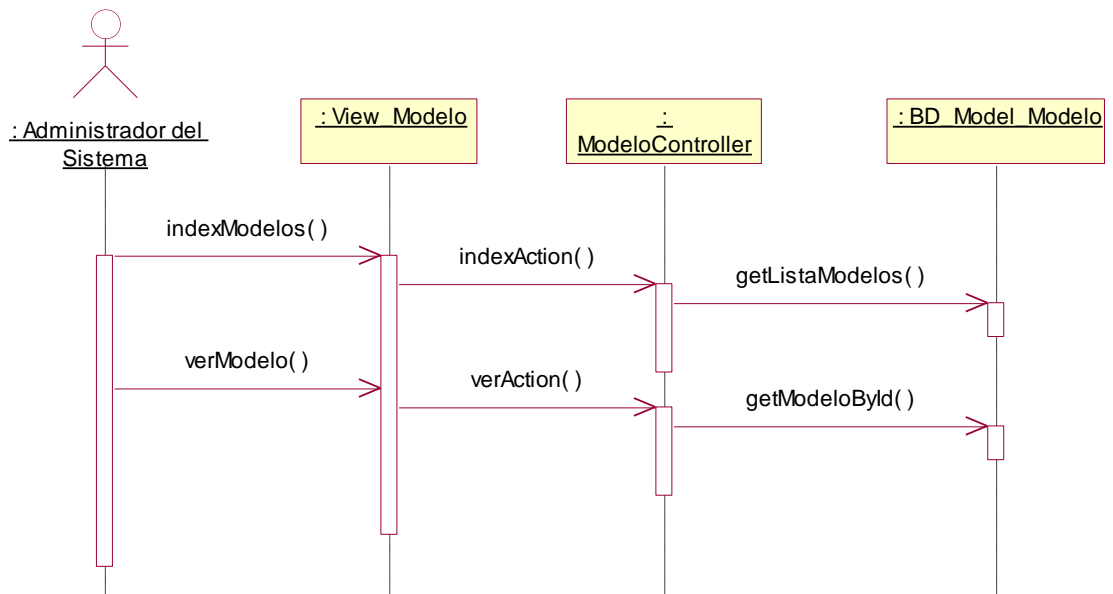
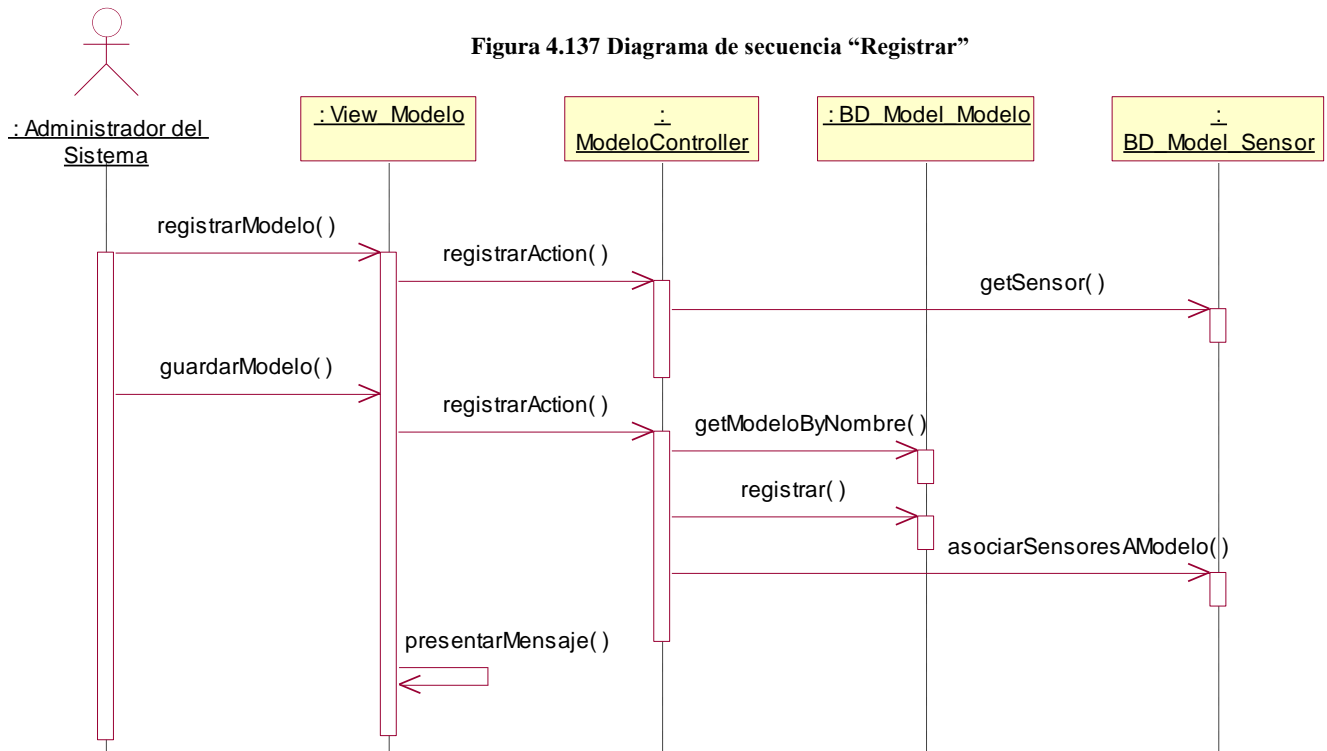


Figura 4.136 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Modelos” del menú de Configuración, invoca a la clase View\_Modelo ejecutando la función indexModelos, la función indexModelos hace una llamada a la función indexAction de la clase ModeloController que se encarga de obtener la lista de modelos registrados en el sistema a través de la función getListaModelos de la clase BD\_Model\_Modelo, una vez que la función indexAction posee los registros, los entrega a la función indexModelos de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verModelo de la clase View\_Modelo, el cual hace una llamada a la función verAction de la clase ModeloController, esta función realiza una llamada a la función getModeloById de la clase BD\_Model\_Modelo que devuelve el detalle del modelo por el id del modelo, una vez que la función posee el detalle del modelo, la información es presentada a través de la clase View\_Modelo.

## Diagrama de Secuencia – Registrar

Figura 4.137 Diagrama de secuencia “Registrar”



**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de modelos, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarModelo de la clase View\_Modelo, la función registrarModelo hace una llamada a la función registrarAction de la clase ModeloController el cual invoca a la clase BD\_Model\_Sensor a fin de tener la lista de sensores del sistema a través de la función getSensor. Una vez que la función registrarAction tiene los datos solicitados, la función registrarModelo puede presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarModelo de la clase View\_Modelo, la función hace una llamada a la función registrarAction de la clase ModeloController el cual trata de verificar que el nombre del modelo sea único a través de la función getModeloByNombre de la clase BD\_Model\_Modelo, si se cumple la condición se ejecuta la función registrar de la misma clase y asocia los sensores al modelo a través de la función asociarSensoresAModelo de la clase BD\_Model\_Sensor, posteriormente la clase View\_Modelo puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

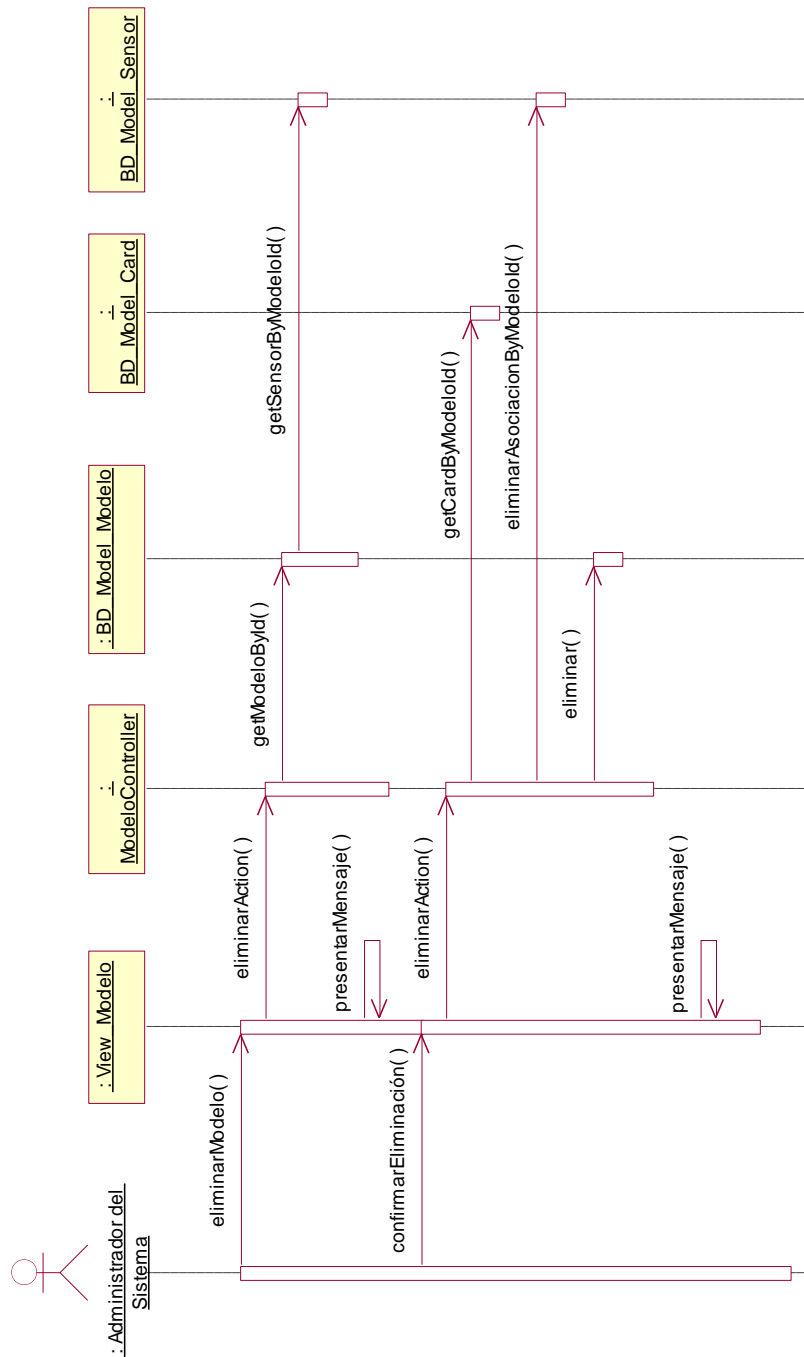


Figura 4.138 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de modelos, hace clic en la opción “Eliminar”, este evento ejecuta la opción `eliminarModelo` de la clase `View_Modelo`, la función `eliminarModelo` hace una llamada a la función `eliminarAction` de la clase `ModeloController` el cual invoca a la función `getModeloById` de la clase `BD_Model_Modelo` a fin de obtener los datos del modelo a eliminar y ejecuta la función `getSensorByModeloId` de la clase

BD\_Model\_Sensor a fin de tener la lista de sensores asociados al modelo, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarModelo y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Modelo el cual invoca nuevamente a la función eliminarAction de la clase ModeloController el cual primero verifica que el modelo no tenga equipos asociados a través de la función getCardByModeloId de la clase BD\_Model\_Card, una vez que la función eliminarAction confirma que ese modelo no tiene registros asociados, ejecuta la función eliminarAsociacionByModeloId de la clase BD\_Model\_Sensor y ejecuta la función eliminar de la clase BD\_Model\_Modelo, posteriormente la clase View\_Modelo a través de la función presentarMensaje indica que la operación tuvo éxito.

#### 4.3.6.2 Caso de Uso – Asociar Comando a Modelo de Equipo

##### Diagrama de Clases

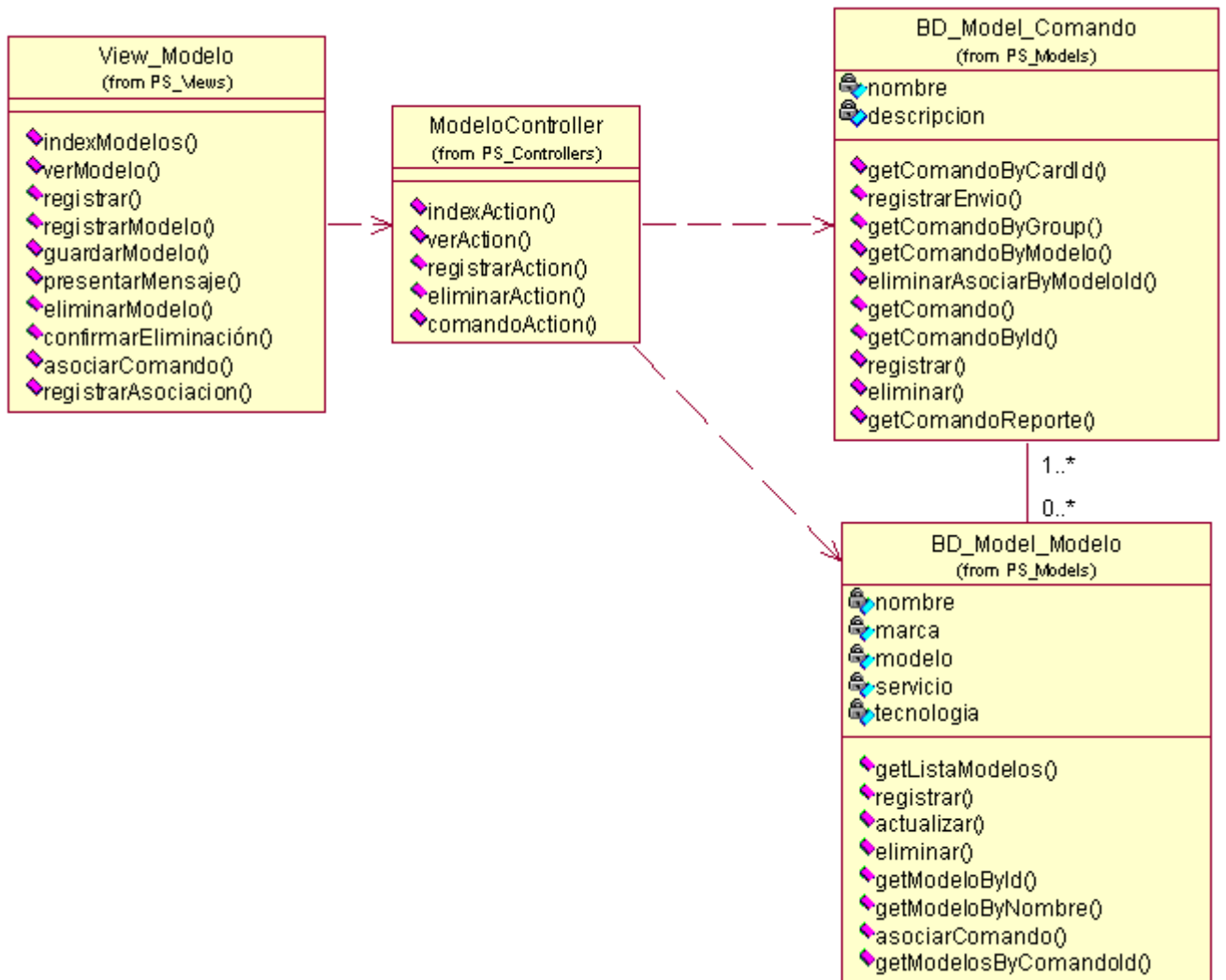


Figura 4.139 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Asociar

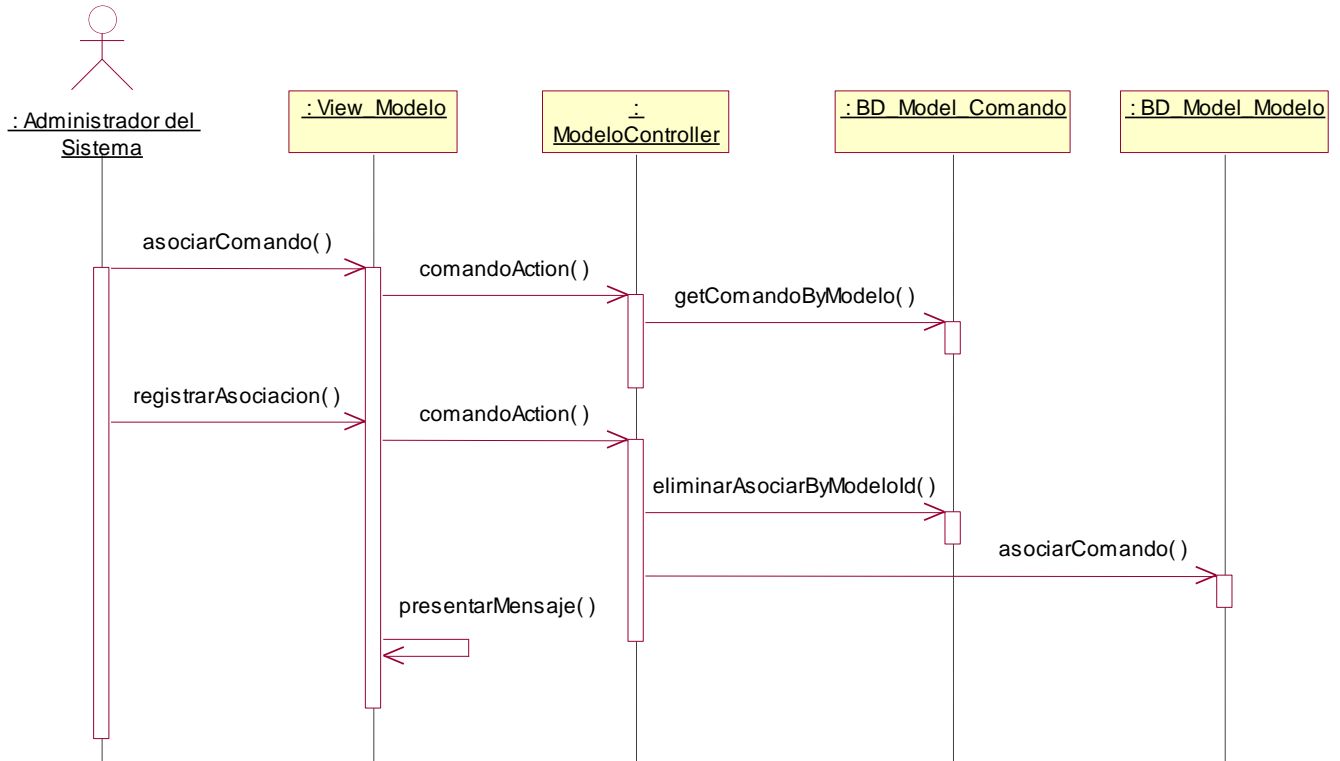


Figura 4.140 Diagrama de secuencia “Asociar”

**Descripción del diagrama de secuencia – Asociar:** El usuario al seleccionar la opción “comandos” de la lista de modelos de la clase View\_Modelo, ejecuta la función asociarComando, esta función llama a la función comandoAction de la clase ModeloController, que se encarga de obtener todos los comandos asociados al modelo a través de la función getComandoByModelo de la clase BD\_Model\_Comando, una vez que la función comandoAction posee la información, presenta la información a través de la función asociarComando de la clase View\_Modelo. El usuario selecciona los comandos y hace clic en el botón “Actualizar” el cual ejecuta la función registrarAsociación de la clase View\_Modelo, esta función llama a la función comandoAction de la clase ComandoController, el cual primero se encarga de eliminar las asociaciones previas a través de la función eliminarAsociarByModeloId de la clase BD\_Model\_Comando y ejecuta la función asociarComando de la clase BD\_Model\_Modelo, una vez que la asociación ha sido registrada, la función comandoAction, solicita a la clase View\_Modelo presentar un mensaje el cual lo hace a través de la función presentarMensaje indicando que la operación tuvo éxito.

### 4.3.6.3 Caso de Uso – Mantenimiento de Comandos

#### Diagrama de Clases

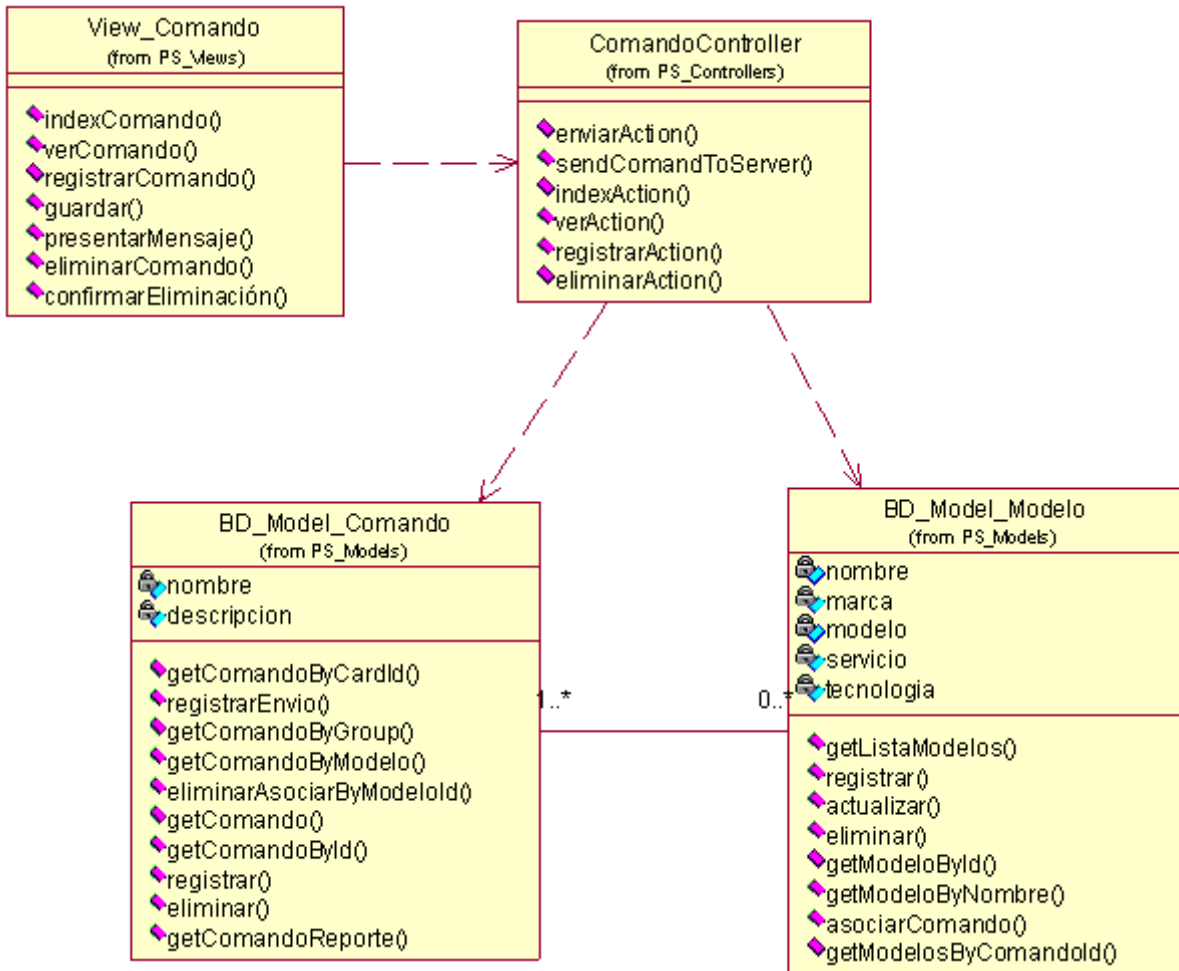


Figura 4.141 Diagrama de clases del caso de uso



## Diagrama de Secuencia –Listar

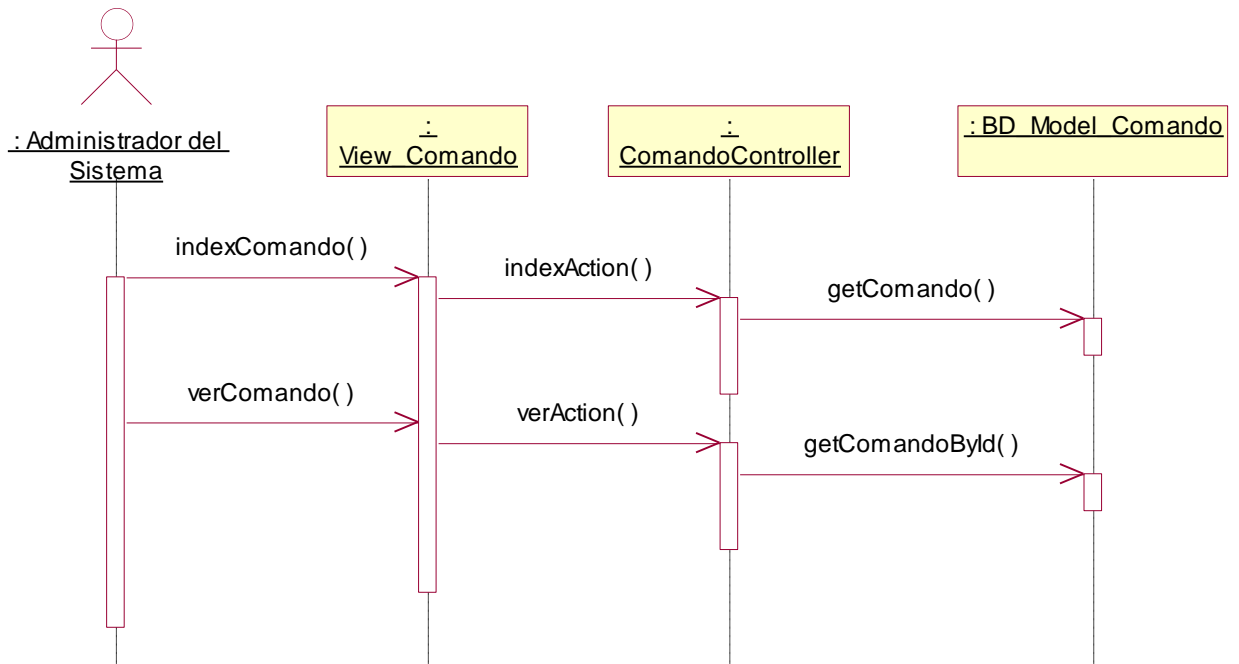


Figura 4.142 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Comandos” del menú de Configuración, invoca a la clase View\_Comando ejecutando la función indexComando, la función indexComando hace una llamada a la función indexAction de la clase ComandoController que se encarga de obtener la lista de comandos registrados en el sistema a través de la función getComando de la clase BD\_Model\_Comando, una vez que la función indexAction posee los registros, los entrega a la función indexComando de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verComando de la clase View\_Comando, el cual hace una llamada a la función verAction de la clase ComandoController, esta función realiza una llamada a la función getComandoById de la clase BD\_Model\_Comando que devuelve el detalle del comando por el id del comando, una vez que la función posee el detalle del comando, la información es presentada a través de la clase View\_Comando.

## Diagrama de Secuencia – Registrar

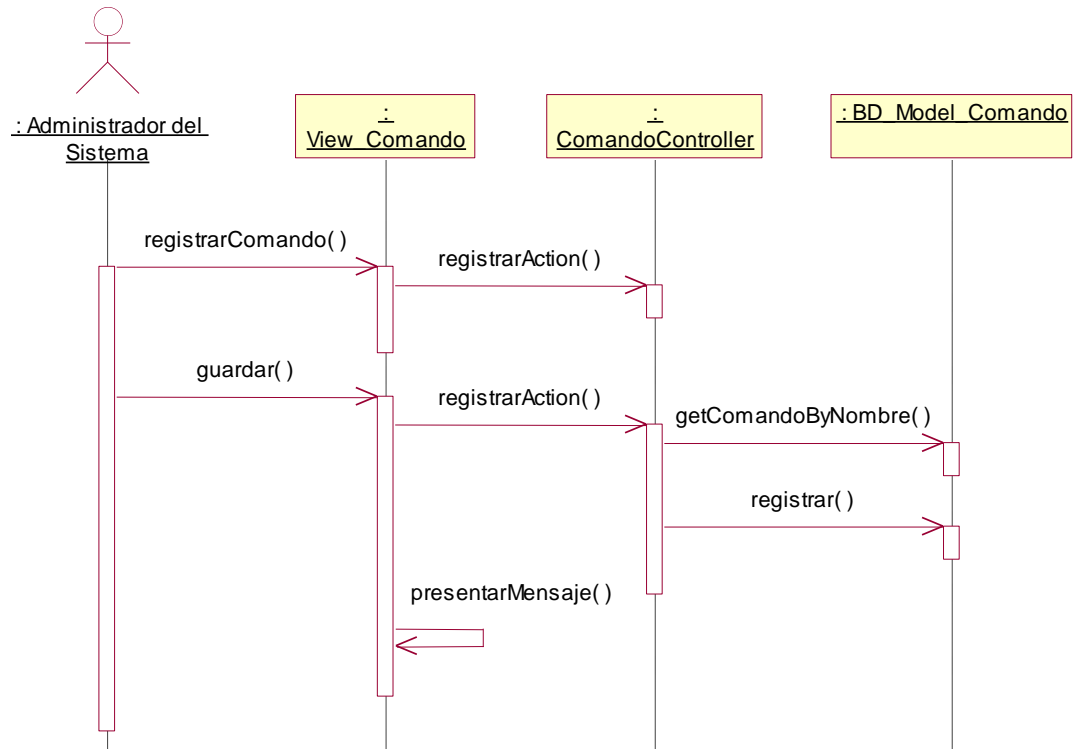


Figura 4.143 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de comandos, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarComando de la clase View\_Comando, la función registrarComando hace una llamada a la función registrarAction de la clase ComandoController la cual invita a la función registrarModelo presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_Comando, la función hace una llamada a la función registrarAction de la clase ComandoController el cual trata de verificar que el nombre del comando sea único a través de la función getComandoByNombre de la clase BD\_Model\_Comando, si se cumple la condición se ejecuta la función registrar de la misma clase, posteriormente la clase View\_Comando puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

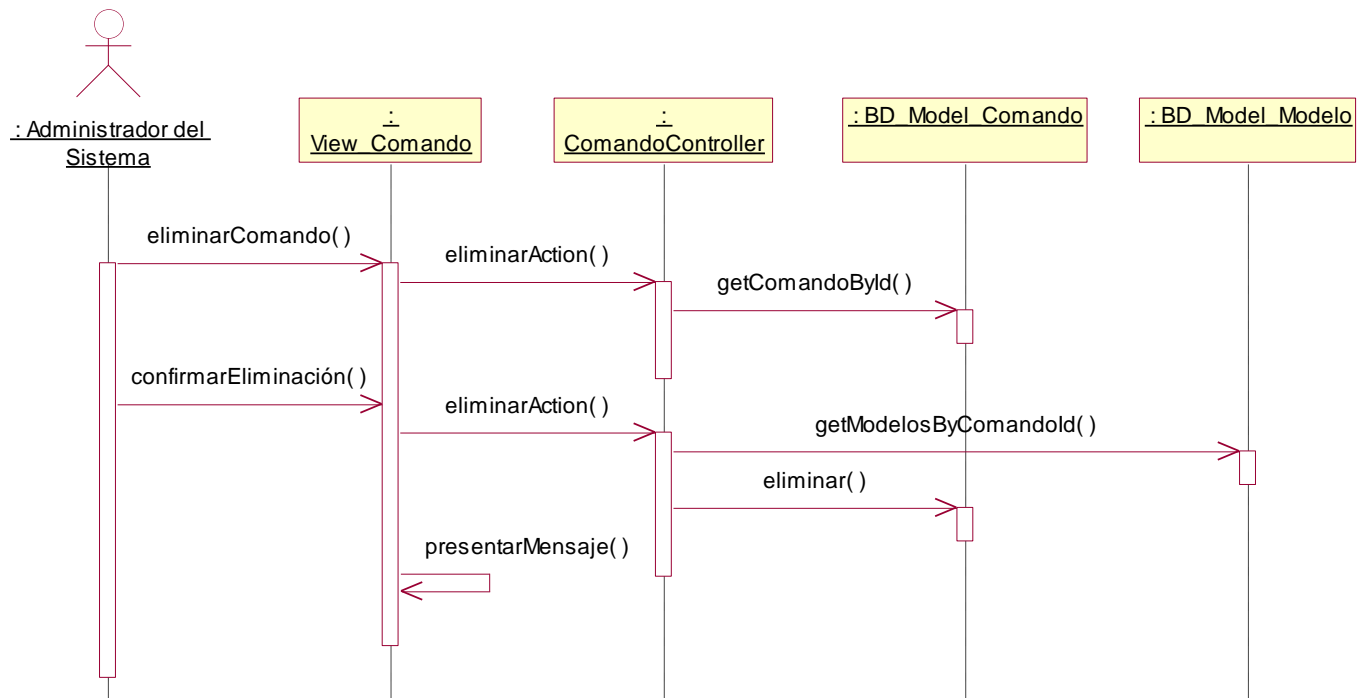


Figura 4.144 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de comandos, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarComando de la clase View\_Comando, la función eliminarComando hace una llamada a la función eliminarAction de la clase ComandoController el cual invoca a la función getComandoById de la clase BD\_Model\_Comando a fin de obtener los datos del comando a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarComando y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Comando el cual invoca nuevamente a la función eliminarAction de la clase ComandoController el cual primero verifica que el comando no tenga modelos de equipos asociados a través de la función getModelosByComandoId de la clase BD\_Model\_Modelo, una vez que la función eliminarAction confirma que ese comando no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Comando, posteriormente la clase View\_Comando a través de la función presentarMensaje indica que la operación tuvo éxito.

#### 4.3.6.4 Caso de Uso – Mantenimiento de Eventos

##### Diagrama de Clases

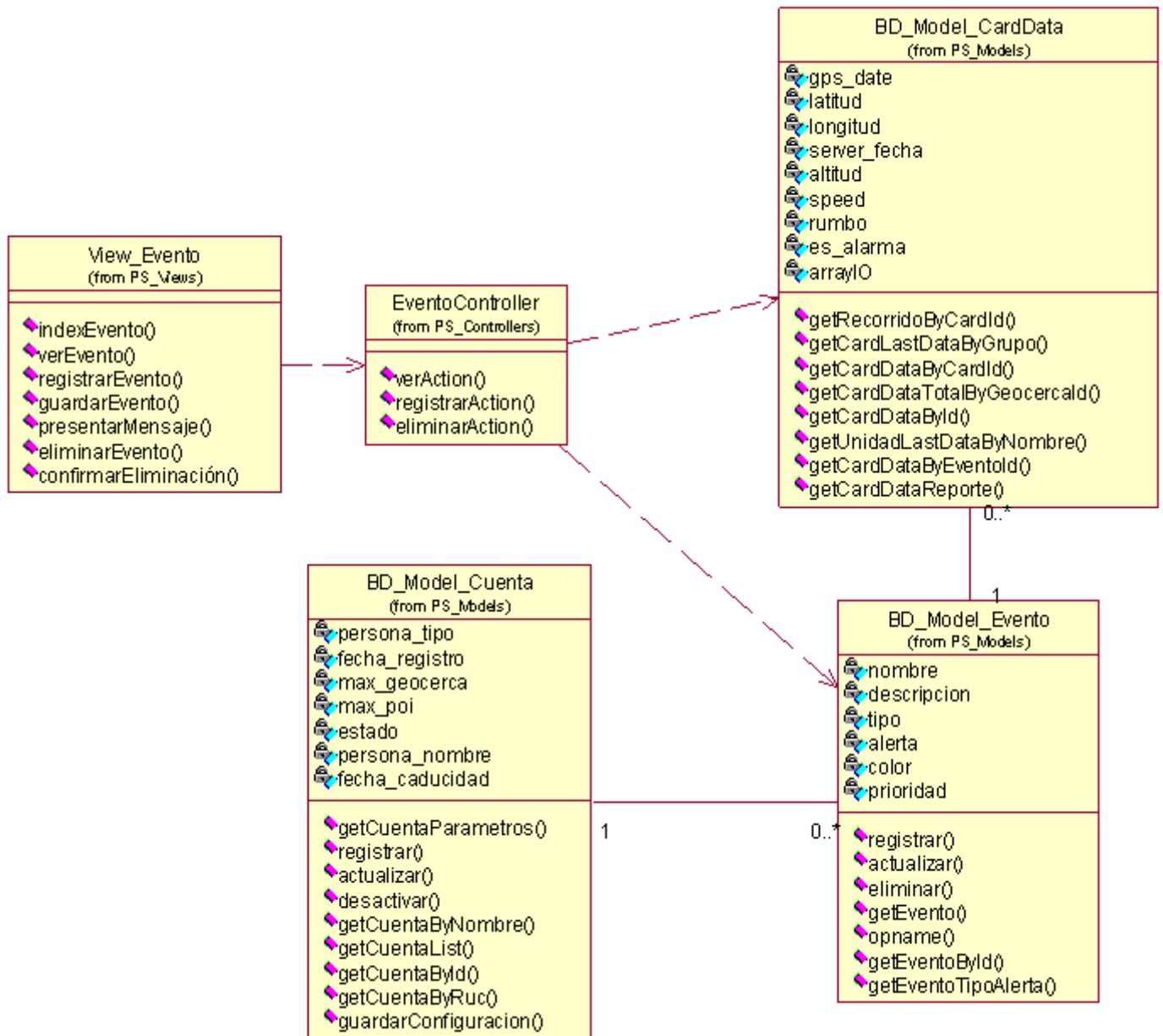


Figura 4.145 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

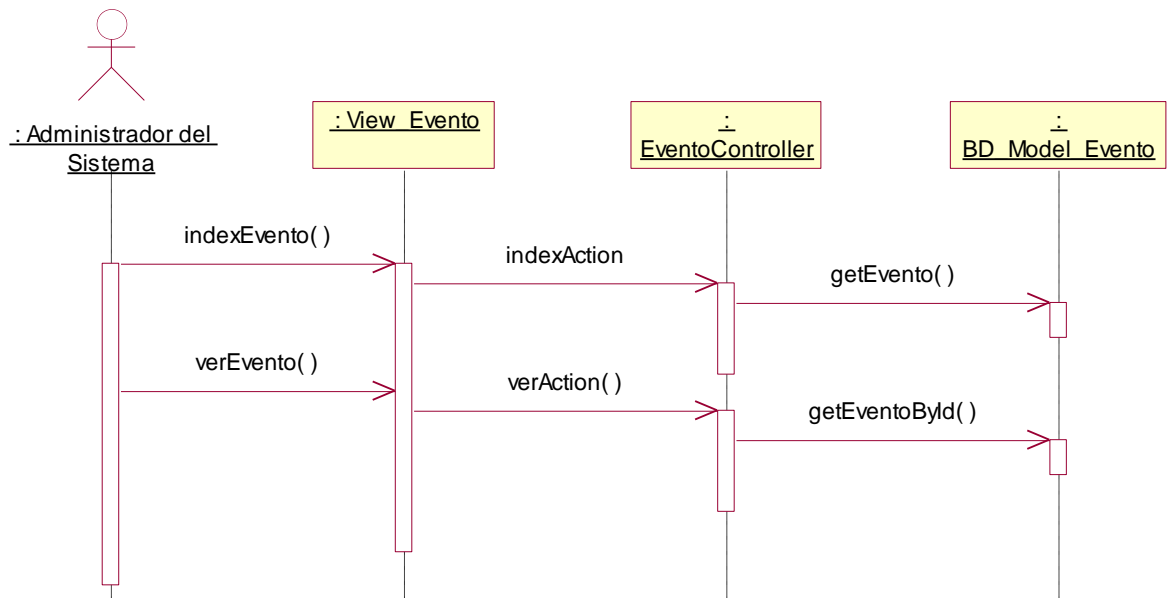


Figura 4.146 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Eventos” del menú de Configuración, invoca a la clase View\_Evento ejecutando la función indexEvento, la función indexEvento hace una llamada a la función indexAction de la clase EventoController que se encarga de obtener la lista de eventos registrados en el sistema a través de la función getEvento de la clase BD\_Model\_Evento, una vez que la función indexAction posee los registros, los entrega a la función indexEvento de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verEvento de la clase View\_Evento, el cual hace una llamada a la función verAction de la clase EventoController, esta función realiza una llamada a la función getEventoById de la clase BD\_Model\_Evento que devuelve el detalle del evento por el id del evento, una vez que la función posee el detalle del evento, la información es presentada a través de la clase View\_Evento.

## Diagrama de Secuencia – Registrar

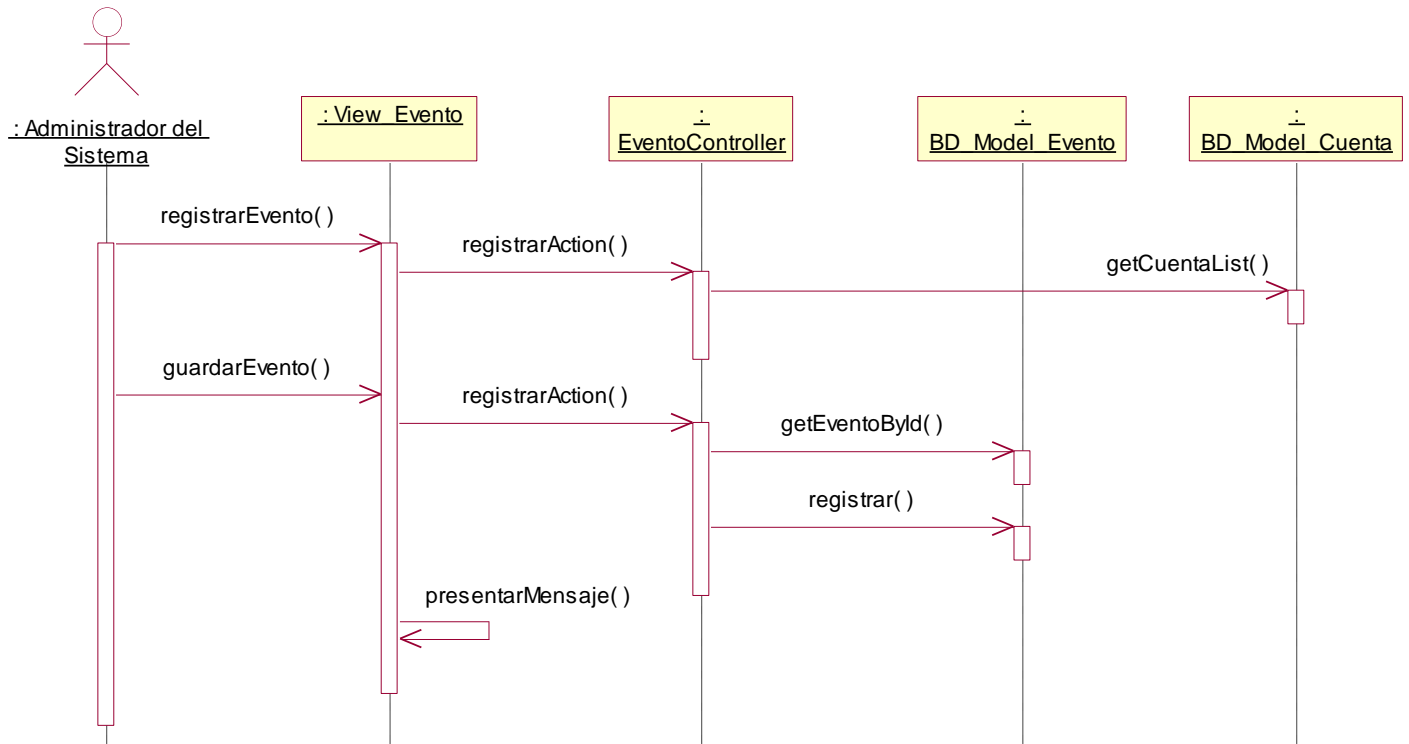


Figura 4.147 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de eventos, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarEvento de la clase View\_Evento, la función registrarEvento hace una llamada a la función registrarAction de la clase EventoController que solicita la lista de cuentas clientes a la clase BD\_Model\_Cuenta a través de la función getCuentaList, posteriormente invita a la función registrarEvento presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_Evento, la función hace una llamada a la función registrarAction de la clase EventoController el cual trata de verificar que el código del evento sea único a través de la función getEventoById de la clase BD\_Model\_Evento, si se cumple la condición se ejecuta la función registrar de la misma clase, posteriormente la clase View\_Evento puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

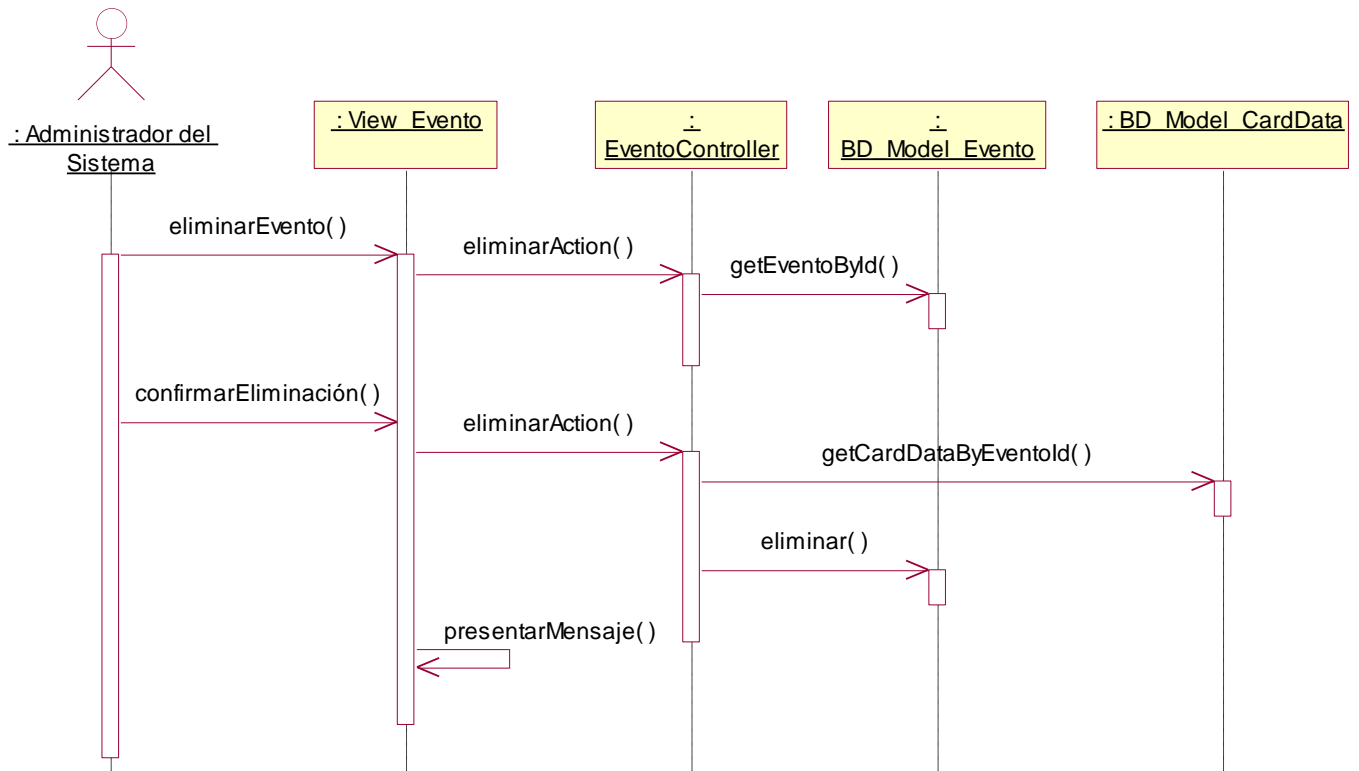


Figura 4.148 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de eventos, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarEvento de la clase View\_Evento, la función eliminarEvento hace una llamada a la función eliminarAction de la clase EventoController el cual invoca a la función getEventoById de la clase BD\_Model\_Evento a fin de obtener los datos del evento a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarEvento y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Evento el cual invoca nuevamente a la función eliminarAction de la clase EventoController el cual primero verifica que el evento no tenga tramas asociadas a través de la función getCardDataByEventoId de la clase BD\_Model\_CardData, una vez que la función eliminarAction confirma que ese evento no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Evento, posteriormente la clase View\_Evento a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.6.5 Caso de Uso – Mantenimiento de Tipos de Puntos de Interés

#### Diagrama de Clases

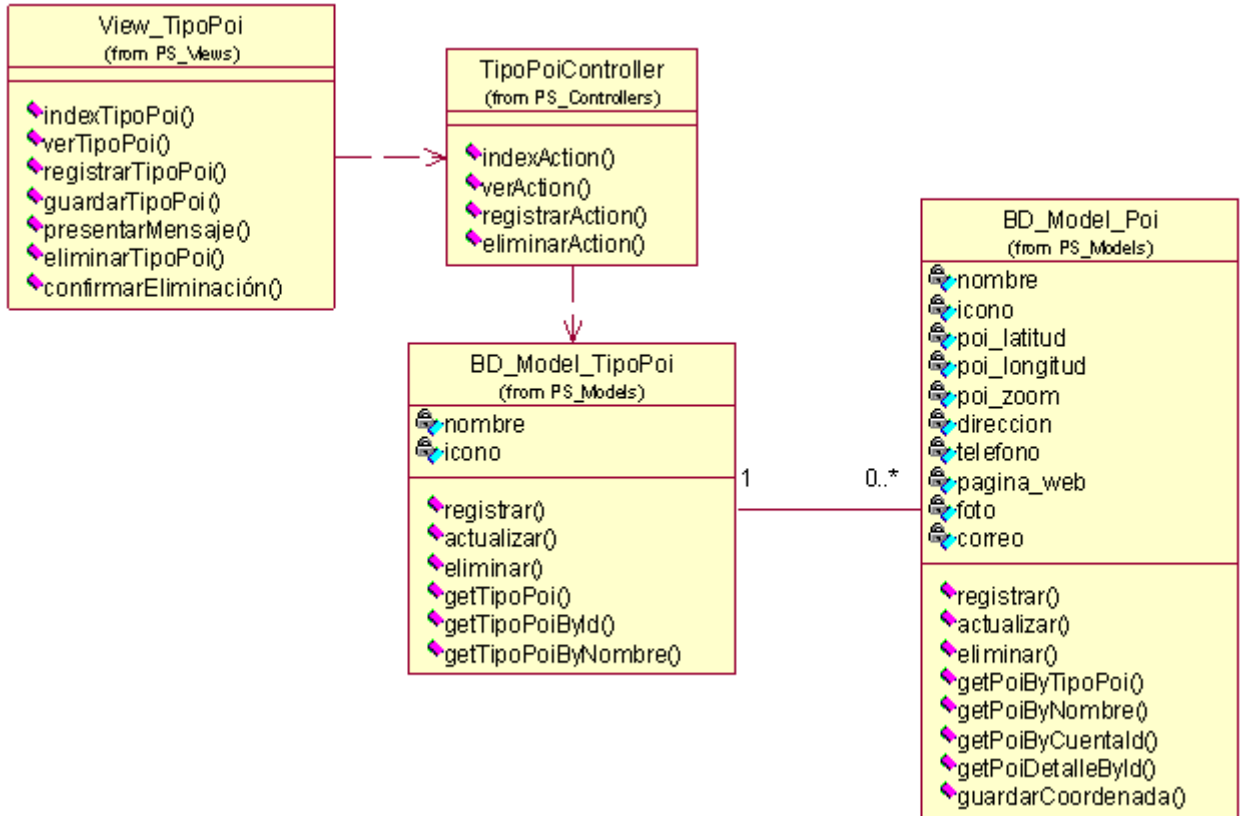


Figura 4.149 Diagrama de clases del caso de uso



## Diagrama de Secuencia –Listar

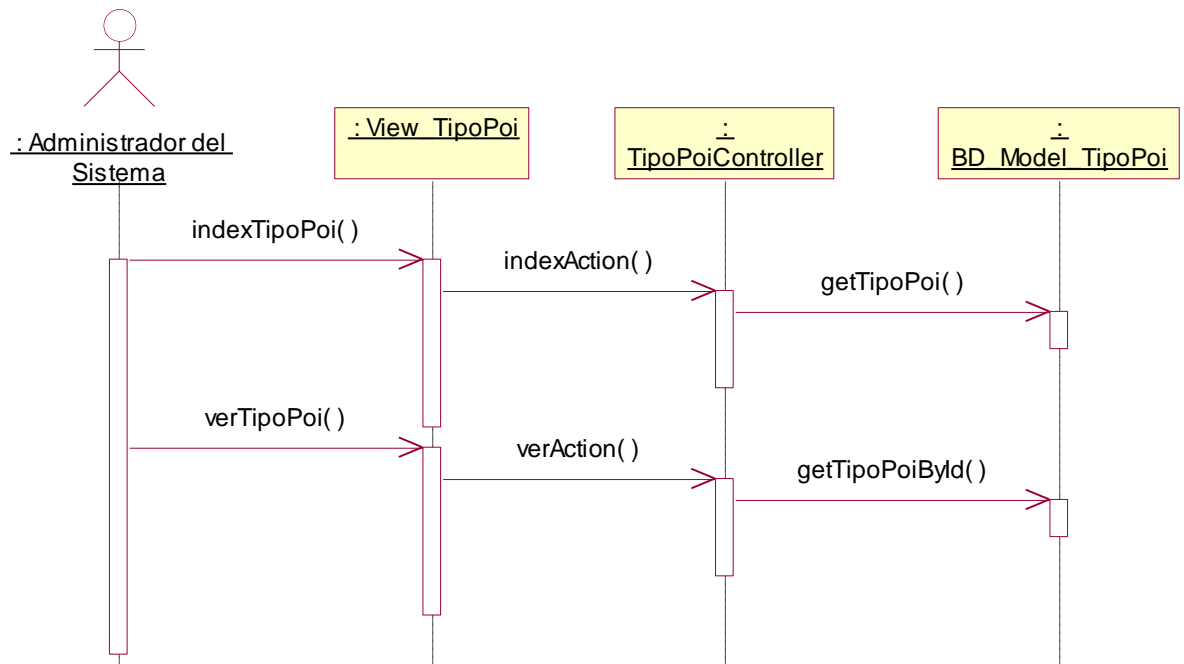


Figura 4.150 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Tipo Poi” del menú de Configuración, invoca a la clase View\_TipoPoi ejecutando la función indexTipoPoi, la función indexTipoPoi hace una llamada a la función indexAction de la clase TipoPoiController que se encarga de obtener la lista de tipos de puntos de interés registrados en el sistema a través de la función getTipoPoi de la clase BD\_Model\_TipoPoi, una vez que la función indexAction posee los registros, los entrega a la función indexTipoPoi de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verTipoPoi de la clase View\_TipoPoi, el cual hace una llamada a la función verAction de la clase TipoPoiController, esta función realiza una llamada a la función getTipoPoiById de la clase BD\_Model\_TipoPoi que devuelve el detalle del tipo de punto de interés por el id del tipo del punto de interés, una vez que la función posee el detalle del tipo de punto de interés, la información es presentada a través de la clase View\_TipoPoi.

## Diagrama de Secuencia – Registrar

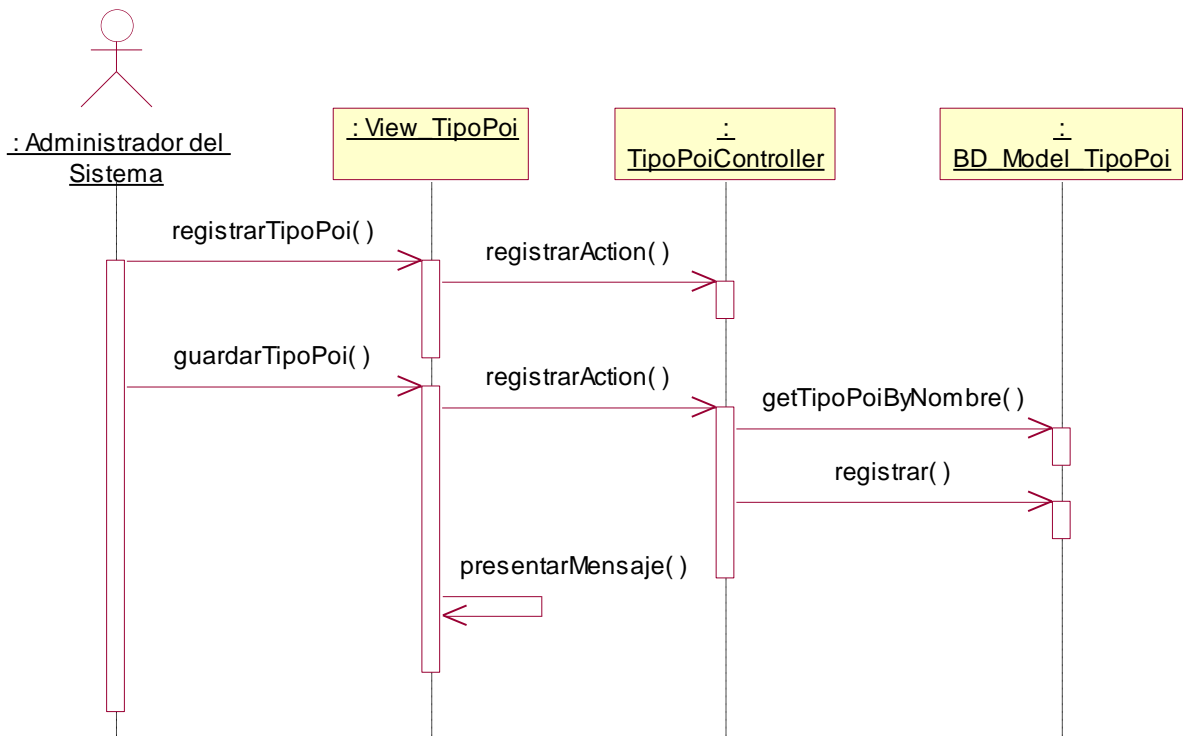


Figura 4.151 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de tipos de puntos de interés, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarTipoPoi de la clase View\_TipoPoi, la función registrarTipoPoi hace una llamada a la función registrarAction de la clase TipoPoiController que invita a la función registrarEvento presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_TipoPoi, la función hace una llamada a la función registrarAction de la clase TipoPoiController el cual trata de verificar que el nombre del tipo de punto de interés sea único a través de la función getTipoPoiByNombre de la clase BD\_Model\_TipoPoi, si se cumple la condición se ejecuta la función registrar de la misma clase, posteriormente la clase View\_TipoPoi puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

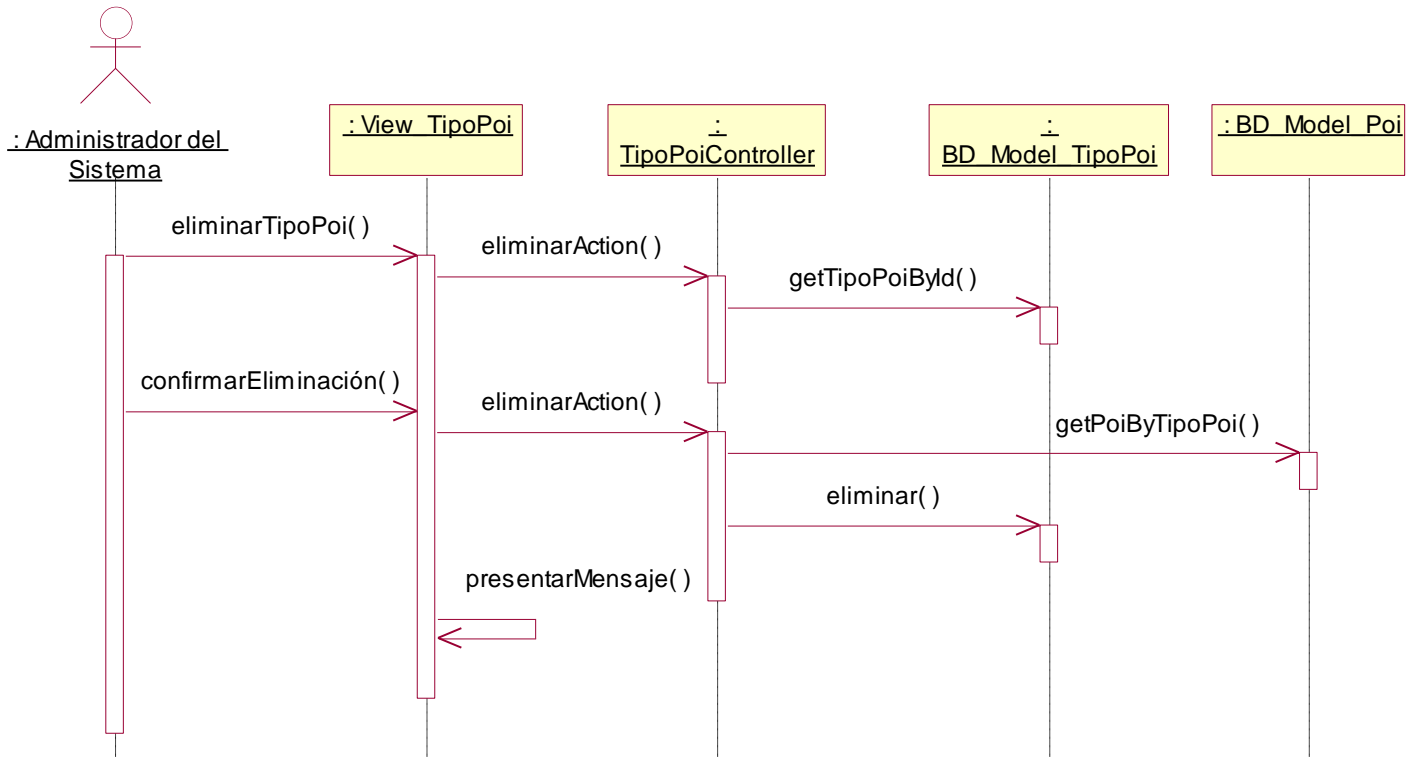


Figura 4.152 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de tipos de puntos de interés, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarTipoPoi de la clase View\_TipoPoi, la función eliminarTipoPoi hace una llamada a la función eliminarAction de la clase TipoPoiController el cual invoca a la función getTipoPoiById de la clase BD\_Model\_TipoPoi a fin de obtener los datos del tipo de punto de interés a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarTipoPoi y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_TipoPoi el cual invoca nuevamente a la función eliminarAction de la clase TipoPoiController el cual primero verifica que el tipo de punto de interés no tenga puntos de interés asociados a través de la función getPoiByTipoPoi de la clase BD\_Model\_Poi, una vez que la función eliminarAction confirma que ese tipo de punto de interés no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_TipoPoi, posteriormente la clase View\_TipoPoi a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.7 Realización de casos de uso de diseño – Paquete Administración

#### 4.3.7.1 Caso de Uso – Mantenimiento de Clientes

#### Diagrama de Clases

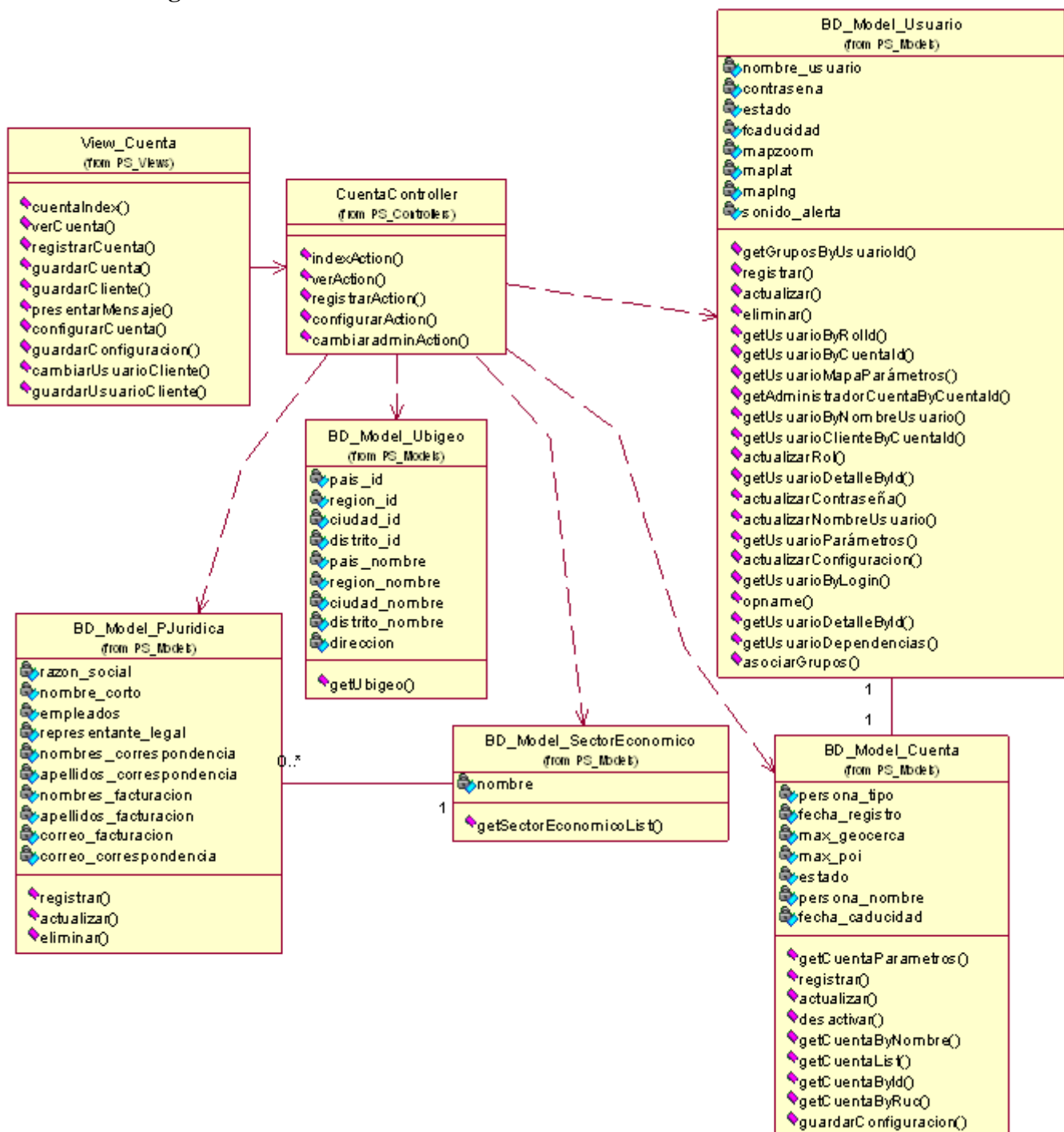


Figura 4.153 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

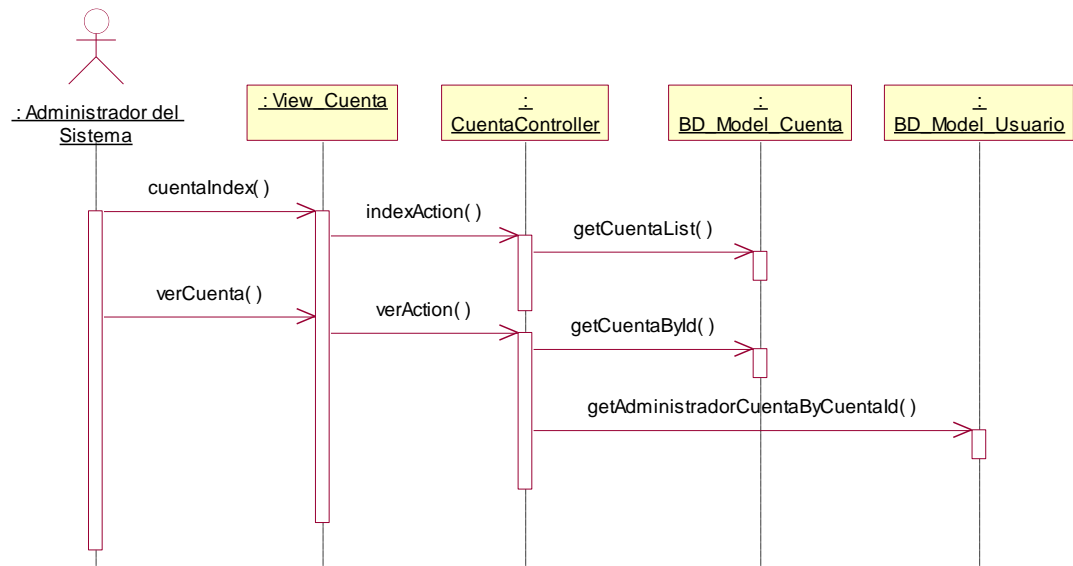


Figura 4.154 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Clientes” del menú de Administración, invoca a la clase View\_Cuenta ejecutando la función cuentaIndex, la función cuentaIndex hace una llamada a la función indexAction de la clase CuentaController que se encarga de obtener la lista de cuentas clientes registradas en el sistema a través de la función getCuentaList de la clase BD\_Model\_Cuenta, una vez que la función indexAction posee los registros, los entrega a la función cuentaIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verCuenta de la clase View\_Cuenta, el cual hace una llamada a la función verAction de la clase CuentaController, esta función realiza una llamada a la función getCuentaById de la clase BD\_Model\_Cuenta que devuelve el detalle de la cuenta cliente por el id de la cuenta y llama a la función getAdministradorCuentaByCuentaId de la clase BD\_Model\_Usuario a fin de tener los datos del administrador de la cuenta cliente, una vez que la función posee el detalle total de la cuenta, la información es presentada a través de la clase View\_Cuenta.

## Diagrama de Secuencia – Registrar

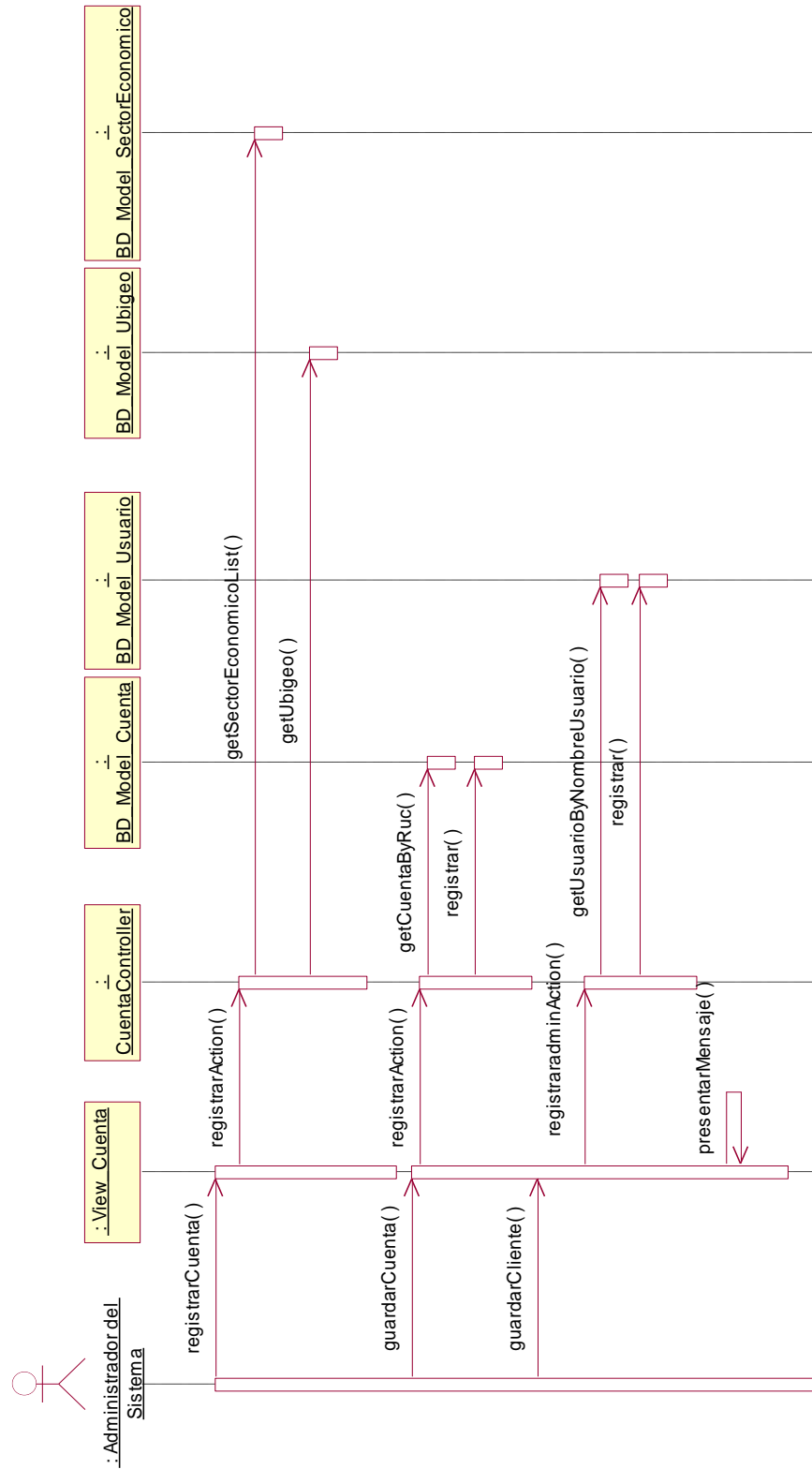


Figura 4.155 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de cuentas, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarCuenta de la clase View\_Cuenta, la función registrarCuenta hace una llamada a la función registrarAction de la clase CuentaController el cual invoca a la clase BD\_Model\_SectorEconomico a fin de tener la lista de sectores económicos a través de la función getSectorEconomicoList, posteriormente ejecuta la función getUbigeo de la clase BD\_Model\_Ubigeo a fin de tener la lista de departamentos, ciudades y distritos. Una vez que la función registrarAction tiene los datos solicitados, la función registrarCuenta puede presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarCuenta de la clase View\_Cuenta, la función hace una llamada a la función registrarAction de la clase CuentaController el cual trata de verificar que el ruc de la cuenta sea único a través de la función getCuentaByRuc de la clase BD\_Model\_Cuenta, si se cumple la condición se ejecuta la función registrar de la misma clase, posteriormente se presenta el formulario de registro de administrador de cuenta cliente al usuario, el usuario ingresa los datos del administrador de la cuenta y hace clic en el botón “Guardar”, este evento ejecuta la función guardarCliente de la clase View\_Cuenta, esta función hace una llamada a la función registraradminAction de la clase CuentaController que primero verifica que el usuario tenga un nombre de usuario único a través de la función getUsuarioByNombreUsuario de la clase BD\_Model\_Usuario, de cumplirse la condición se ejecuta la función registrar de la misma clase, posteriormente la clase View\_Cuenta puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Configurar

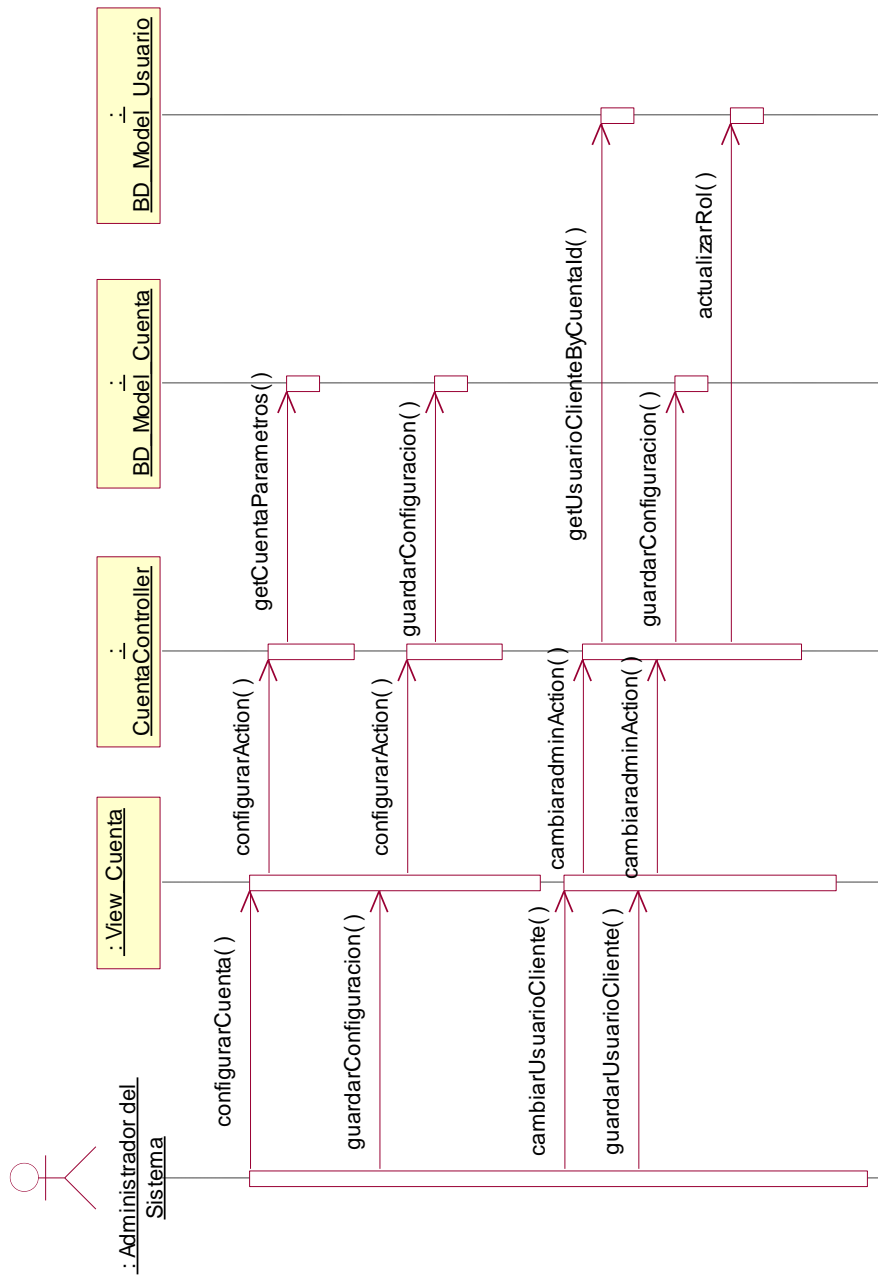


Figura 4.156 Diagrama de secuencia “Configurar”



**Descripción del diagrama de secuencia – Configurar:** El usuario en la lista de cuentas clientes, hace clic en la opción “Configurar”, este evento ejecuta la función configurarCuenta de la clase View\_Cuenta, la función hace una llamada a la función configurarAction de la clase CuentaController la cual obtiene todos los parámetros de la cuenta cliente a través de la función getCuentaParametros de la clase BD\_Model\_Cuenta, una vez que la función configurarAction tiene los datos listos, invita a la función configurarCuenta presentar los datos en un formulario, el usuario actualiza los datos y hace clic en el botón “Actualizar”, este evento ejecuta la función guardarConfiguración de la clase View\_Cuenta que hace una llamada nuevamente a la función configurarAction de la clase CuentaController la cual actualiza los datos de la cuenta a través de la función guardarConfiguración de la clase BD\_Model\_Cuenta. Posteriormente si el usuario hace clic en el enlace cambiarUsuarioCliente de la clase View\_Cuenta, este evento ejecuta la función cambiaradminAction de la clase CuentaController la cual obtiene todos los datos del usuario administrador de la cuenta cliente a través de la función getUsuarioClienteByCuentaId de la clase BD\_Model\_Usuario una vez que la función cambiaradminAction tiene los datos invita a la función cambiarUsuarioCliente presentar un formulario de actualización con los datos cargados, cuando el usuario actualiza los datos y hace clic en la función guardarUsuarioCliente de la clase View\_Cuenta, ejecuta la función cambiaradminAction de la clase CuentaController que ejecuta la función guardarConfiguracion de la clase BD\_Model\_Cuenta y ejecuta la función actualizarRol de la clase BD\_Model\_Usuario a fin de actualizar el rol del usuario a administrador de la cuenta cliente.

### 4.3.7.2 Caso de Uso – Mantenimiento de Equipos de Localización

#### Diagrama de Clases

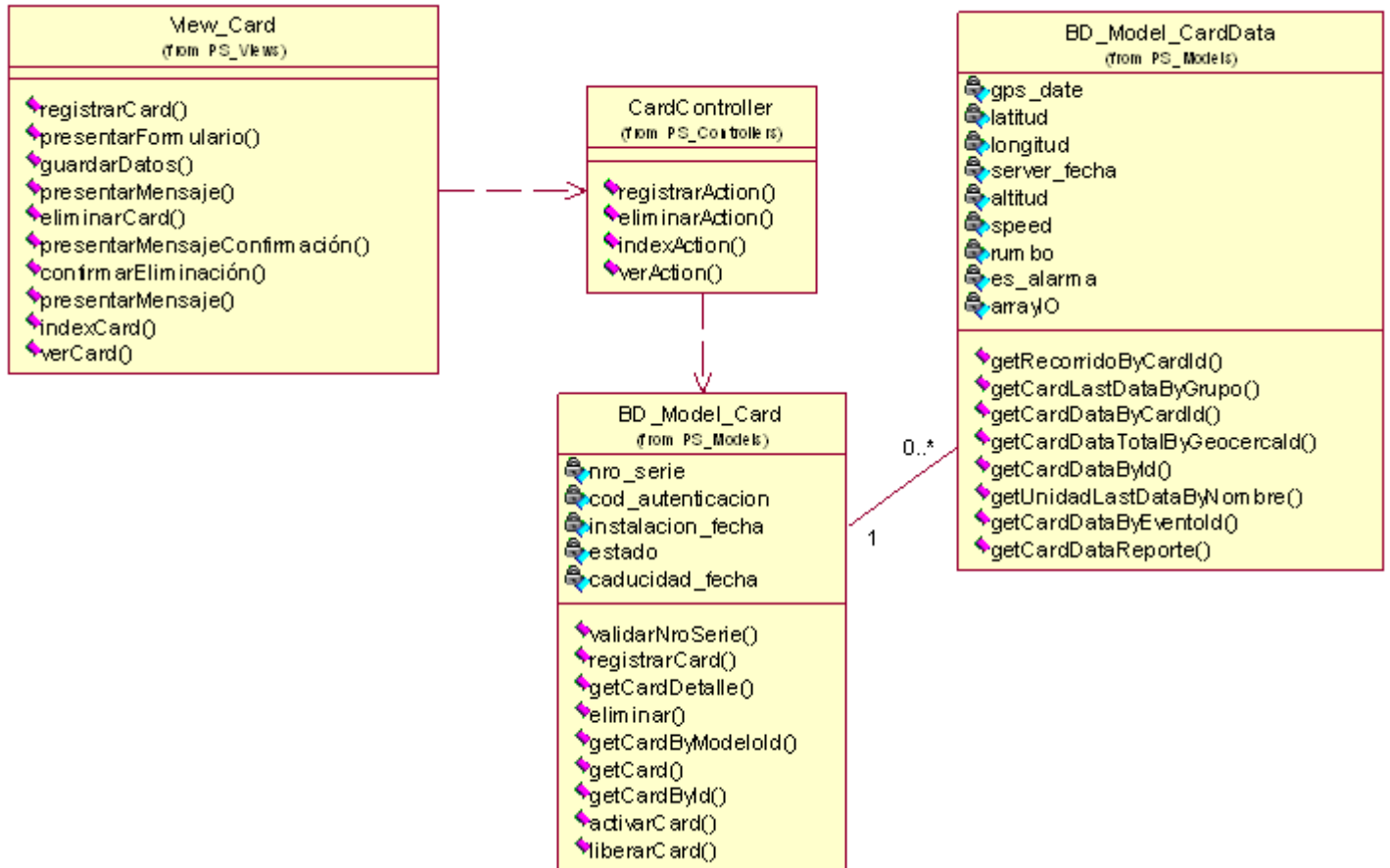


Figura 4.157 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

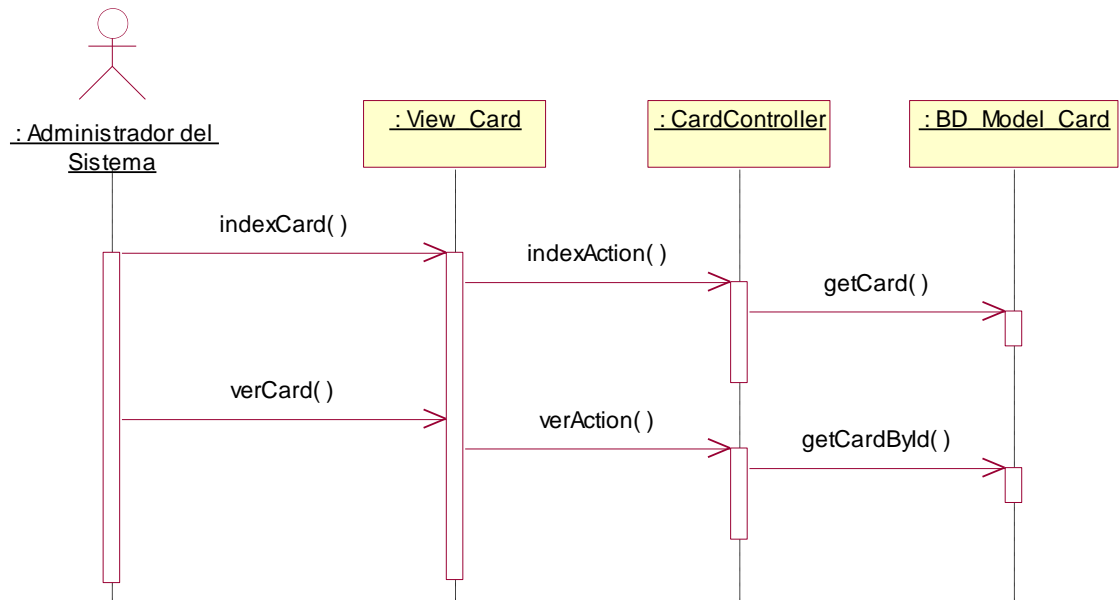
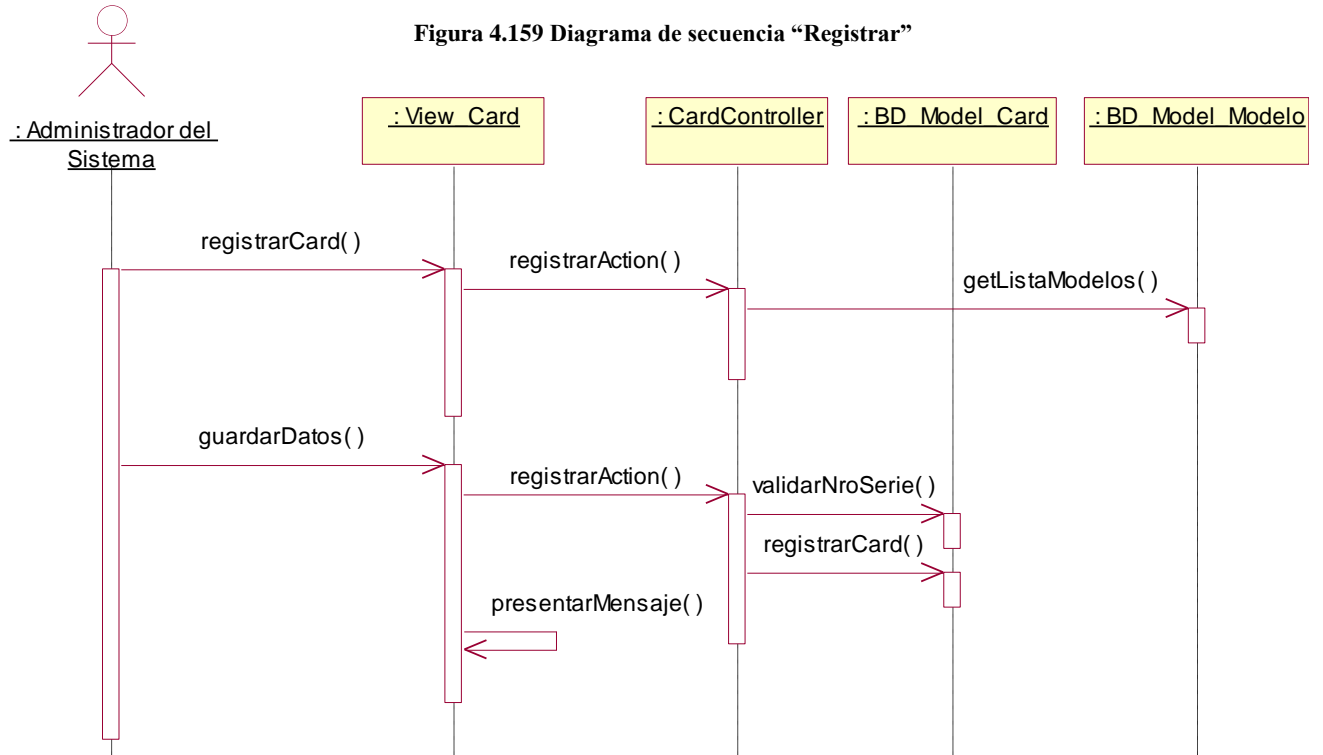


Figura 4.158 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Card” del menú de Administración, este evento invoca a la clase View\_Card ejecutando la función indexCard, la función indexCard hace una llamada a la función indexAction de la clase CardController que se encarga de obtener la lista de equipos de localización registrados en el sistema a través de la función getCard de la clase BD\_Model\_Card, una vez que la función indexAction posee los registros, los entrega a la función indexCard de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verCard de la clase View\_Card, el cual hace una llamada a la función verAction de la clase CardController, esta función realiza una llamada a la función getCardById de la clase BD\_Model\_Card que devuelve el detalle del equipo de localización por el id del equipo, una vez que la función posee el detalle del equipo, la información es presentada a través de la clase View\_Card.

## Diagrama de Secuencia – Registrar



**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de equipos de localización, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarCard de la clase View\_Card, la función registrarCard hace una llamada a la función registrarAction de la clase CardController que solicita la lista de modelos de equipo a la clase BD\_Model\_Modelo a través de la función getListaModelos, posteriormente invita a la función registrarCard presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_Card, la función hace una llamada a la función registrarAction de la clase CardController el cual trata de verificar que el número de serie del equipo sea único y correcto a través de la función validarNroSerie de la clase BD\_Model\_Card, si se cumple la condición se ejecuta la función registrarCard de la misma clase, posteriormente la clase View\_Card puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

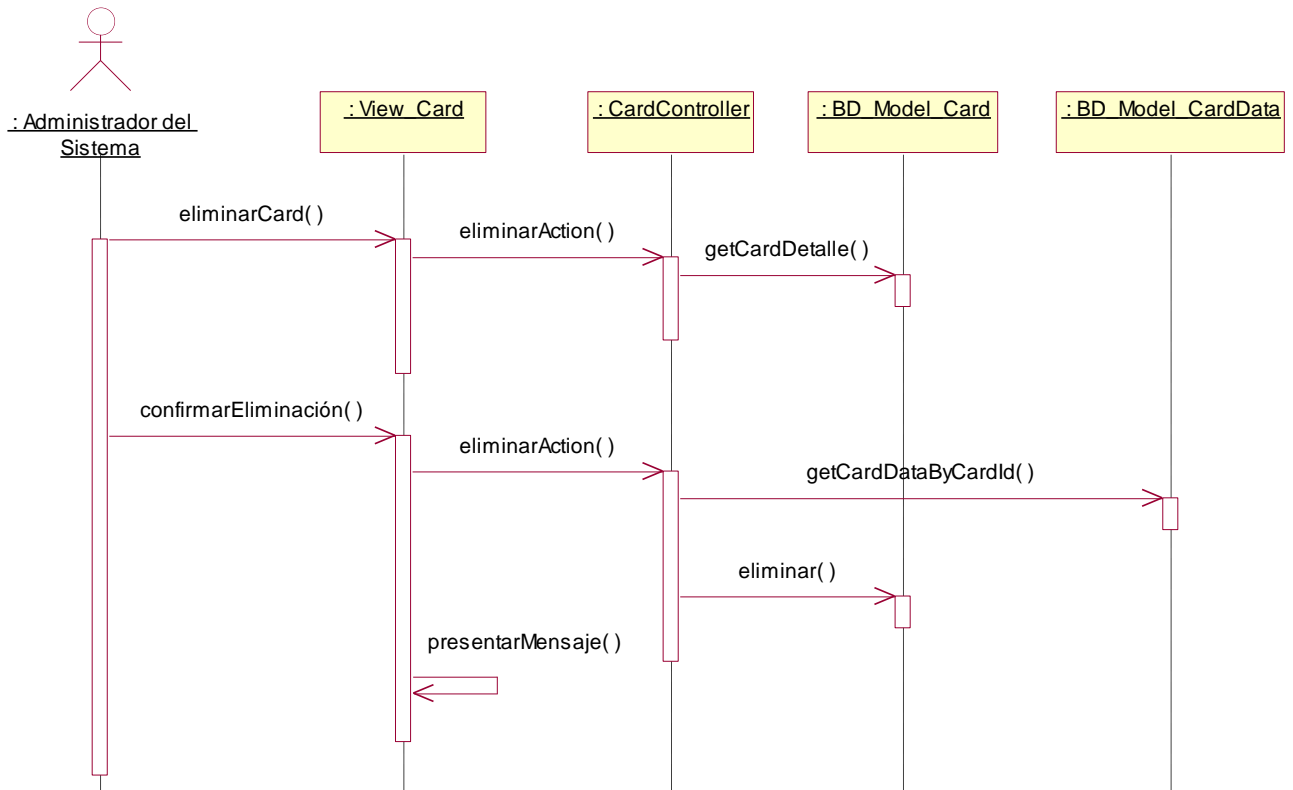


Figura 4.160 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de equipos de localización, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarCard de la clase View\_Card, la función eliminarCard hace una llamada a la función eliminarAction de la clase CardController el cual invoca a la función getCardDetalle de la clase BD\_Model\_Card a fin de obtener los datos del equipo a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarCard y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Card el cual invoca nuevamente a la función eliminarAction de la clase CardController la cual primero verifica que el equipo tenga tramas asociadas a través de la función getCardDataByCardId de la clase BD\_Model\_CardData, una vez que la función eliminarAction confirma que ese equipo no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Card, posteriormente la clase View\_Card a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.7.3 Caso de Uso – Mantenimiento de Unidades

#### Diagrama de Clases

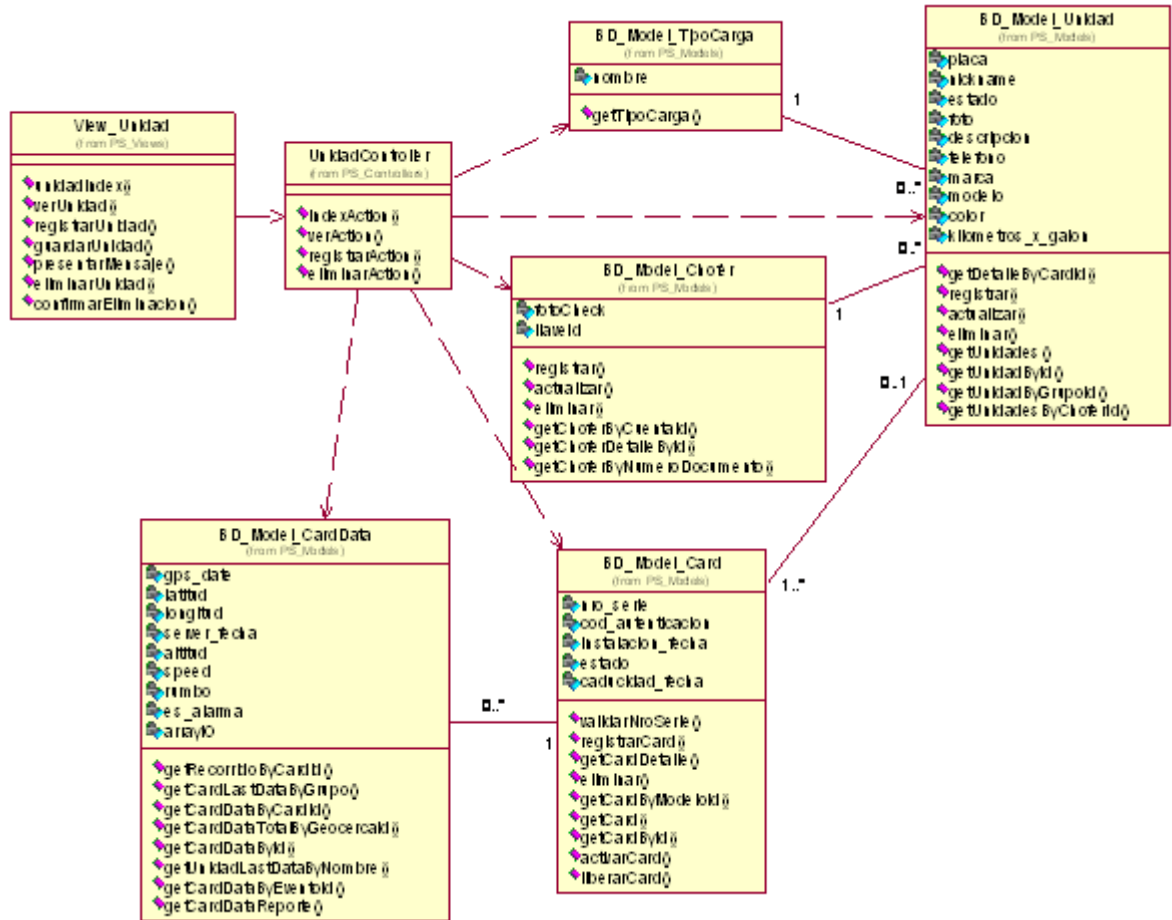


Figura 4.161 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

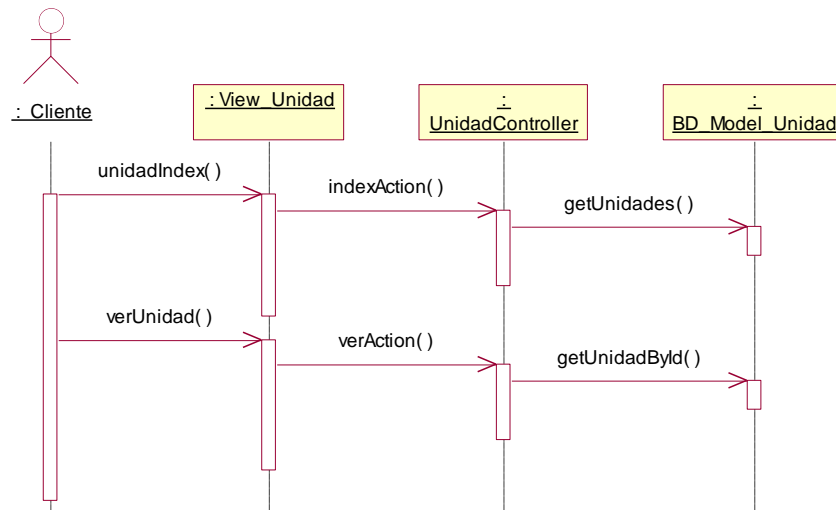


Figura 4.162 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Unidades” del menú de Administración, este evento invoca a la clase View\_Unidad ejecutando la función unidadIndex, la función unidadIndex hace una llamada a la función indexAction de la clase UnidadController que se encarga de obtener la lista de unidades vehiculares registradas en el sistema a través de la función getUnidades de la clase BD\_Model\_Unidad, una vez que la función indexAction posee los registros, los entrega a la función unidadIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verUnidad de la clase View\_Unidad, el cual hace una llamada a la función verAction de la clase UnidadController, esta función realiza una llamada a la función getUnidadById de la clase BD\_Model\_Unidad que devuelve el detalle de la unidad vehicular por el id de la unidad, una vez que la función posee el detalle de la unidad, la información es presentada a través de la clase View\_Unidad.

## Diagrama de Secuencia – Registrar

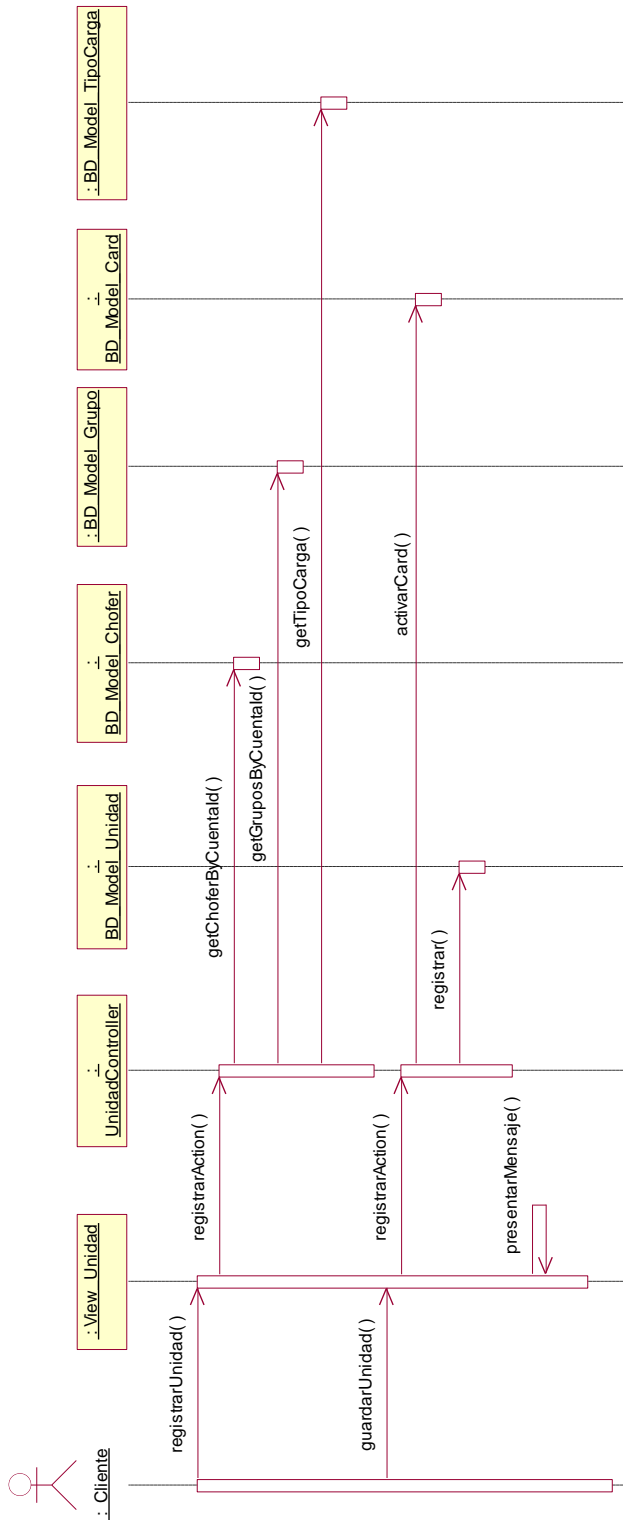


Figura 4.163 Diagrama de secuencia “Registrar”



**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de unidades vehiculares, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarUnidad de la clase View\_Unidad, la función registrarUnidad hace una llamada a la función registrarAction de la clase UnidadController que solicita la lista de choferes de la cuenta cliente a la clase BD\_Model\_Chofer a través de la función getChoferByCuentaId, también obtiene la lista de flotas a través de la función getGruposByCuentaId de la clase BD\_Model\_Grupo, y finalmente de la misma manera obtiene la lista de tipos de cargas a través de la función getTipoCarga de la clase BD\_Model\_TipoCarga, posteriormente invita a la función registrarUnidad presentar el formulario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_Unidad, la función hace una llamada a la función registrarAction de la clase UnidadController la cual activa el equipo de localización a través de la función activarCard de la clase BD\_Model\_Card, y se ejecuta la función registrar de la clase BD\_Model\_Unidad, posteriormente la clase View\_Unidad puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

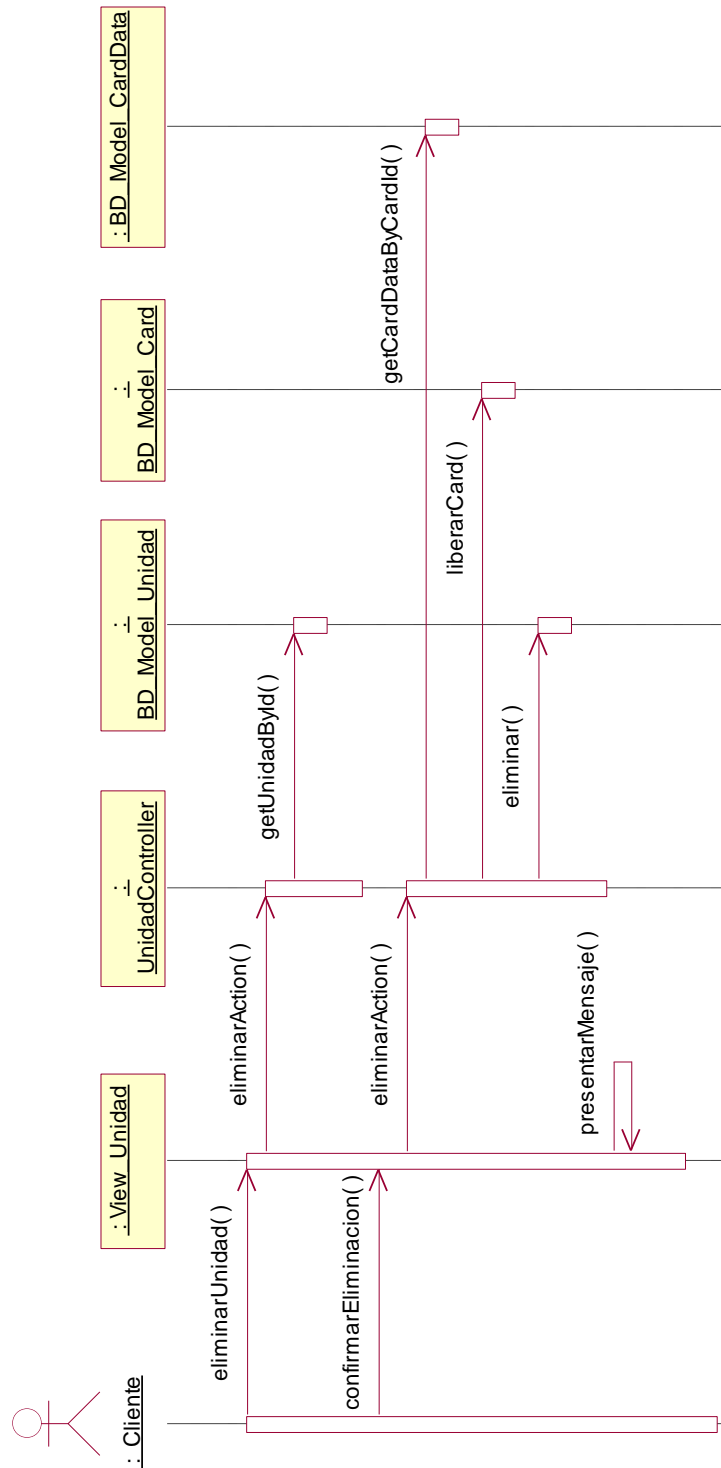


Figura 4.164 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de unidades vehiculares, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarUnidad de la clase View\_Unidad, la función eliminarUnidad hace una llamada a la función eliminarAction de la clase UnidadController el cual invoca a la función getUnidadById de la clase BD\_Model\_Unidad a fin de obtener el detalle de la unidad a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarUnidad y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Unidad el cual invoca nuevamente a la función eliminarAction de la clase UnidadController la cual primero verifica que la unidad no tenga tramas asociadas a través de la función getCardDataByCardId de la clase BD\_Model\_CardData, una vez que la función eliminarAction confirma que ese equipo no tiene registros asociados, ejecuta la función liberarCard de la clase BD\_Model\_Card, posteriormente, ejecuta la función eliminar de la clase BD\_Model\_Unidad, y luego la clase View\_Unidad a través de la función presentarMensaje indica que la operación tuvo éxito.

#### 4.3.7.4 Caso de Uso – Mantenimiento de Flotas

##### Diagrama de Clases

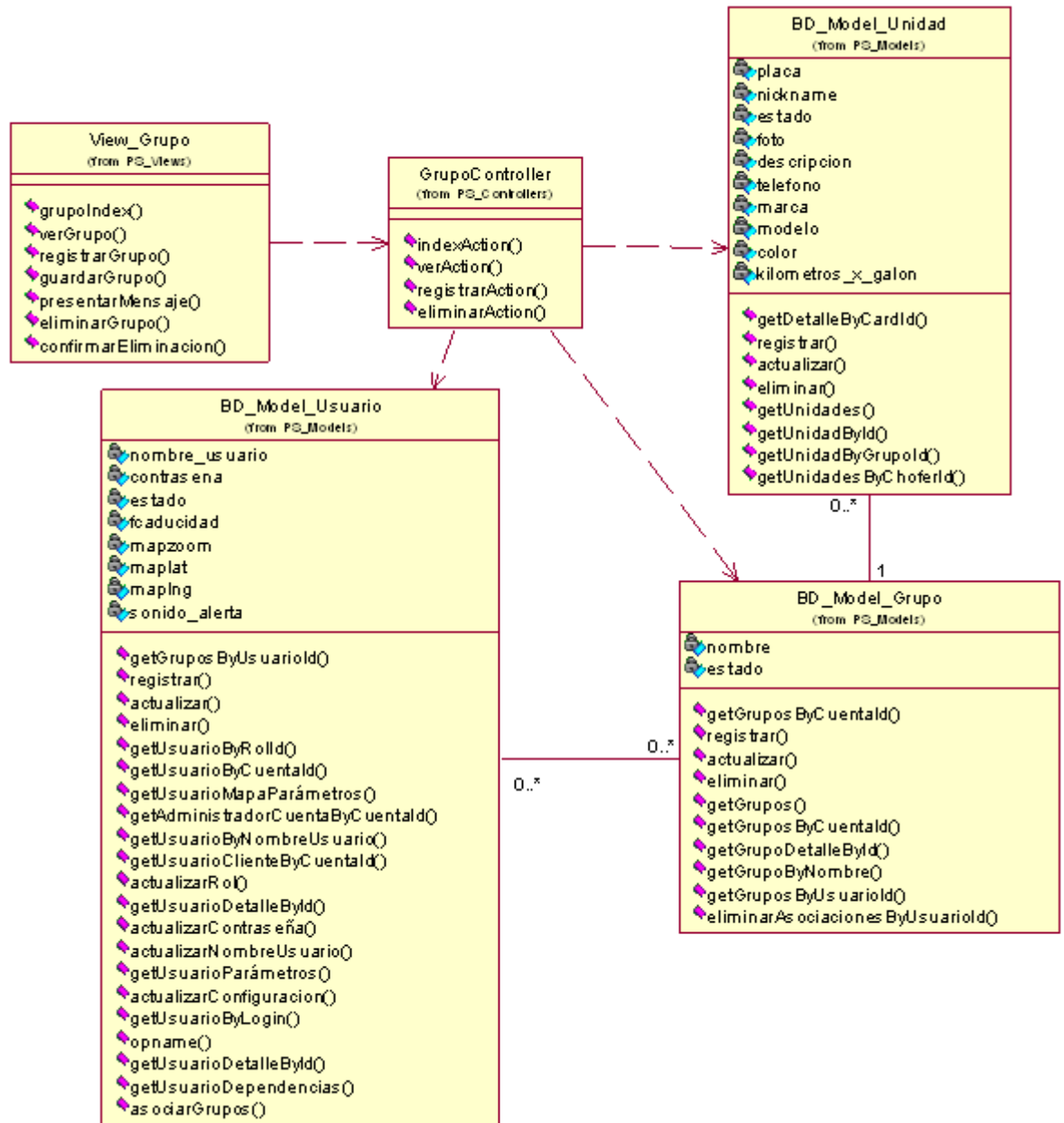


Figura 4.165 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

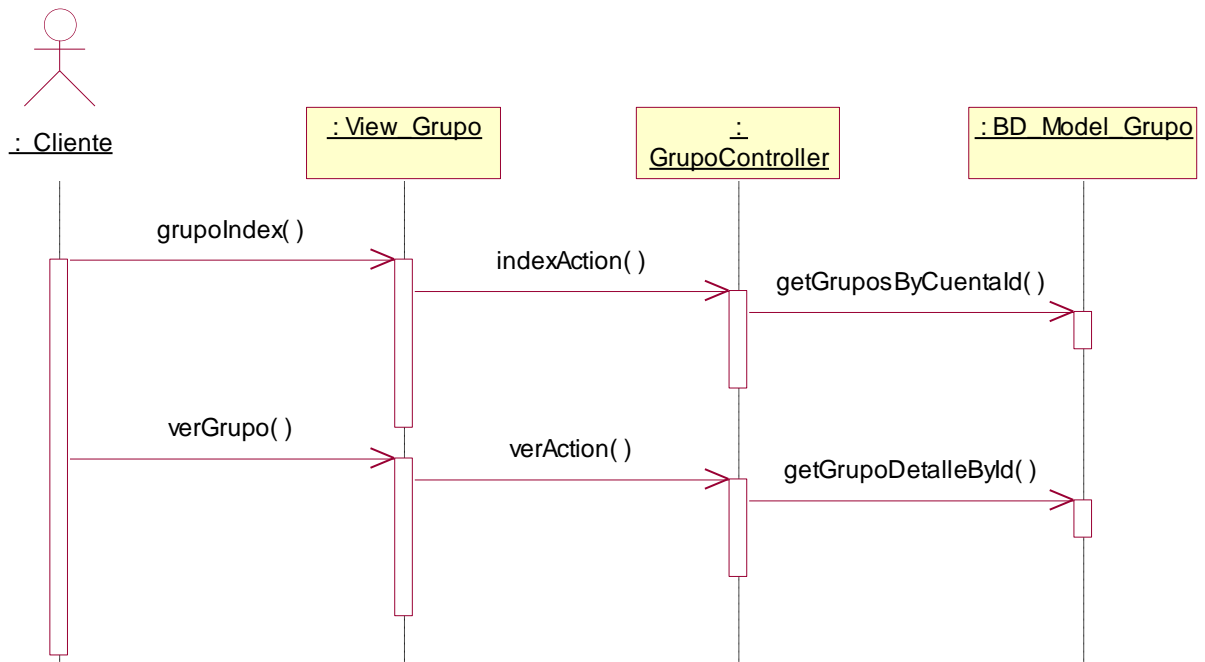


Figura 4.166 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Flotas” del menú de Administración, este evento invoca a la clase View\_Grupo ejecutando la función grupoIndex, la función grupoIndex hace una llamada a la función indexAction de la clase GrupoController que se encarga de obtener la lista de flotas vehiculares registradas en la cuenta cliente a través de la función getGruposByCuentaId de la clase BD\_Model\_Grupo, una vez que la función indexAction posee los registros, los entrega a la función grupoIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verGrupo de la clase View\_Grupo, el cual hace una llamada a la función verAction de la clase GrupoController, esta función realiza una llamada a la función getGrupoDetalleById de la clase BD\_Model\_Grupo que devuelve el detalle de la flota vehicular por el id del grupo, una vez que la función posee el detalle del grupo, la información es presentada a través de la clase View\_Grupo.

## Diagrama de Secuencia – Registrar

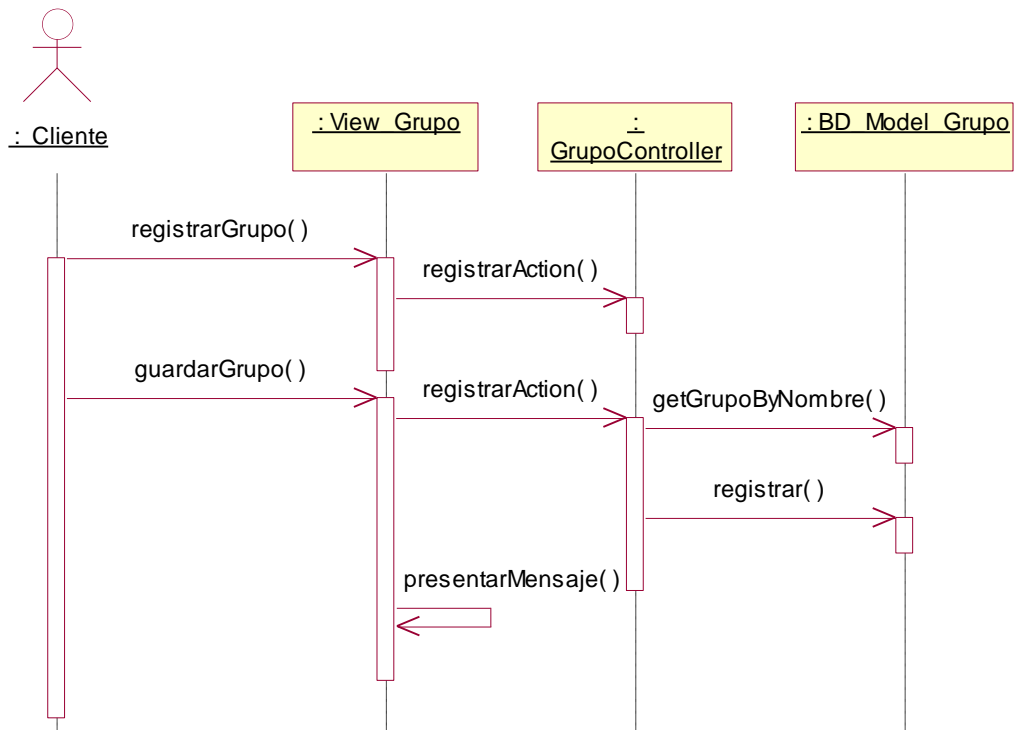


Figura 4.167 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de flotas vehiculares, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarGrupo de la clase View\_Gruppo, la función registrarGrupo hace una llamada a la función registrarAction de la clase GrupoController que invita a la función registrarGrupo presentar el formulario de registro de flotas. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardar de la clase View\_Gruppo, la función hace una llamada a la función registrarAction de la clase GrupoController verifica que el nombre de la flota sea única, por eso llama a la función getGrupoByNombre de la clase BD\_Model\_Gruppo, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Gruppo, posteriormente la clase View\_Gruppo puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

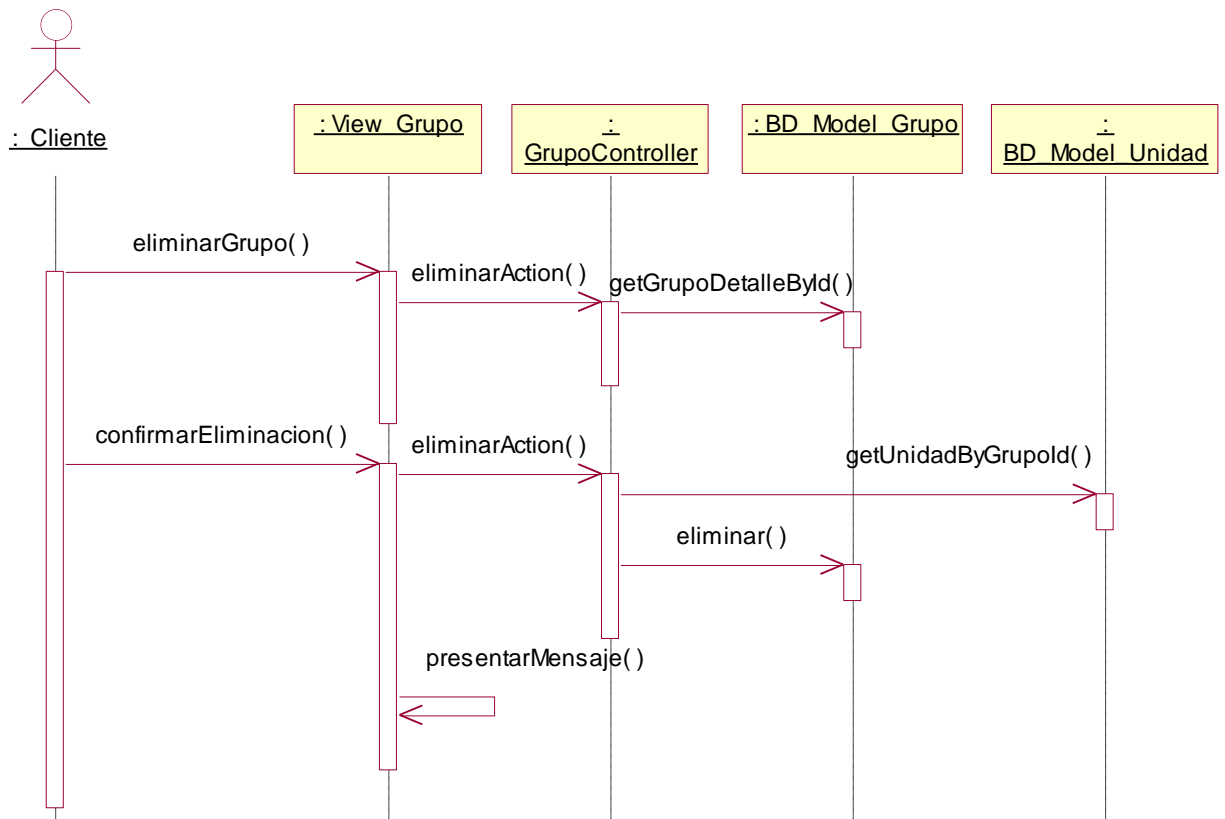


Figura 4.168 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de flotas vehiculares, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarGrupo de la clase View\_Grupo, la función eliminarGrupo hace una llamada a la función eliminarAction de la clase GrupoController el cual invoca a la función getGrupoDetalleById de la clase BD\_Model\_Grupo a fin de obtener el detalle de la flota a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarGrupo y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Grupo el cual invoca nuevamente a la función eliminarAction de la clase GrupoController la cual primero verifica que la flota no tenga unidades asociadas a través de la función getUnidadByGrupold de la clase BD\_Model\_Unidad, una vez que la función eliminarAction confirma que esa flota no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Grupo, y luego la clase View\_Grupo a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.7.5 Caso de Uso – Mantenimiento de Conductores

#### Diagrama de Clases

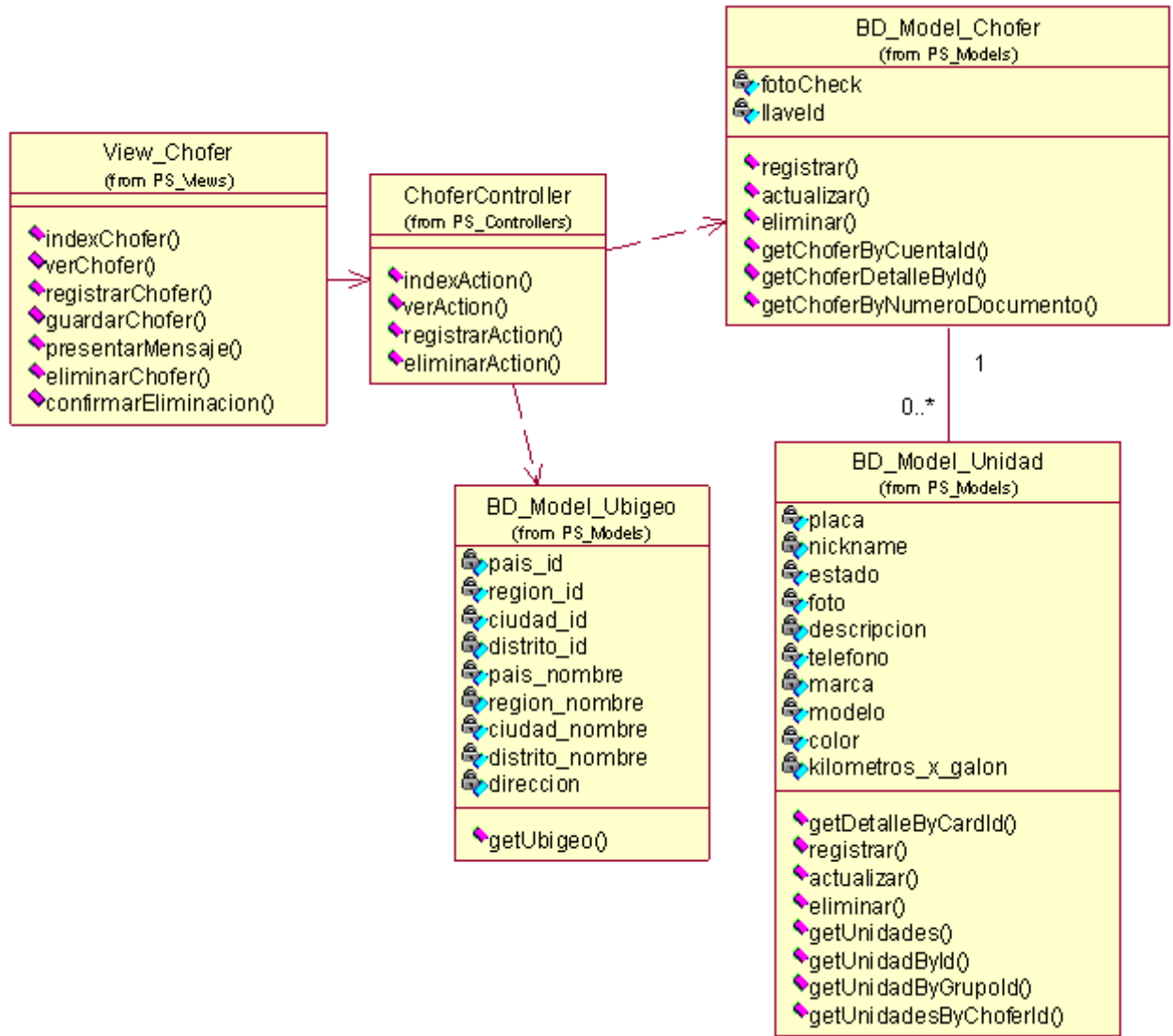


Figura 4.169 Diagrama de clases del caso de uso



## Diagrama de Secuencia –Listar

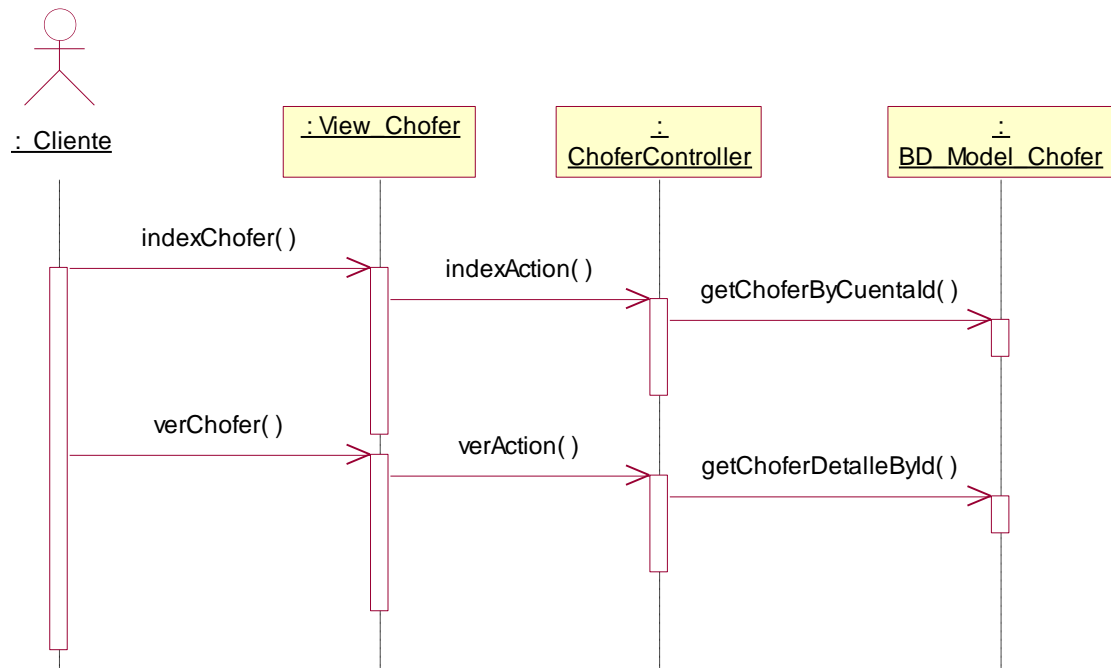


Figura 4.170 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Conductores” del menú de Administración, este evento invoca a la clase View\_Chofer ejecutando la función indexChofer, la función indexChofer hace una llamada a la función indexAction de la clase ChoferController que se encarga de obtener la lista de conductores registrados en la cuenta cliente a través de la función getChoferByCuentaId de la clase BD\_Model\_Chofer, una vez que la función indexAction posee los registros, los entrega a la función indexChofer de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verChofer de la clase View\_Chofer, el cual hace una llamada a la función verAction de la clase ChoferController, esta función realiza una llamada a la función getChoferDetalleById de la clase BD\_Model\_Chofer que devuelve el detalle del conductor por el id del chofer, una vez que la función posee el detalle del chofer, la información es presentada a través de la clase View\_Chofer.

## Diagrama de Secuencia – Registrar

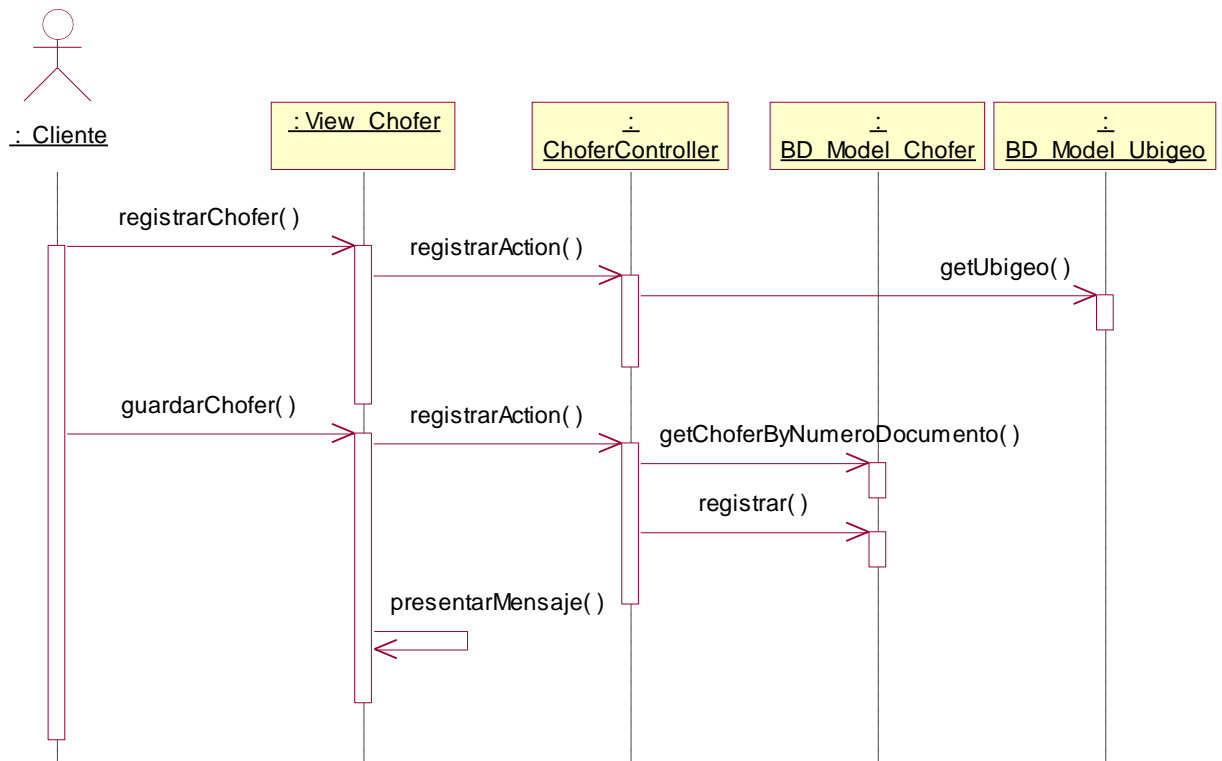


Figura 4.171 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de conductores, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarChofer de la clase View\_Chofer, la función registrarChofer hace una llamada a la función registrarAction de la clase ChoferController que solicita la lista de departamentos, ciudades y distritos a la clase BD\_Model\_Ubigeo a través de la función getUbigeo, posteriormente invita a la función registrarChofer presentar el formulario de registro de conductor. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarChofer de la clase View\_Chofer, la función hace una llamada a la función registrarAction de la clase ChoferController verifica que el documento de identidad del conductor sea único, por eso llama a la función getChoferByNumeroDocumento de la clase BD\_Model\_Chofer, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Chofer, posteriormente la clase View\_Chofer puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

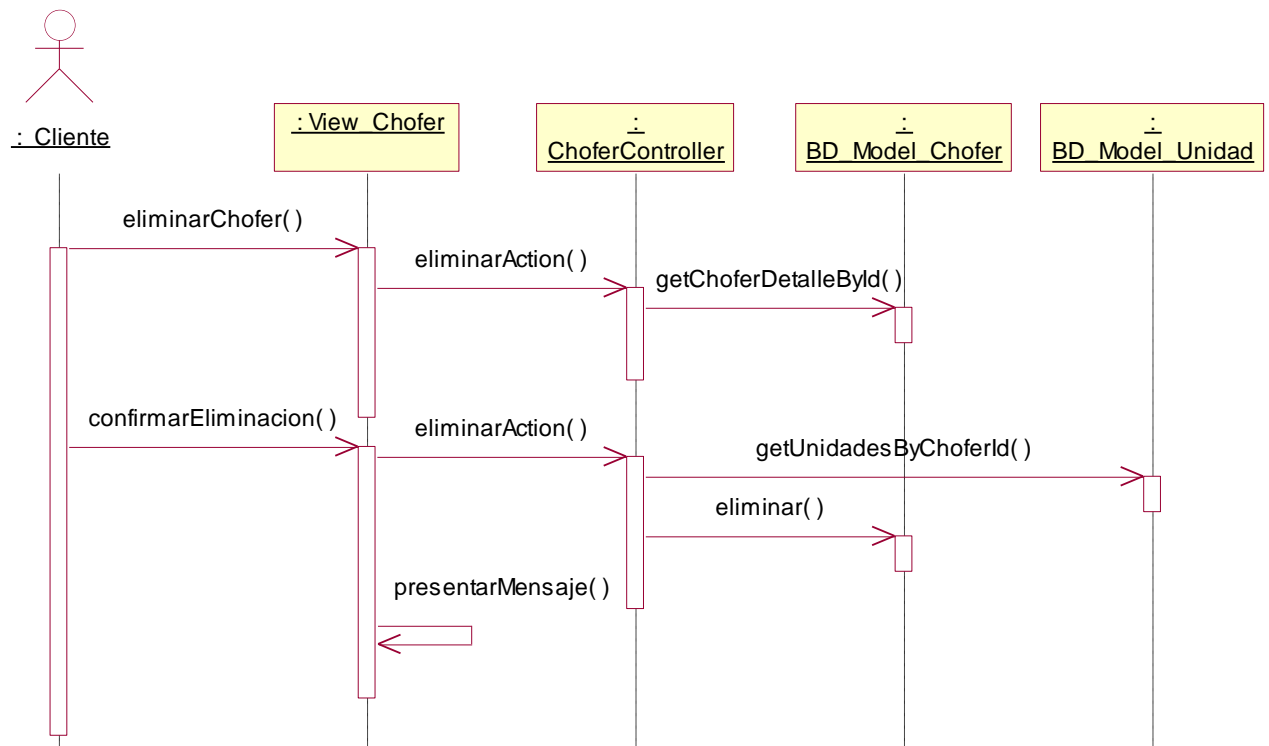


Figura 4.172 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de conductores, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarChofer de la clase View\_Chofer, la función eliminarChofer hace una llamada a la función eliminarAction de la clase ChoferController el cual invoca a la función getChoferDetalleById de la clase BD\_Model\_Chofer a fin de obtener el detalle del conductor a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarChofer y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Chofer el cual invoca nuevamente a la función eliminarAction de la clase ChoferController la cual primero verifica que el conductor no tenga unidades asociadas a través de la función getUnidadByChoferId de la clase BD\_Model\_Unidad, una vez que la función eliminarAction confirma que ese conductor no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Chofer, y luego la clase View\_Chofer a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.7.6 Caso de Uso – Mantenimiento de Geocercas

#### Diagrama de Clases

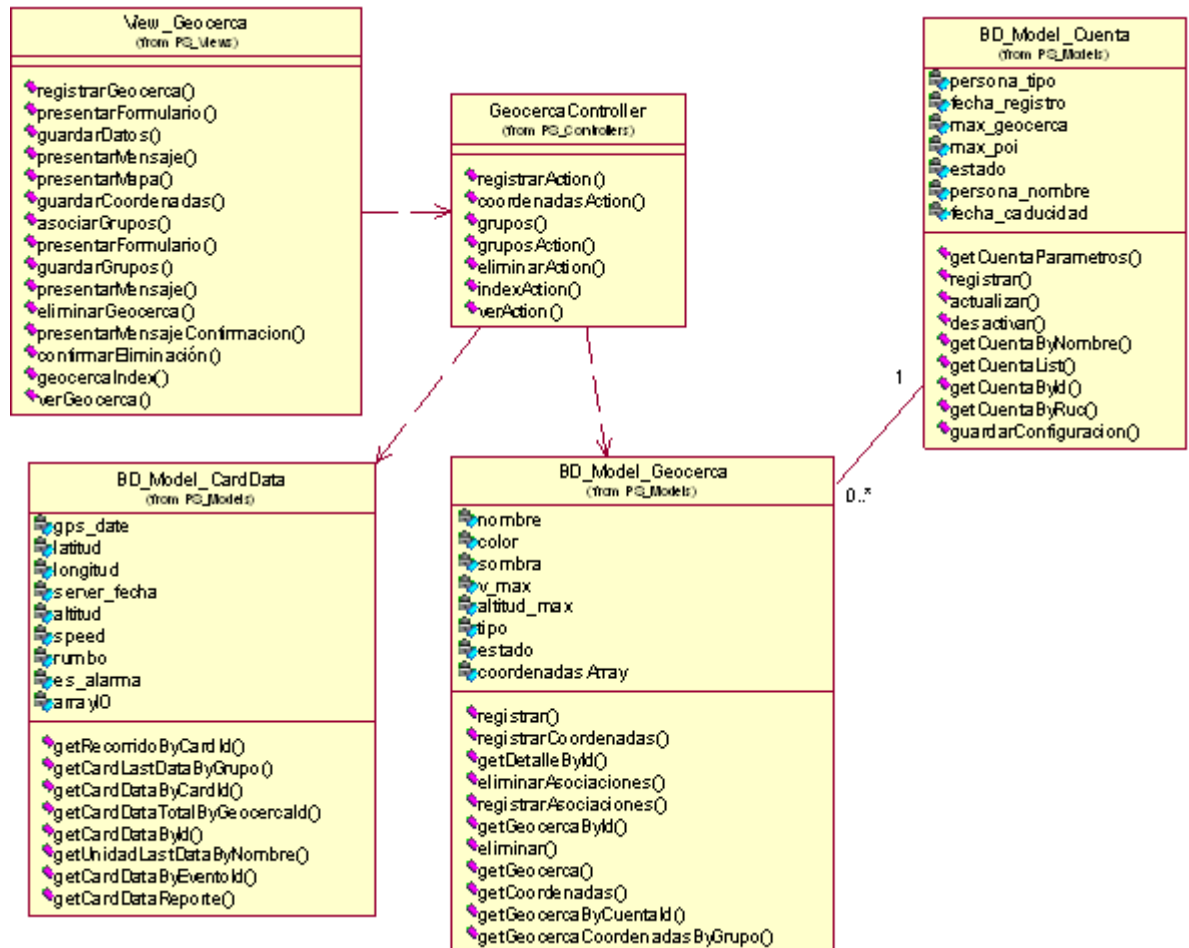


Figura 4.173 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

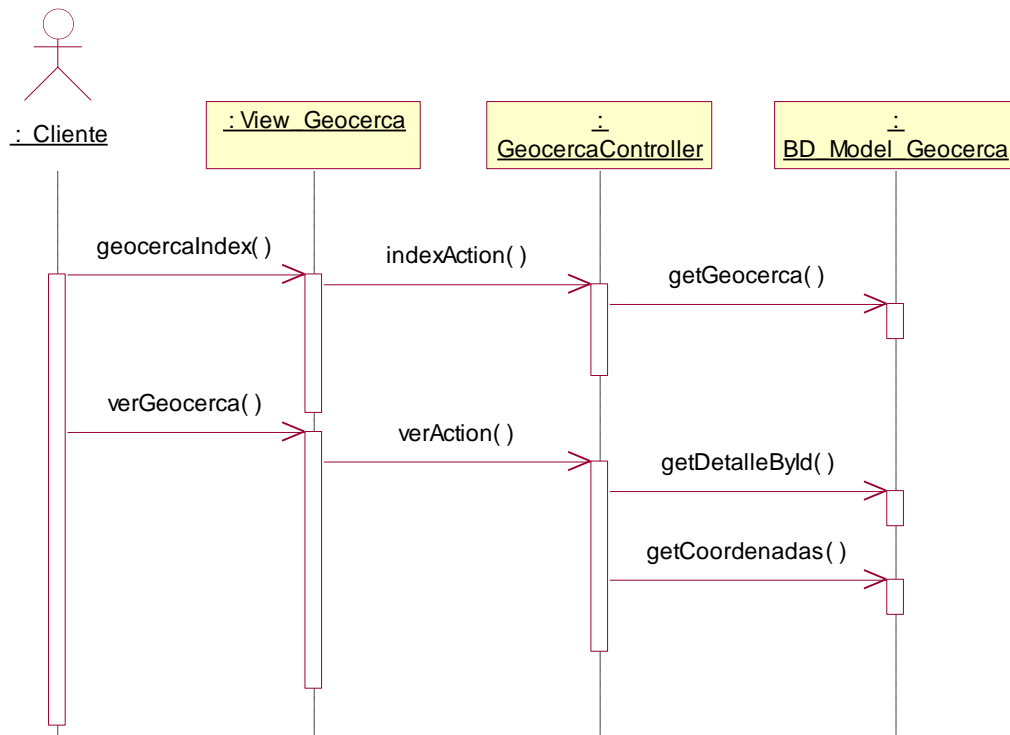


Figura 4.174 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Geocercas” del menú de Administración, este evento invoca a la clase View\_Geocerca ejecutando la función geocercaIndex, la función geocercaIndex hace una llamada a la función indexAction de la clase GeocercaController que se encarga de obtener la lista de geocercas registradas en la cuenta cliente a través de la función getGeocerca de la clase BD\_Model\_Geocerca, una vez que la función indexAction posee los registros, los entrega a la función geocercaIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verGeocerca de la clase View\_Geocerca, el cual hace una llamada a la función verAction de la clase GeocercaController, esta función realiza una llamada a la función getGeocercaById de la clase BD\_Model\_Geocerca y getCoordenadas de la misma clase, que devuelve el detalla de la geocerca y sus coordenadas respectivamente, una vez que la función posee el detalle de la geocerca, la información es presentada a través de la clase View\_Geocerca.

## Diagrama de Secuencia – Registrar

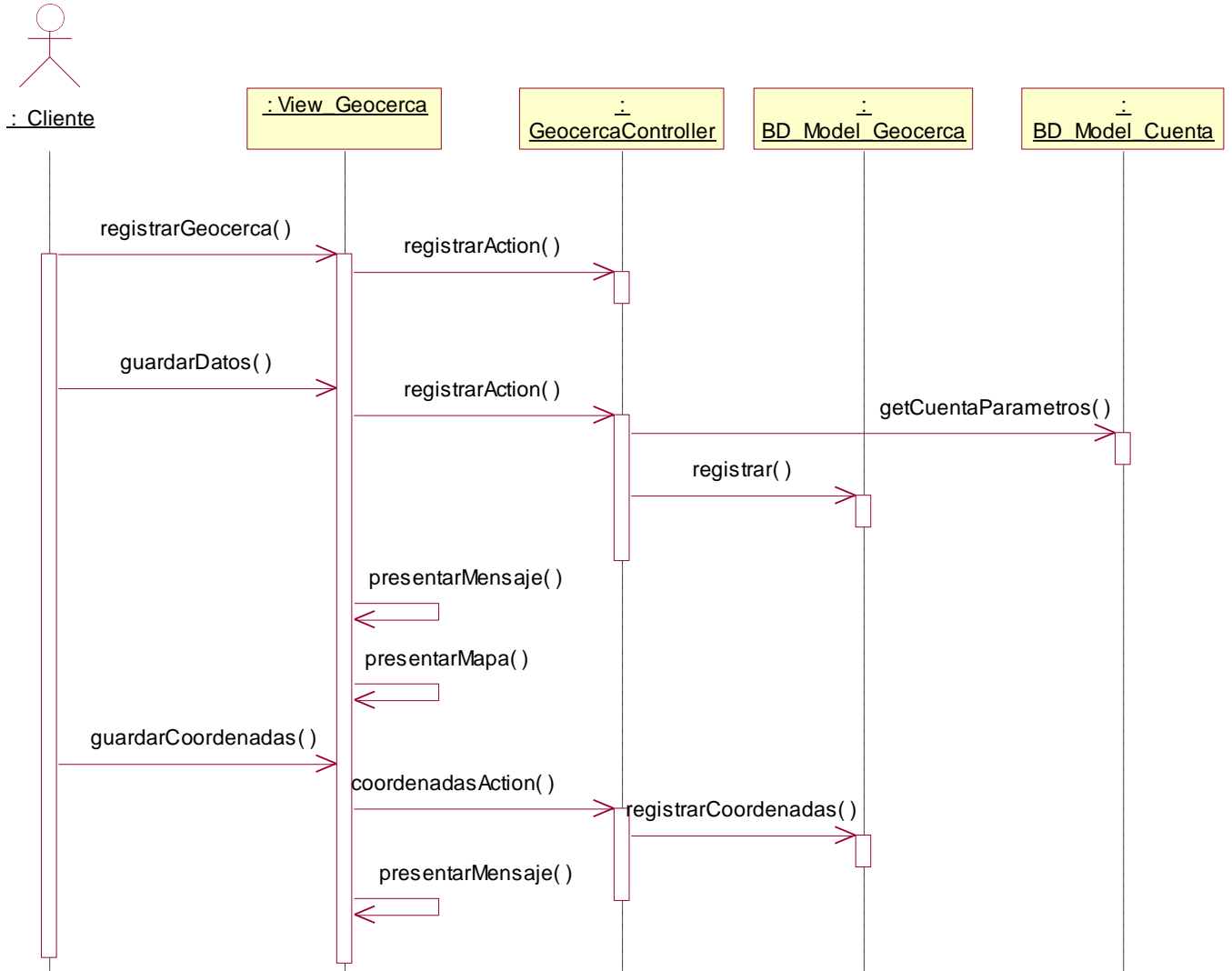


Figura 4.175 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de geocercas, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarGeocerca de la clase View\_Geocerca, la función registrarGeocerca hace una llamada a la función registrarAction de la clase GeocercaController que invita a la función registrarGeocerca presentar el formulario de registro de geocerca. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarDatos de la clase View\_Geocerca, la función hace una llamada a la función registrarAction de la clase GeocercaController verifica que los parámetros de la cuenta permitan registrar más geocercas, por eso llama a la función getContaParámetros de la clase BD\_Model\_Cuenta, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Geocerca, posteriormente la clase

View\_Geocerca puede presentar un mensaje indicando que la operación tuvo éxito, seguido a eso la clase View\_Geocerca a través de la función presentarMapa entrega un mapa donde el usuario puede indicar las coordenadas de la geocerca, una vez que definió las coordenadas de la geocercas, el usuario hace clic en el botón “Actualizar”, este evento ejecuta la función guardarCoordenadas de la clase View\_Geocerca, esta función llama a la función coordenadasAction de la clase GeocercaController la cual registra las coordenadas de la geocerca a través de la función registrarCoordenadas de la clase BD\_Model\_Geocerca, posteriormente la clase View\_Geocerca puede mostrar un mensaje de éxito a través de la función presentarMensaje.

### Diagrama de Secuencia – Eliminar

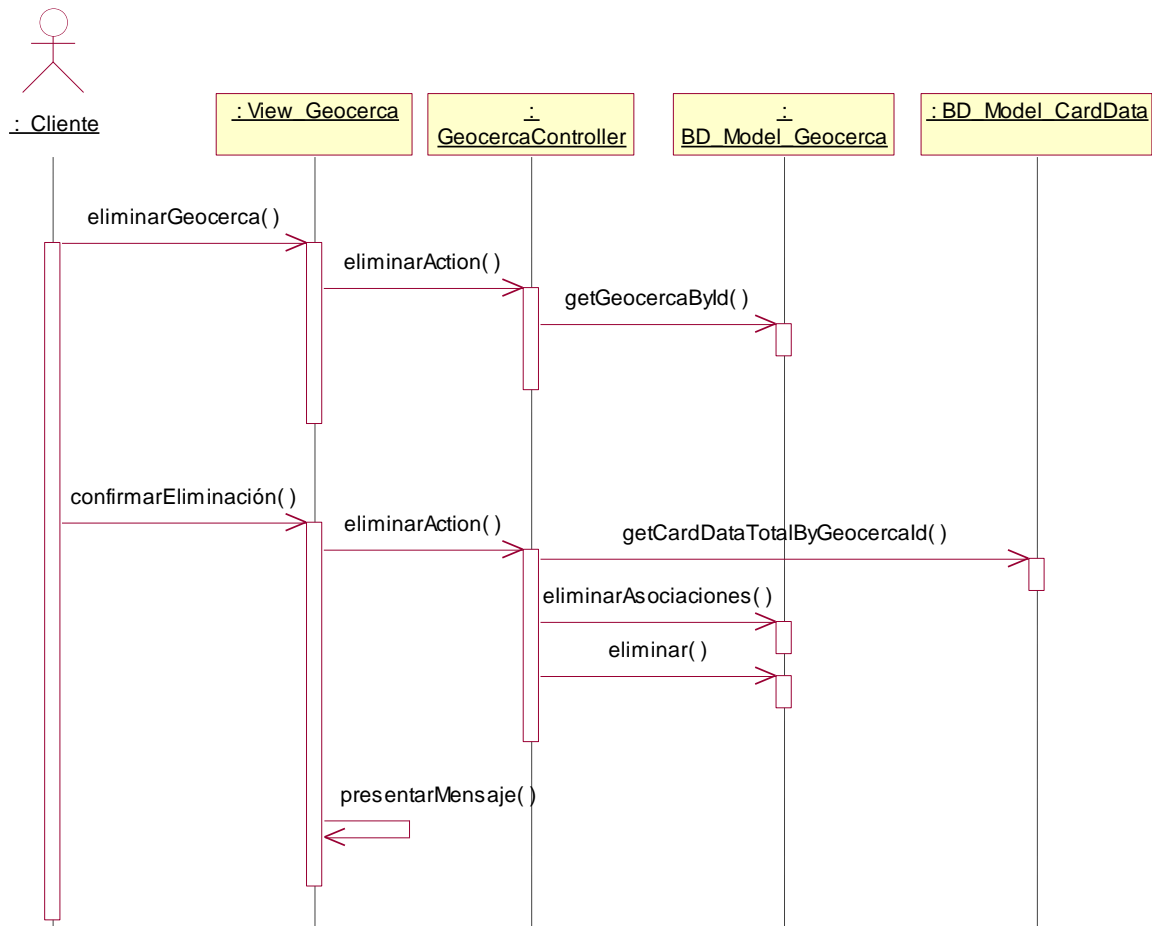


Figura 4.176 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de geocercas, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarGeocerca de la clase View\_Geocerca, la función eliminarGeocerca hace una llamada a la función eliminarAction de la clase GeocercaController el cual invoca a

la función `getGeocercaById` de la clase `BD_Model_Geocerca` a fin de obtener el detalle de la geocerca a eliminar, una vez que la función `eliminarAction` tiene los datos, los pasa a la función `eliminarGeocerca` y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función `confirmarEliminación` de la clase `View_Geocerca` el cual invoca nuevamente a la función `eliminarAction` de la clase `GeocercaController` la cual primero verifica que la geocerca no tenga tramas asociadas a través de la función `getCardDataTotalByGeocercaId` de la clase `BD_Model_CardData`, una vez que la función `eliminarAction` confirma que ese conductor no tiene registros asociados, ejecuta la función `eliminarAsociaciones` que se encarga de eliminar las coordenadas de la geocerca, luego ejecuta la función `eliminar` de la clase `BD_Model_Geocerca`, y luego la clase `View_Geocerca` a través de la función `presentarMensaje` indica que la operación tuvo éxito.

### Diagrama de Secuencia – Asociar Flotas

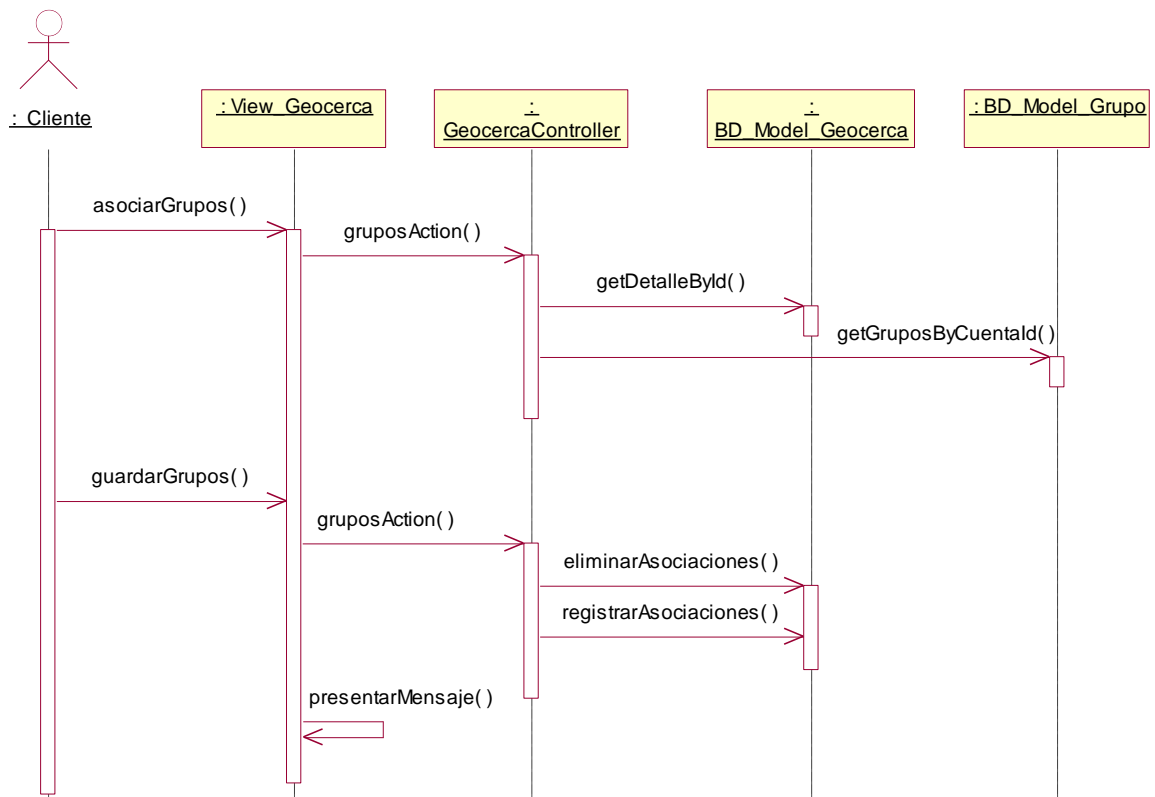


Figura 4.177 Diagrama de secuencia “Asociar Flotas”

**Descripción del diagrama de secuencia – Asociar Flotas:** El usuario en la lista de geocercas, hace clic en la opción “Flotas”, este evento ejecuta la opción `asociarGrupos` de la clase `View_Geocerca`, la función `asociarGrupos` hace una llamada a la función `gruposAction` de la clase `GeocercaController` el cual invoca a la función `getDetalleById` de la clase `BD_Model_Geocerca` a fin de obtener el detalle de la geocerca y llama a la función `getGruposByCuentaId` de la cuenta



BD\_Model\_Grupo a fin de obtener a las flotas de la cuenta cliente, una vez que la función gruposAction tiene los datos, los pasa a la función asociarGrupos y estos son presentados al usuario. El usuario selecciona las flotas y confirma la asociación haciendo clic en el botón “Actualizar”, este evento ejecuta la función guardarGrupos de la clase View\_Geocerca el cual invoca nuevamente a la función gruposAction de la clase GeocercaController la cual primero elimina las asociaciones anteriores a través de la función eliminarAsociaciones de la clase BD\_Model\_CardData, una vez que la función eliminarAction confirma que ese conductor no tiene registros asociados, ejecuta la función eliminarAsociaciones que se encarga de eliminar las coordenadas previas de la geocerca, luego ejecuta la función registrarAsociaciones de la clase BD\_Model\_Geocerca a fin de guardar las nuevas coordenadas, y luego la clase View\_Geocerca a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.7.7 Caso de Uso – Mantenimiento de Puntos de Interés

#### Diagrama de Clases

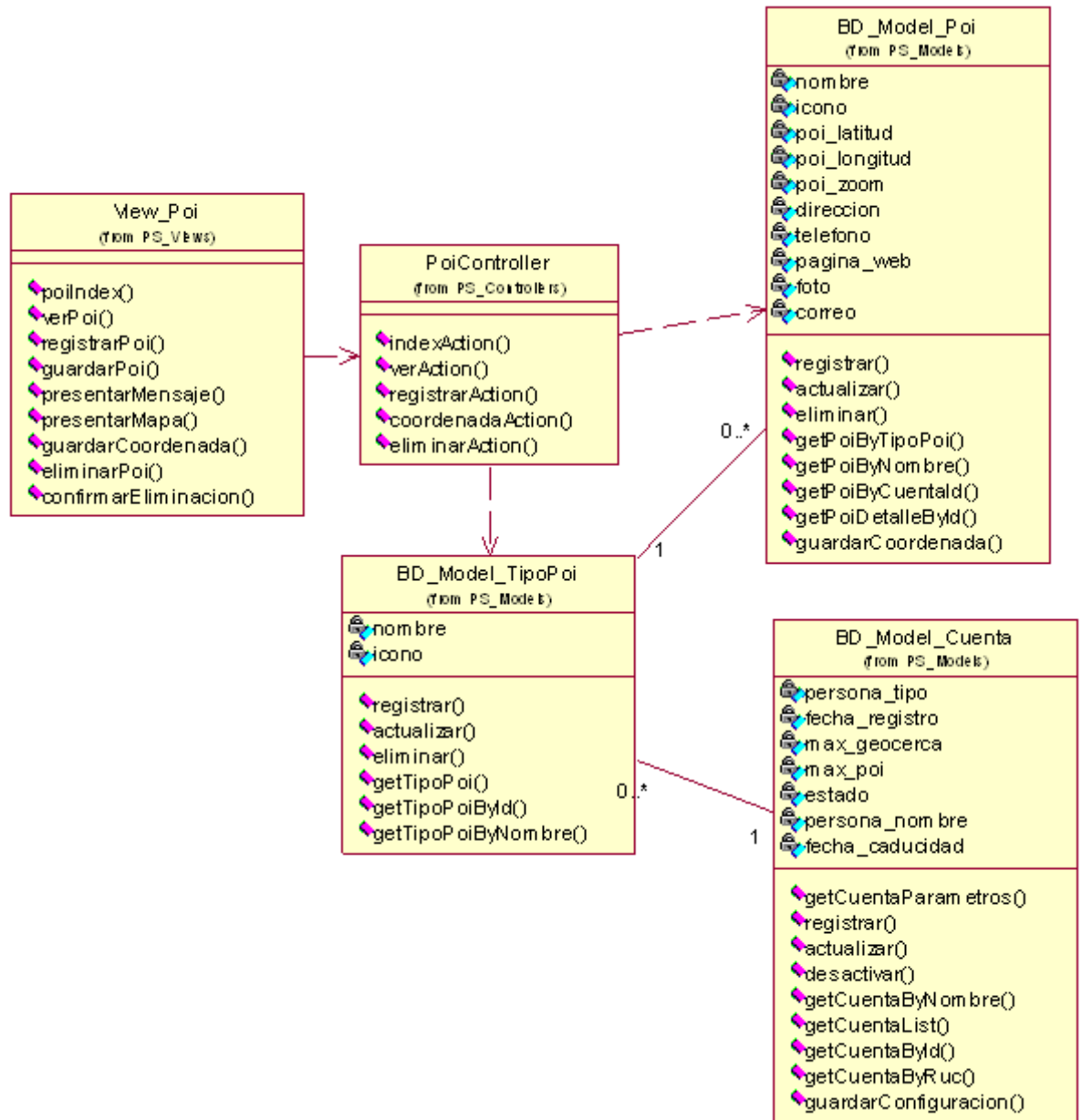


Figura 4.178 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

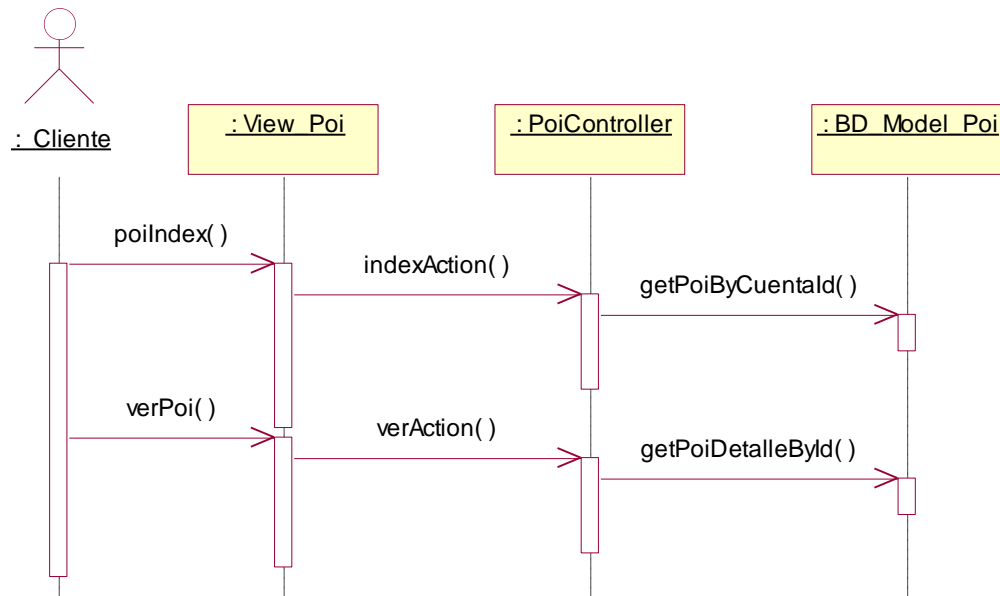


Figura 4.179 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “POI” del menú de Administración, este evento invoca a la clase View\_Poi ejecutando la función poiIndex, la función poiIndex hace una llamada a la función indexAction de la clase PoiController que se encarga de obtener la lista de puntos de interés registrados en la cuenta cliente a través de la función getPoiByCuentaId de la clase BD\_Model\_Poi, una vez que la función indexAction posee los registros, los entrega a la función poiIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verPoi de la clase View\_Poi, el cual hace una llamada a la función verAction de la clase PoiController, esta función realiza una llamada a la función getPoiDetalleById de la clase BD\_Model\_Poi, una vez que la función posee el detalle del punto de interés, la información es presentada a través de la clase View\_Poi.

## Diagrama de Secuencia – Registrar

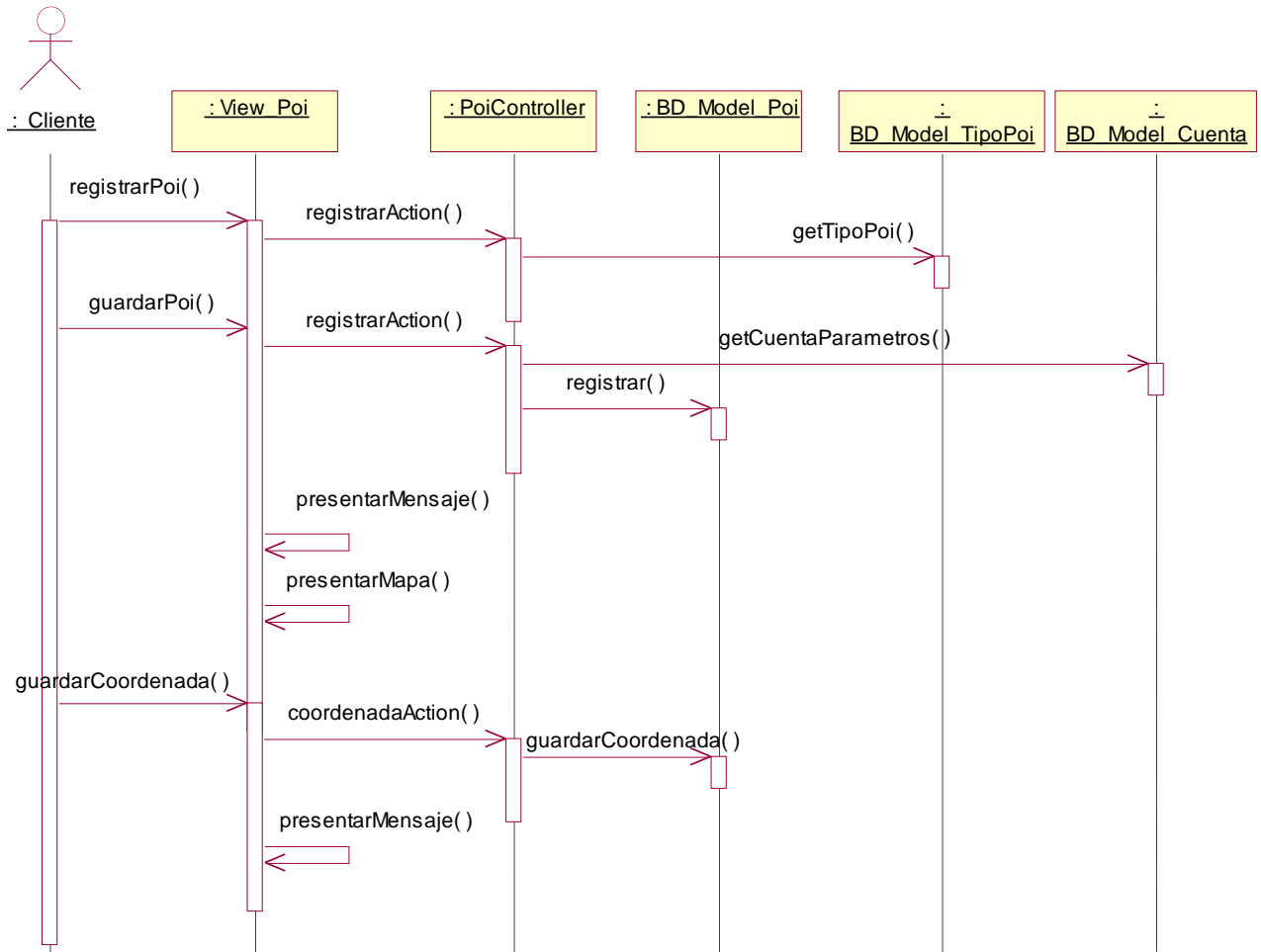


Figura 4.180 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de puntos de interés, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarPoi de la clase View\_Poi, la función registrarPoi hace una llamada a la función registrarAction de la clase PoiController que solicita los tipos de puntos de interés de la clase BD\_Model\_TipoPoi a través de la función getTipoPoi, luego invita a la función registrarPoi presentar el formulario de registro de punto de interés. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarPoi de la clase View\_Poi, la función hace una llamada a la función registrarAction de la clase PoiController verifica que los parámetros de la cuenta permitan registrar más puntos de interés, por eso llama a la función getCuentaParámetros de la clase BD\_Model\_Cuenta, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Poi, posteriormente

la clase View\_Poi puede presentar un mensaje indicando que la operación tuvo éxito, seguido a eso la clase View\_Poi a través de la función presentarMapa entrega un mapa donde el usuario puede indicar las coordenadas del punto de interés, una vez que definió las coordenadas del punto de interés, el usuario hace clic en el botón “Actualizar”, este evento ejecuta la función guardarCoordenada de la clase View\_Poi, esta función llama a la función coordenadaAction de la clase PoiController la cual registra la coordenada del punto de interés a través de la función guardarCoordenada de la clase BD\_Model\_Poi, posteriormente la clase View\_Poi puede mostrar un mensaje de éxito a través de la función presentarMensaje.

### Diagrama de Secuencia – Eliminar

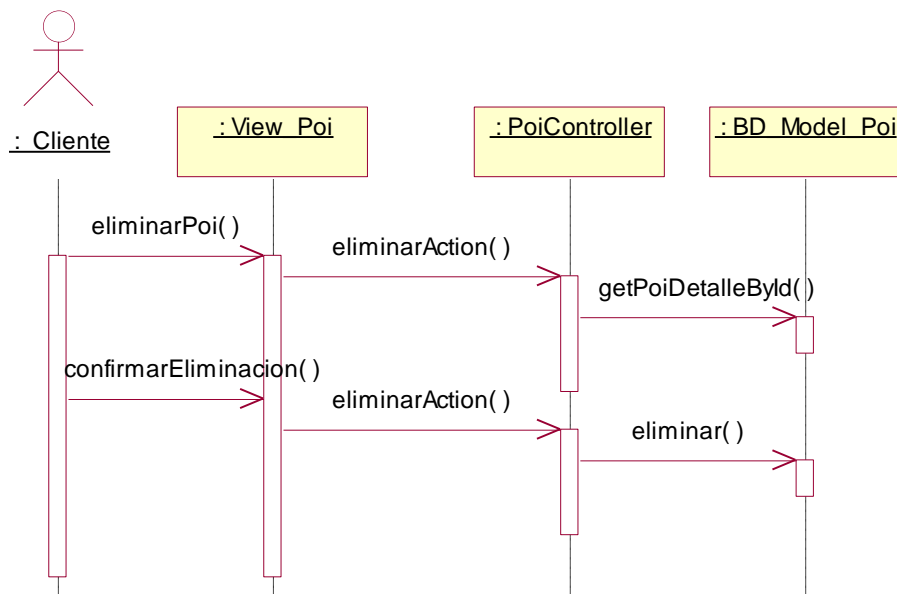


Figura 4.181 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de puntos de interés, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarPoi de la clase View\_Poi, la función eliminarPoi hace una llamada a la función eliminarAction de la clase PoiController el cual invoca a la función getPoiDetalleById de la clase BD\_Model\_Poi a fin de obtener el detalle del punto de interés a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarPoi y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Poi el cual invoca nuevamente a la función eliminarAction de la clase PoiController la ejecuta la función eliminar de la clase BD\_Model\_Poi, y luego la clase View\_Poi a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.8 Realización de casos de uso de diseño – Paquete Reportes

#### 4.3.8.1 Caso de Uso – Generar Reporte de Recorrido

##### Diagrama de Clases

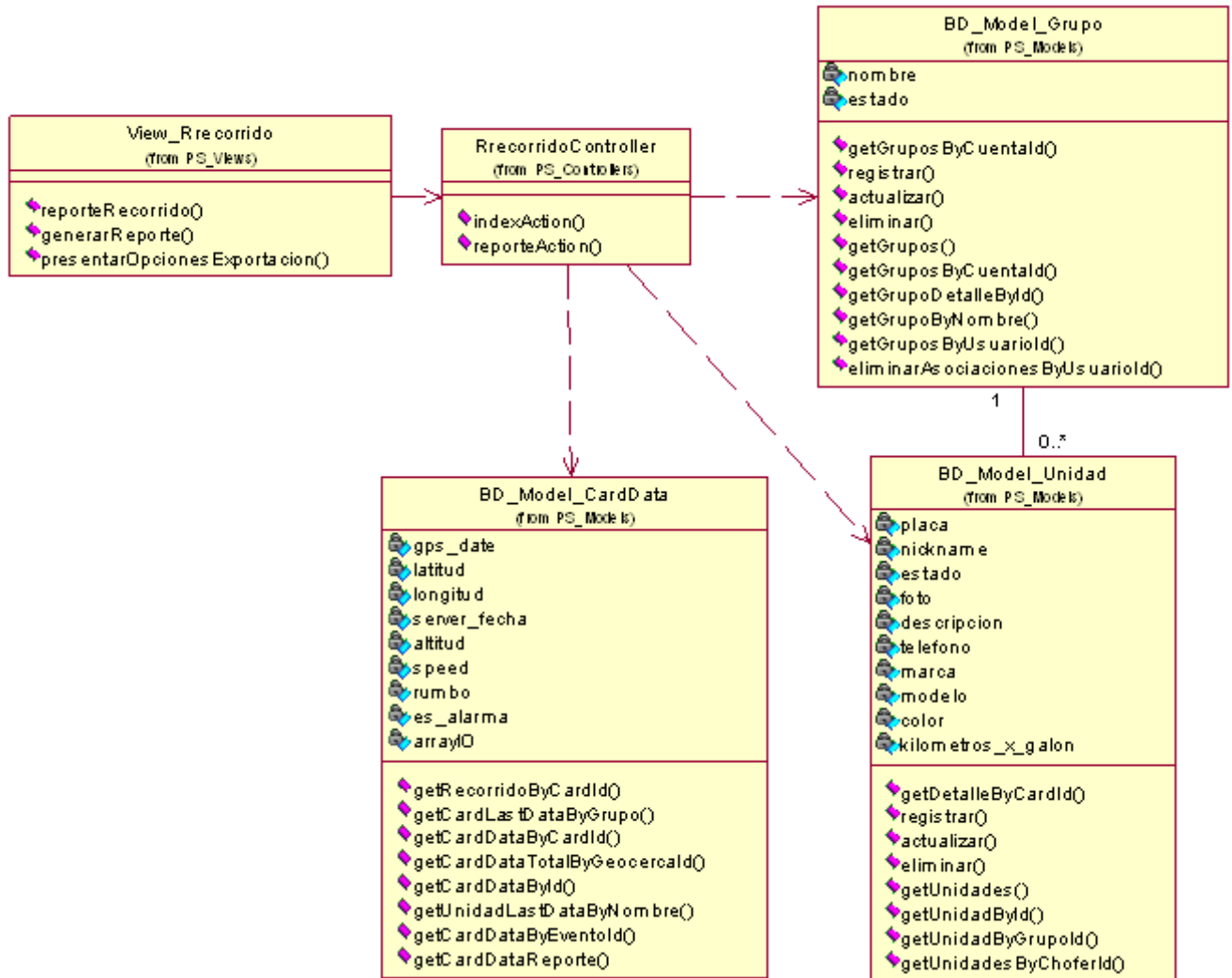


Figura 4.182 Diagrama de clases del caso de uso

## Diagrama de Secuencia – Generar Reporte

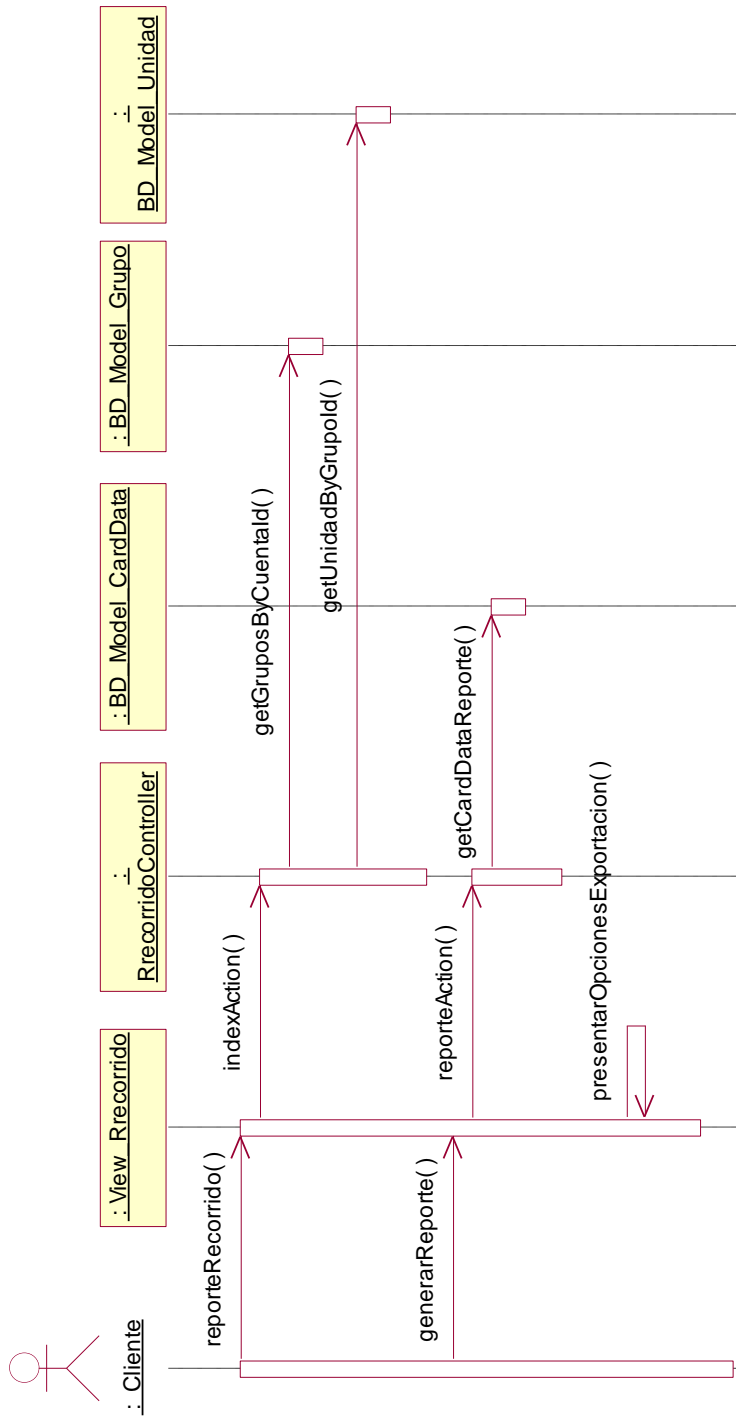


Figura 4.183 Diagrama de secuencia “Generar Reporte”

**Descripción del diagrama de secuencia – Generar Reporte:** El usuario al seleccionar la opción “Recorrido” del menú de Reportes, invoca a la clase View\_Rrecorrido ejecutando la función reporteRecorrido, la función reporteRecorrido hace una llamada a la función indexAction de la clase RrecorridoController que se encarga de obtener la lista de flotas de la cuenta a través de la función getGruposByCuentaId de la clase BD\_Model\_Grupo, así mismo obtiene la lista de unidades a través de la función getUnidadByGrupoId de la clase BD\_Model\_Unidad, una vez que la función indexAction posee los registros, los entrega a la función reporteRecorrido para presentar el formulario de reporte de recorrido. El usuario ingresa los datos solicitados y hace clic en el botón Siguiente, al ejecutar el evento se llama a la función generarReporte de la clase View\_Rrecorrido, la cual llama a la función reporteAction que obtiene los datos del reporte a través de la función getCardDataReporte perteneciente a la clase BD\_Model\_CardData, posteriormente la clase View\_Rrecorrido presenta el reporte en formato HTML de forma paginada y presenta las opciones de exportación a través de la función presentarOpcionesExportacion.



### 4.3.8.2 Caso de Uso – Generar Reporte de Alertas

#### Diagrama de Clases

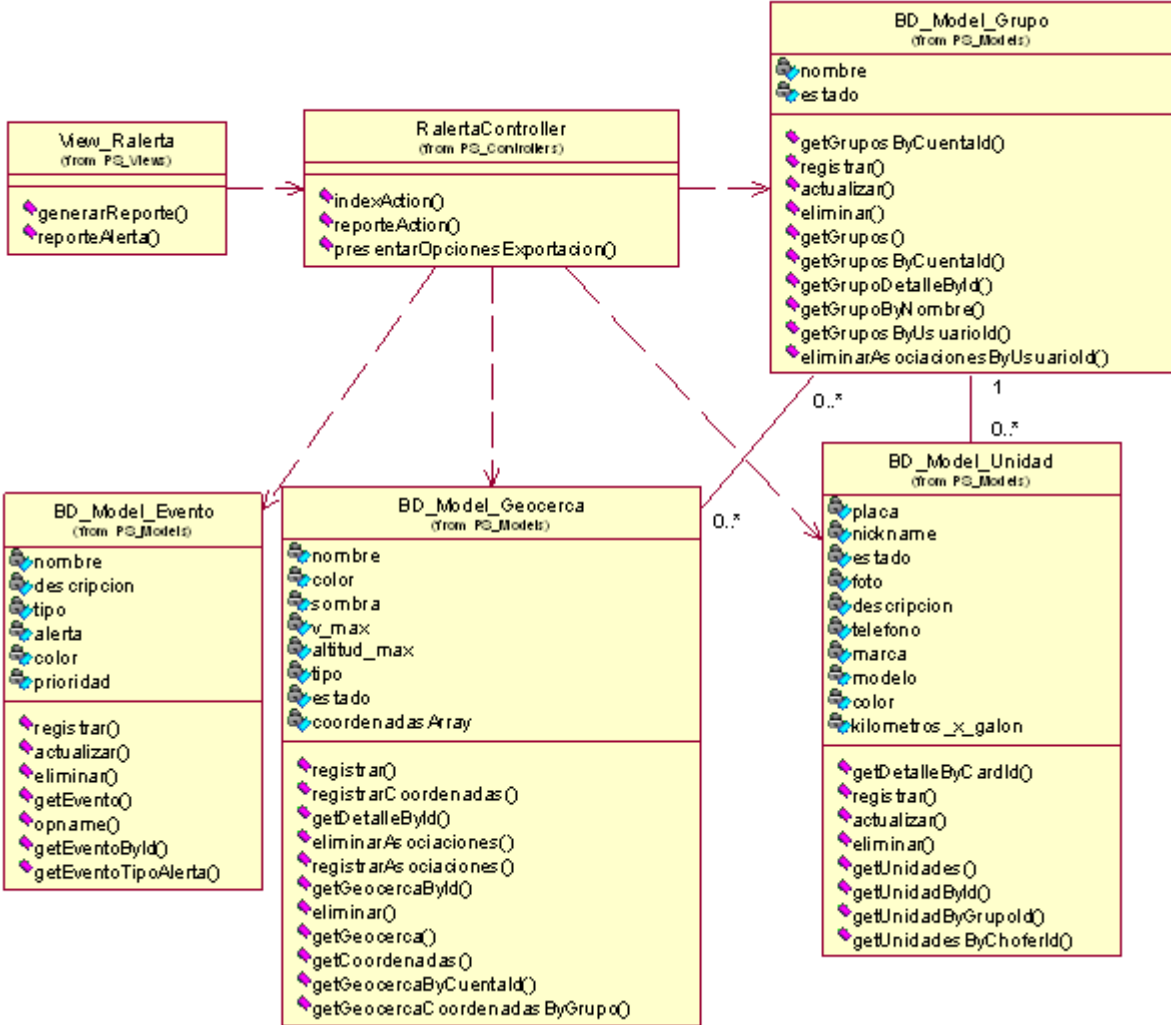


Figura 4.184 Diagrama de clases del caso de uso

## Diagrama de Secuencia – Generar Reporte

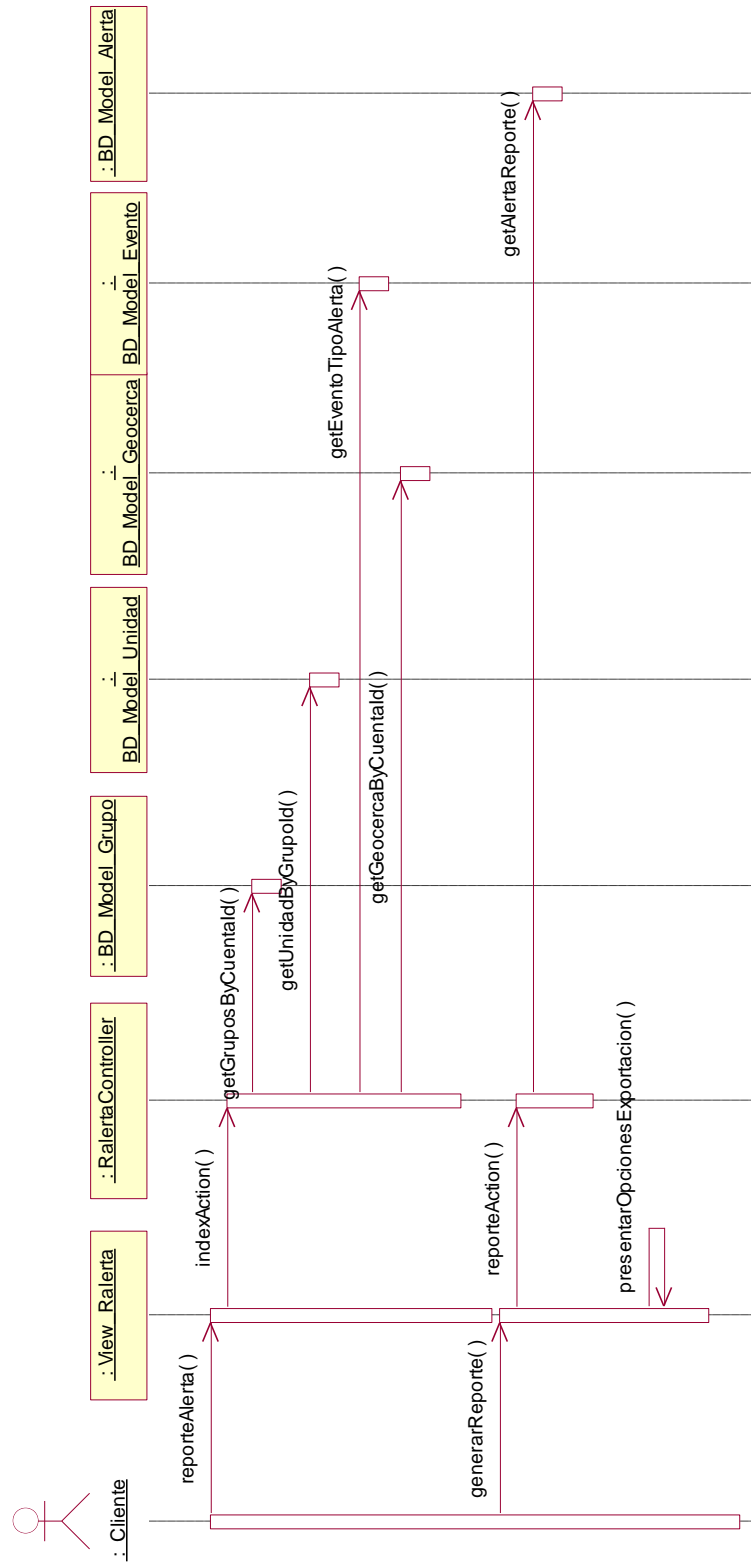


Figura 4.185 Diagrama de secuencia “Generar Reporte”

**Descripción del diagrama de secuencia – Generar Reporte:** El usuario al seleccionar la opción “Alertas” del menú de Reportes, invoca a la clase View\_Ralerta ejecutando la función reporteAlerta, la función reporteAlerta hace una llamada a la función indexAction de la clase RalertaController que se encarga de obtener la lista de flotas de la cuenta a través de la función getGruposByCuentaId de la clase BD\_Model\_Grupo, así mismo obtiene la lista de unidades a través de la función getUnidadByGrupoId de la clase BD\_Model\_Unidad, posteriormente se obtiene la lista de alertas a través de la función getEventoTipoAlerta de la clase BD\_Model\_Evento, luego obtiene las geocercas de la cuenta a través de la función getGeocercaByCuentaId de la clase BD\_Model\_Geocerca, una vez que la función indexAction posee los registros, los entrega a la función reporteAlerta para presentar el formulario de reporte de alertas. El usuario ingresa los datos solicitados y hace clic en el botón Siguiente, al ejecutar el evento se llama a la función generarReporte de la clase View\_Ralerta, la cual llama a la función reporteAction que obtiene los datos del reporte a través de la función getAlertaReporte perteneciente a la clase BD\_Model\_Alerta, posteriormente la clase View\_Ralerta presenta el reporte en formato HTML de forma paginada y presenta las opciones de exportación a través de la función presentarOpcionesExportacion.

### 4.3.8.3 Caso de Uso – Generar Reporte de Comandos

#### Diagrama de Clases

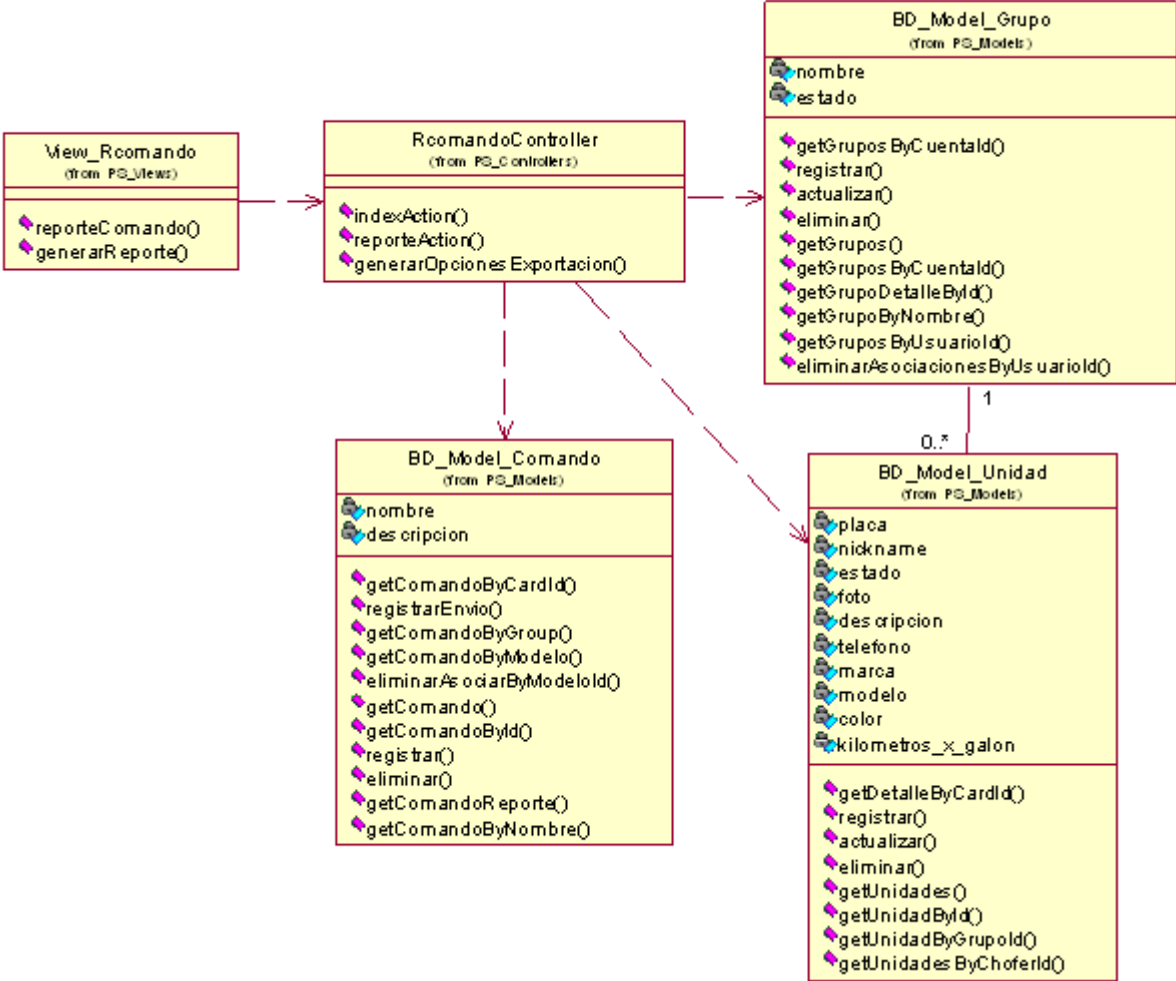


Figura 4.186 Diagrama de clases del caso de uso

## Diagrama de Secuencia – Generar Reporte

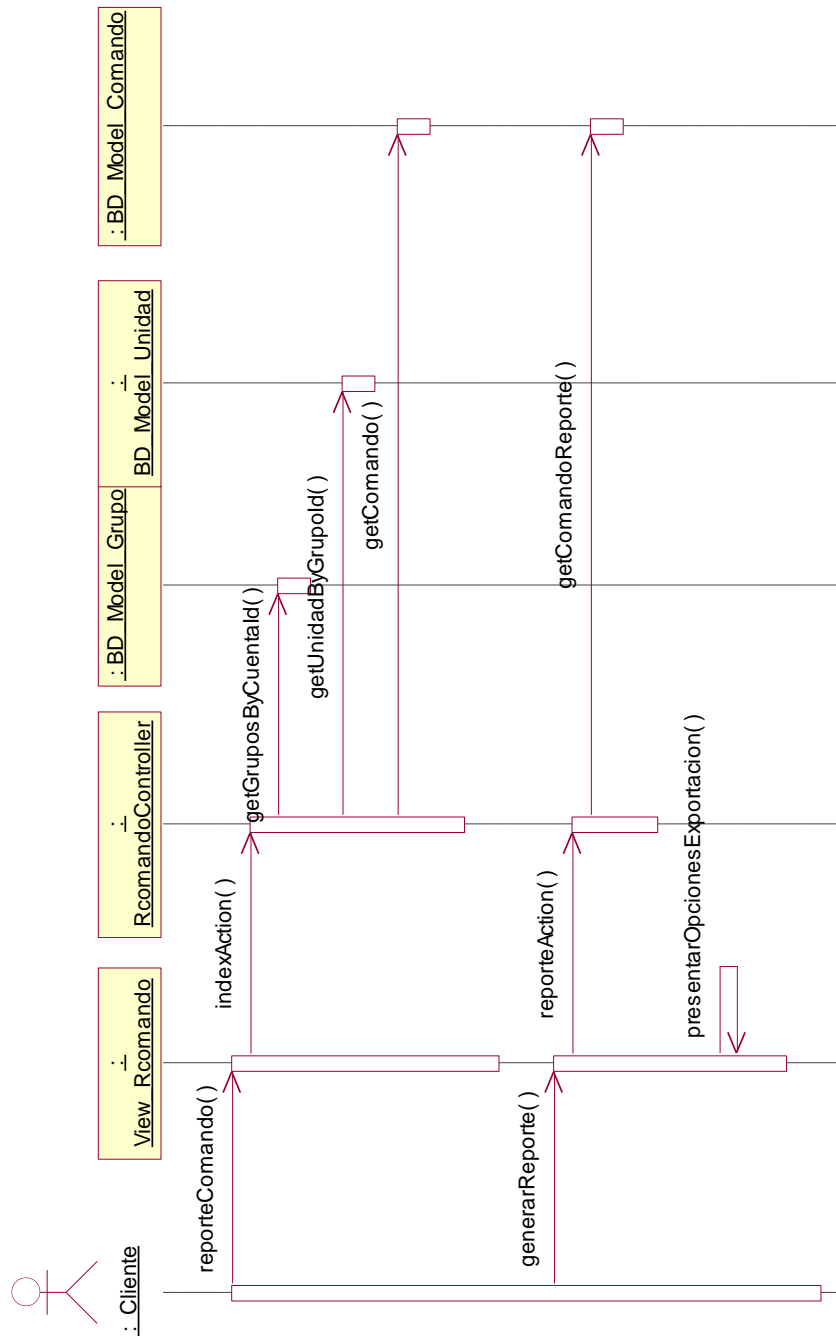


Figura 4.187 Diagrama de secuencia “Generar Reporte”

**Descripción del diagrama de secuencia – Generar Reporte:** El usuario al seleccionar la opción “Comandos” del menú de Reportes, invoca a la clase View\_Rcomando ejecutando la función reporteComando, la función reporteComando hace una llamada a la función indexAction de la clase RcomandoController que se encarga de obtener la lista de flotas de la cuenta a través de la función getGruposByCuentaId de la clase BD\_Model\_Grupo, así mismo obtiene la lista de unidades a través de la función getUnidadByGrupoId de la clase BD\_Model\_Unidad, posteriormente se obtiene la lista comandos a través de la función getComando de la clase BD\_Model\_Comando, una vez que la función indexAction posee los registros, los entrega a la función reporteComando para presentar el formulario de reporte de comandos. El usuario ingresa los datos solicitados y hace clic en el botón Siguiente, al ejecutar el evento se llama a la función generarReporte de la clase View\_Rcomando, la cual llama a la función reporteAction que obtiene los datos del reporte a través de la función getComandoReporte perteneciente a la clase BD\_Model\_Comando, posteriormente la clase View\_Rcomando presenta el reporte en formato HTML de forma paginada y presenta las opciones de exportación a través de la función presentarOpcionesExportacion.

## 4.3.9 Realización de casos de uso de diseño – Paquete Seguridad

### 4.3.9.1 Caso de Uso – Actualizar Cuenta

#### Diagrama de Clases

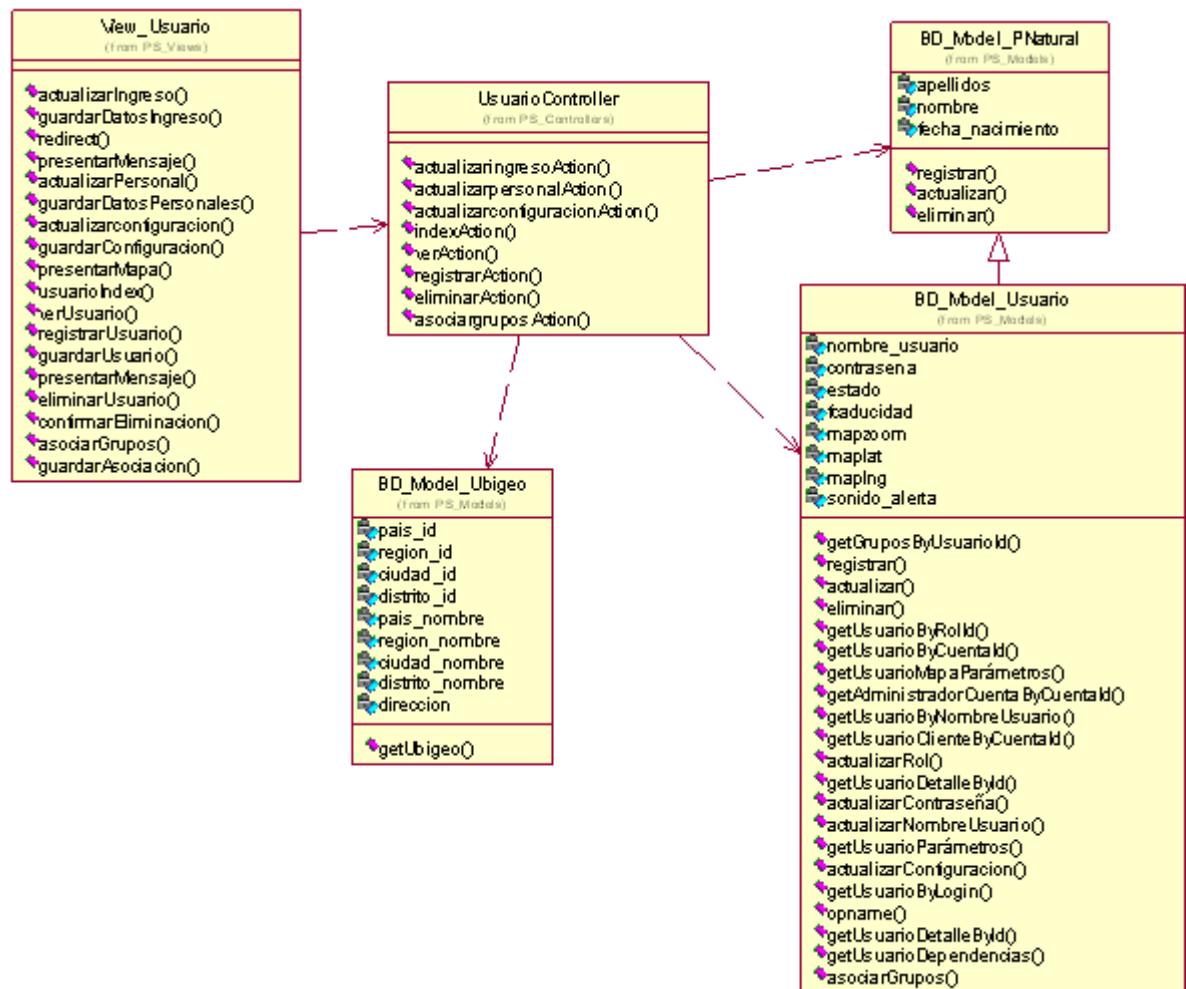


Figura 4.188 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Actualizar Ingreso al Sistema

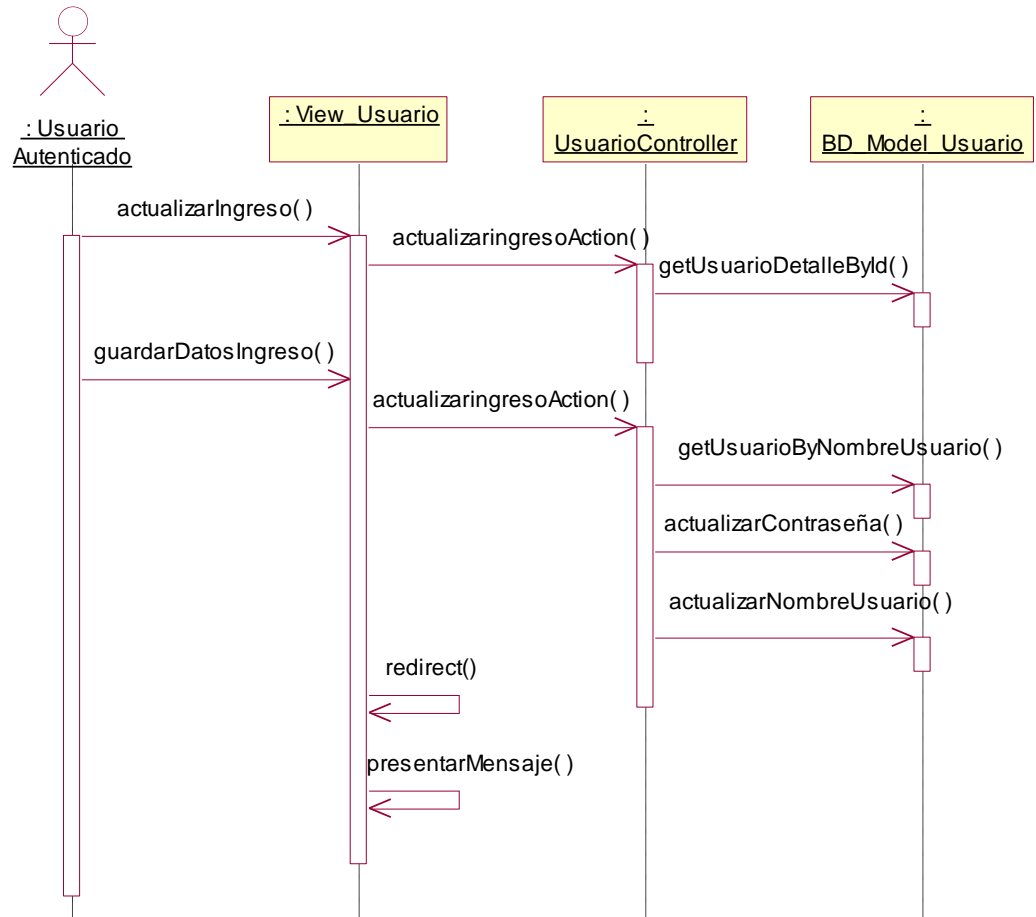


Figura 4.189 Diagrama de secuencia “Actualizar Ingreso al Sistema”

**Descripción del diagrama de secuencia – Actualizar Ingreso al Sistema:** El usuario al seleccionar la opción “Cuenta” del menú de Seguridad, invoca a la clase View\_Usuario que presenta la opción de “Actualizar Ingreso al Sistema”, el usuario al hacer clic en dicha opción la función actualizarIngreso de la clase View\_Usuario es llamada, esta función invoca a la función actualizarIngresoAction de la clase UsuarioController la cual obtiene los datos de ingreso del usuario a través de la función getUsuarioDetalleById de la clase BD\_Model\_Usuario, una vez que la función actualizarIngreso tiene los datos, la función actualizarIngreso puede presentar el formulario correspondiente. El usuario ingresa los datos de acceso al formulario y hace clic en el botón “Actualizar”, este evento ejecuta la función guardarDatosIngreso de la clase View\_Usuario, la función guardarDatosIngreso llama a la función actualizarIngresoAction de la clase UsuarioController que verifica que el nombre de usuario no se repita a través de la función getUsuarioByNombreUsuario de la clase BD\_Model\_Usuario, actualiza la contraseña a través de la función actualizarContraseña de la misma clase y si se da el caso actualiza el nombre del usuario a través de la función



actualizarNombreUsuario de la clase BD\_Model\_Usuario, posteriormente la clase View\_Usuario redirige la página a fin de visualizar el detalle del usuario con la función redirect y presenta un mensaje de éxito a través de la función presentarMensaje.

### Diagrama de Secuencia – Actualizar datos personales

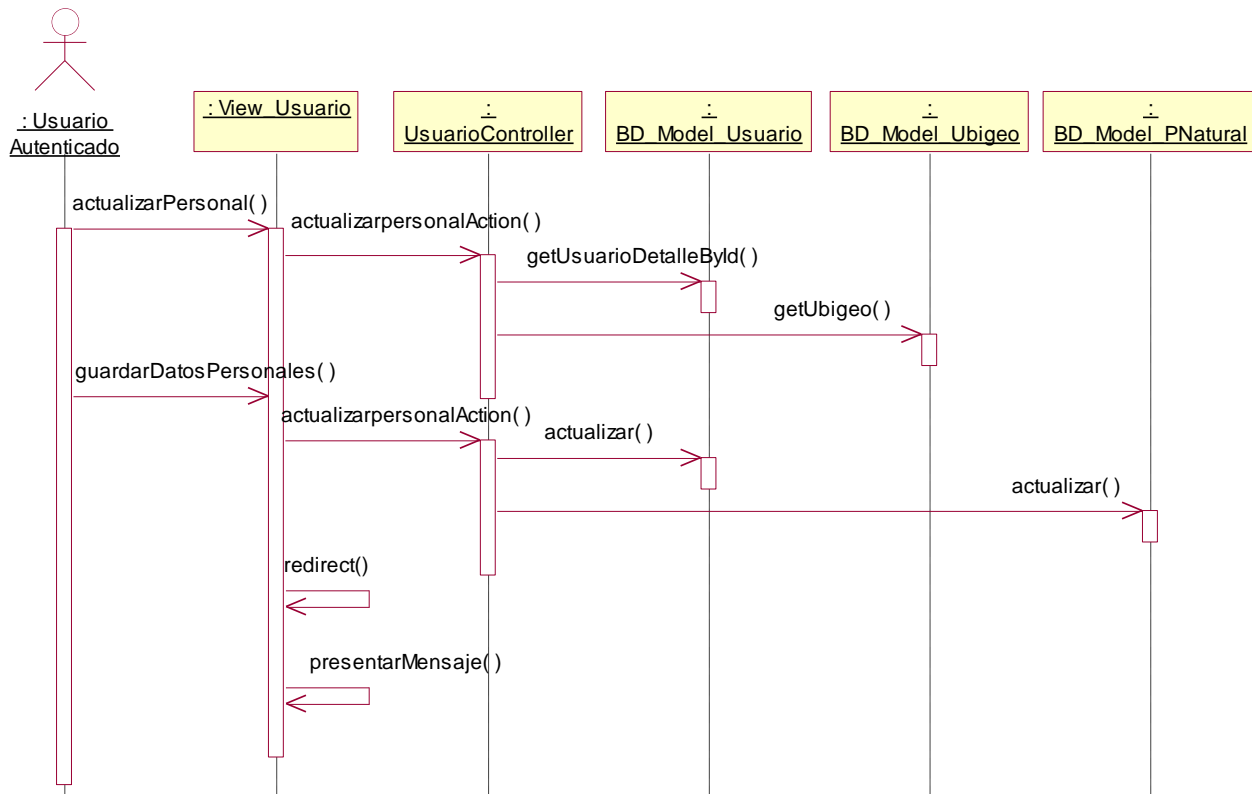


Figura 4.190 Diagrama de secuencia “Actualizar datos personales”

**Descripción del diagrama de secuencia – Actualizar datos personales:** El usuario al seleccionar la opción “Cuenta” del menú de Seguridad, invoca a la clase View\_Usuario que presenta la opción de “Actualizar datos personales”, el usuario al hacer clic en dicha opción la función actualizarPersonal de la clase View\_Usuario es llamada, esta función invoca a la función actualizarpersonalAction de la clase UsuarioController la cual obtiene el detalle del usuario a través de la función getUsuarioDetalleById de la clase BD\_Model\_Usuario y la lista de regiones, ciudades y distritos a través de la función getUbigeo de la clase BD\_Model\_Ubigeo, una vez que la función actualizarpersonalAction tiene los datos, la función actualizarPersonal puede presentar el formulario correspondiente. El usuario ingresa los datos de acceso al formulario y hace clic en el botón “Actualizar”, este evento ejecuta la función guardarDatosPersonales de la clase View\_Usuario, la función guardarDatosPersonales llama a la función actualizarpersonalAction de la clase

UsuarioController que actualiza los datos de la persona natural a través de la función actualizar de la clase BD\_Model\_Pnatural y actualiza los datos del usuario a través de la función actualizar de la clase BD\_Model\_Usuario, posteriormente la clase View\_Usuario redirige la página a fin de visualizar el detalle del usuario con la función redirect y presenta un mensaje de éxito a través de la función presentarMensaje.

### Diagrama de Secuencia – Actualizar Configuración del Mapa

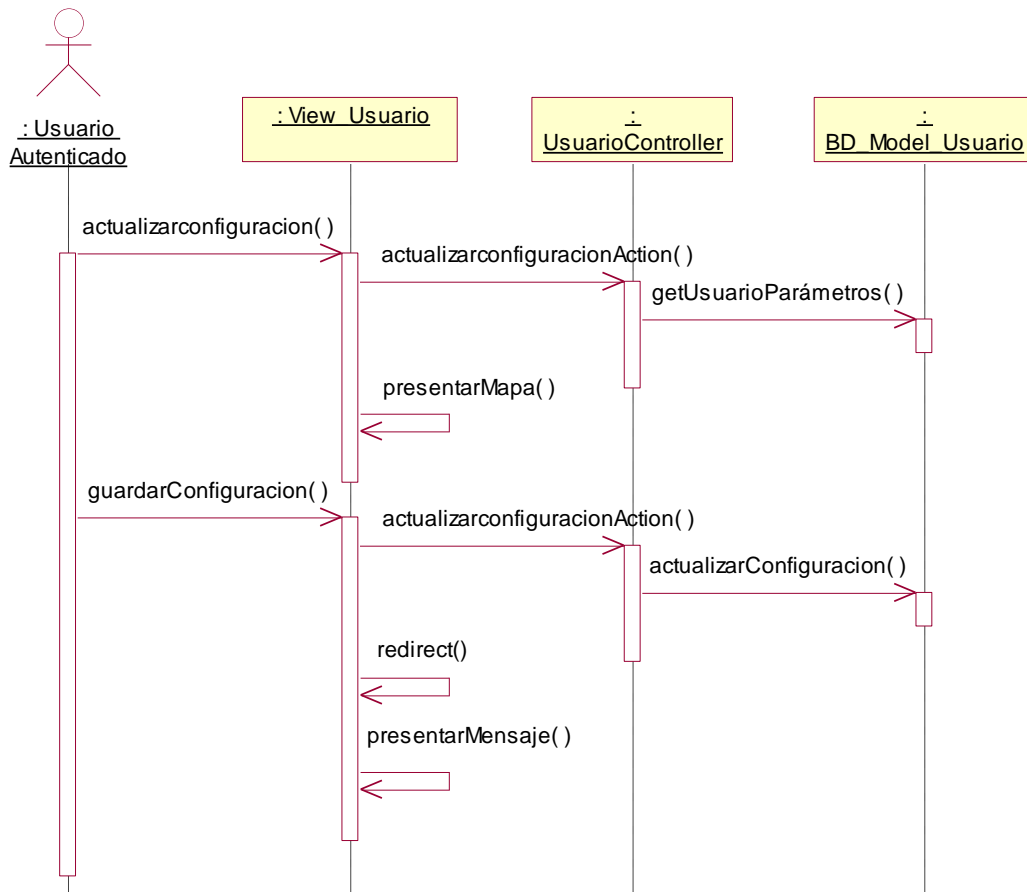


Figura 4.191 Diagrama de secuencia “Actualizar Configuración del Mapa”

#### Descripción del diagrama de secuencia – Actualizar Configuración del Mapa:

El usuario al seleccionar la opción “Cuenta” del menú de Seguridad, invoca a la clase View\_Usuario que presenta la opción de “Actualizar configuración del Mapa”, el usuario al hacer clic en dicha opción la función actualizarconfiguracion de la clase View\_Usuario es llamada, esta función invoca a la función actualizarconfiguracionAction de la clase UsuarioController la cual obtiene el detalle del de los parámetros del usuario a través de la función getUsuarioParametros de la clase BD\_Model\_Usuario, una vez que la función

actualizarconfiguracionAction tiene los datos, la función actualizarconfiguracion puede presentar el formulario correspondiente y presentar un mapa donde el usuario puede escoger una posición inicial, esto lo hace a través de la función presentarMapa. El usuario ingresa los datos de configuración al formulario, selecciona la posición inicial en el mapa y hace clic en el botón “Actualizar”, este evento ejecuta la función guardarConfiguracion de la clase View\_Usuario, la función guardarConfiguracion llama a la función actualizarconfiguracionAction de la clase UsuarioController que actualiza los datos de configuración del usuario a través de la función actualizarConfiguracion de la clase BD\_Model\_Usuario, posteriormente la clase View\_Usuario redirige la página a fin de visualizar el detalle del usuario con la función redirect y presenta un mensaje de éxito a través de la función presentarMensaje.

#### 4.3.9.2 Caso de Uso – Ingresar al Sistema

##### Diagrama de Clases

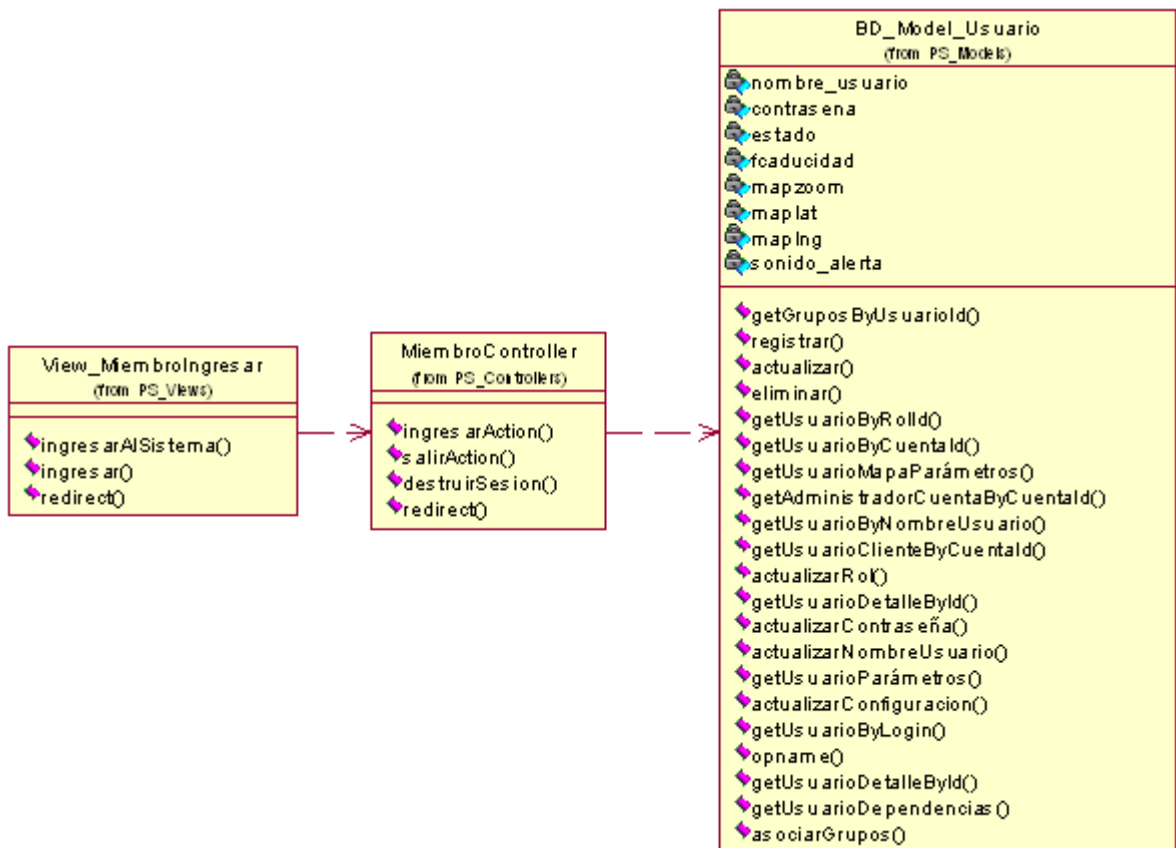


Figura 4.191 Diagrama de clases del caso de uso

## Diagrama de Secuencia – Ingresar al Sistema

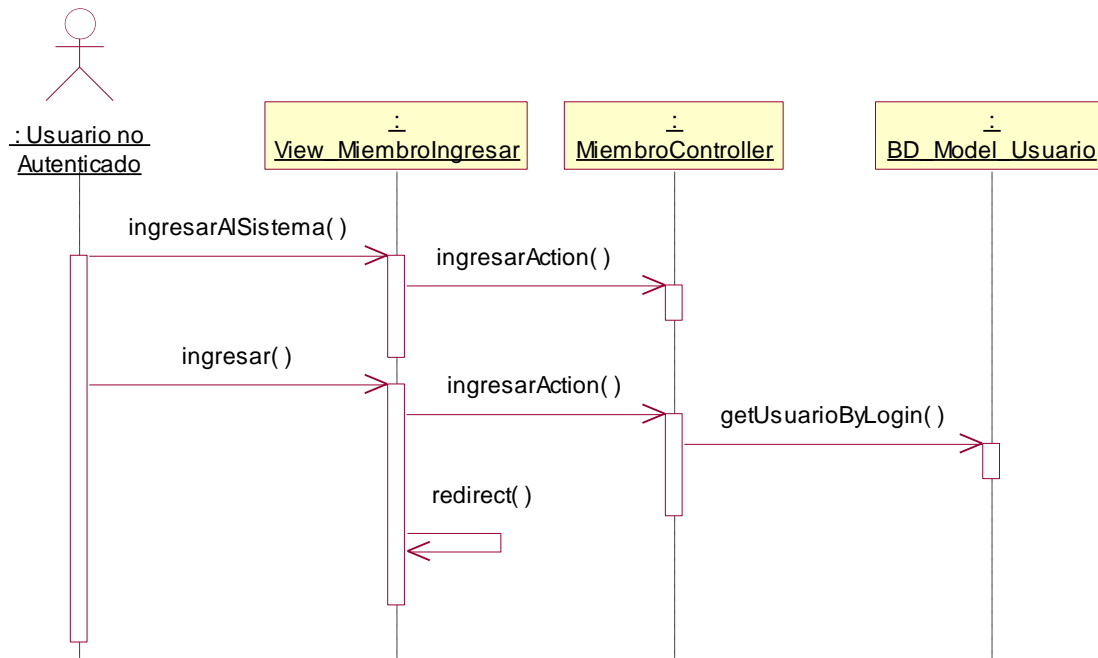


Figura 4.193 Diagrama de secuencia “Ingresar al Sistema”

**Descripción del diagrama de secuencia – Ingresar al Sistema:** El usuario al seleccionar la opción “acceder” de la ventana de inicio, ejecuta el evento que llama a la función `ingresarAlSistema` de la clase `View_MiembroIngresar`, esta función invoca a la función `ingresarAction` de la clase `MiembroController` que permite que la clase `View_MiembroIngresar` presente el formulario de ingreso al sistema, el usuario ingresa su nombre de usuario, contraseña y hace clic en el botón “Acceder” el cual ejecuta la función `ingresar` de la clase `View_MiembroIngresar`, esta función llama a la función `ingresarAction` de la clase `MiembroController` y valida los datos ingresados a través de la función `getUsuarioByLogin` de la clase `BD_Model_Usuario`, si los datos son correctos, la función `ingresarAction` invitará a la clase `View_MiembroIngresar` redirigir la pantalla de acceso a la pantalla de bienvenida a través de la función `redirect`.

### 4.3.9.3 Caso de Uso – Recuperar Contraseña

#### Diagrama de Clases

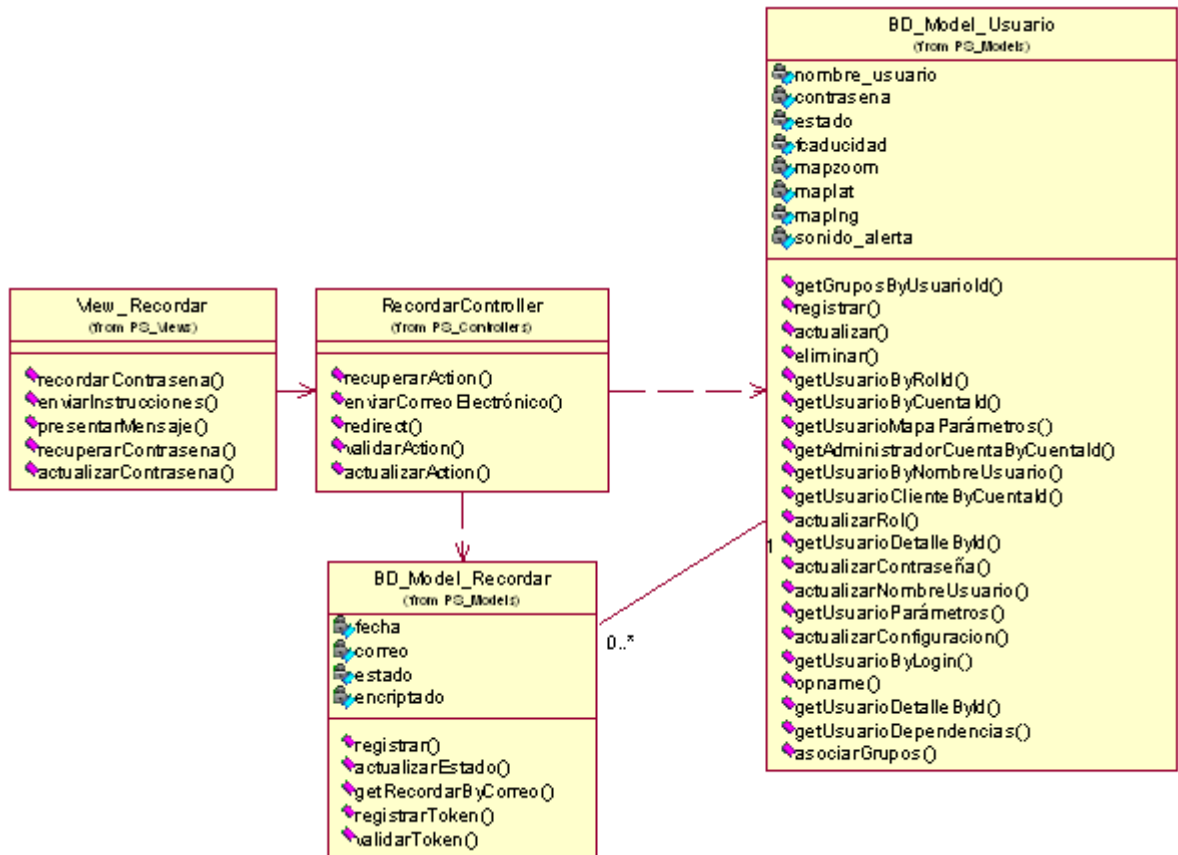


Figura 4.194 Diagrama de clases del caso de uso

## Diagrama de Secuencia – Enviar Instrucciones

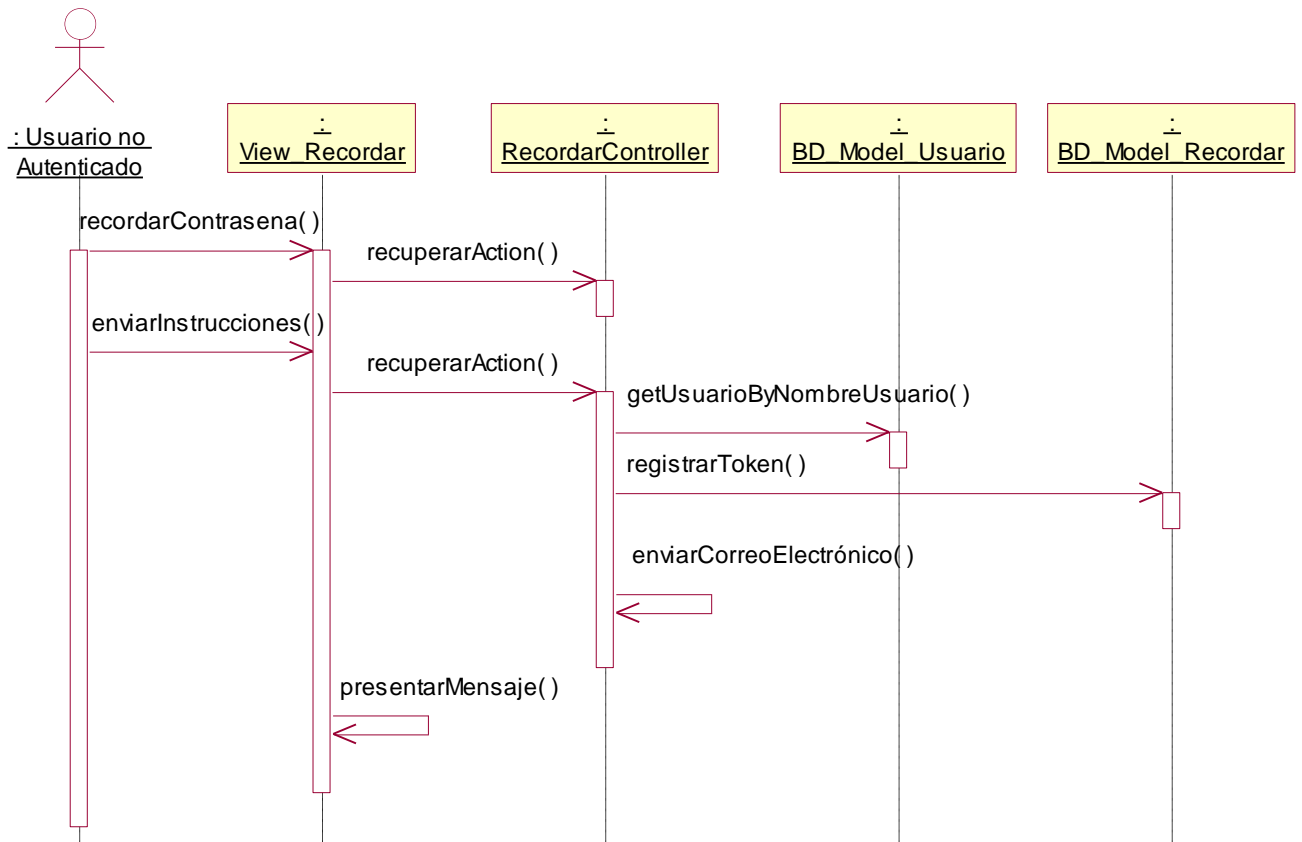


Figura 4.195 Diagrama de secuencia “Enviar Instrucciones”

**Descripción del diagrama de secuencia – Enviar Instrucciones:** El usuario al seleccionar la opción “Recordar Contraseña” de la ventana de acceso al sistema, ejecuta el evento que llama a la función `recordarContrasena()` de la clase `View_Recordar`, esta función invoca a la función `recuperarAction()` de la clase `RecordarController` que permite que la clase `View_Recordar` presente el formulario de envío de instrucciones de recuperación de contraseña, el usuario ingresa su nombre de usuario y hace clic en el botón “Enviar Instrucciones” el cual ejecuta la función `enviarInstrucciones()` de la clase `View_Recordar`, esta función llama a la función `recuperarAction()` de la clase `RecordarController` y valida el nombre de usuario a través de la función `getUsuarioByNombreUsuario()` de la clase `BD_Model_Usuario`, si los datos son correctos, registra el token de recuperación a través de la función `registrarToken()` de la clase `BD_Model_Recordar` y envía un correo electrónico a la cuenta de correo electrónico del usuario a través de la función `enviarCorreoElectrónico()` de la clase `RecordarController`, posteriormente la clase `View_Recordar` presenta un mensaje indicando que las instrucciones fueron enviadas a su cuenta de correo electrónico a través de la función `presentarMensaje()`.

## Diagrama de Secuencia – Cambiar Contraseña

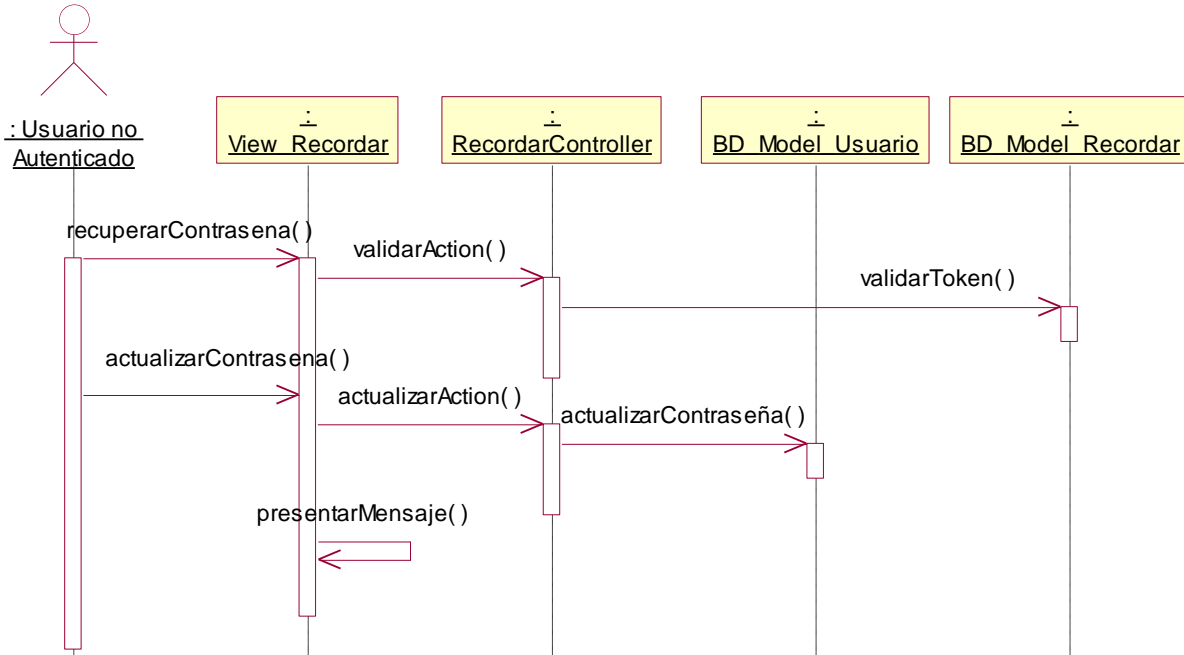


Figura 4.196 Diagrama de secuencia “Cambiar Contraseña”

**Descripción del diagrama de secuencia – Cambiar Contraseña:** El usuario al abrir el mensaje enviado por el sistema, según las instrucciones hace clic en el enlace sugerido, este enlace llama a la función recuperarContrasena de la clase View\_Recordar, la función recuperarContrasena invoca a la función validarAction de la clase RecordarController a fin de validar el token que llegó como parámetro URL, el token es validado a través de la función validarToken de la clase BD\_Model\_Recordar. Si el token es correcto, este es activado en la misma función validarToken, ya la clase View\_Recordar presenta el formulario de actualización de contraseña. El usuario ingresa la nueva contraseña y su confirmación y hace clic en el botón “Cambiar Contraseña”, al hacer clic en dicho botón la función actualizarContraseña es invocada, la cual llama a la función actualizarAction de la clase RecordarController y actualiza la contraseña con la función actualizarContraseña de la clase BD\_Model\_Usuario, posteriormente la clase View\_Recordar presenta un mensaje indicando el éxito de la operación a través de la función presentarMensaje.

#### 4.3.9.4 Caso de Uso – Salir del sistema

##### Diagrama de Clases

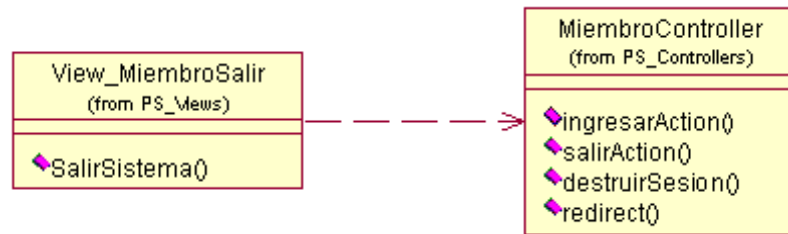


Figura 4.197 Diagrama de clases del caso de uso

##### Diagrama de Secuencia – Salir del Sistema

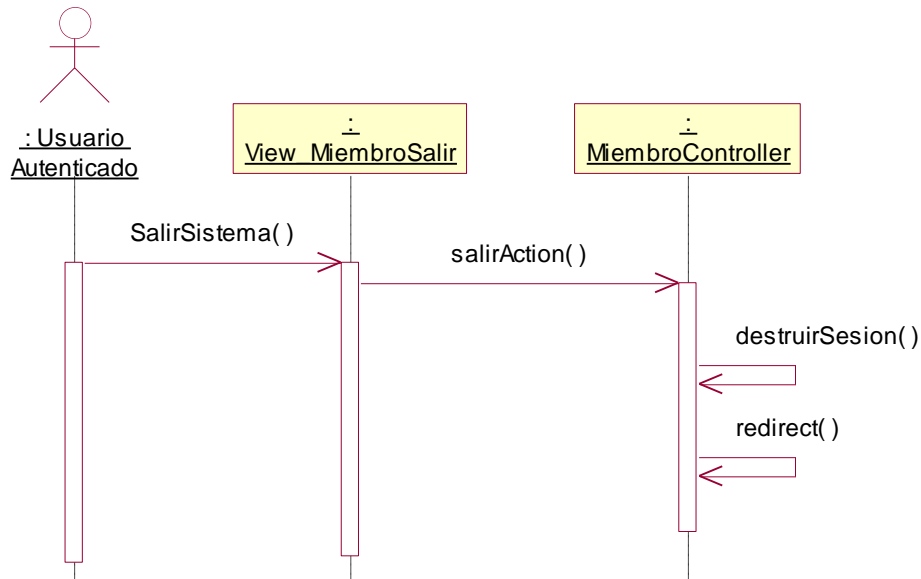


Figura 4.198 Diagrama de secuencia “Salir del Sistema”

**Descripción del diagrama de secuencia – Salir del Sistema:** El usuario al seleccionar la opción “salir del sistema”, ejecuta el evento que llama a la función `SalirSistema` de la clase `View_MiembroSalir`, esta función invoca a la función `salirAction` de la clase `MiembroController`, que destruye la sesión con la función `destruirSesion` y re direcciona al formulario de acceso al sistema a través de la función `redirect`.



### 4.3.9.5 Caso de Uso – Mantenimiento de usuarios

#### Diagrama de Clases

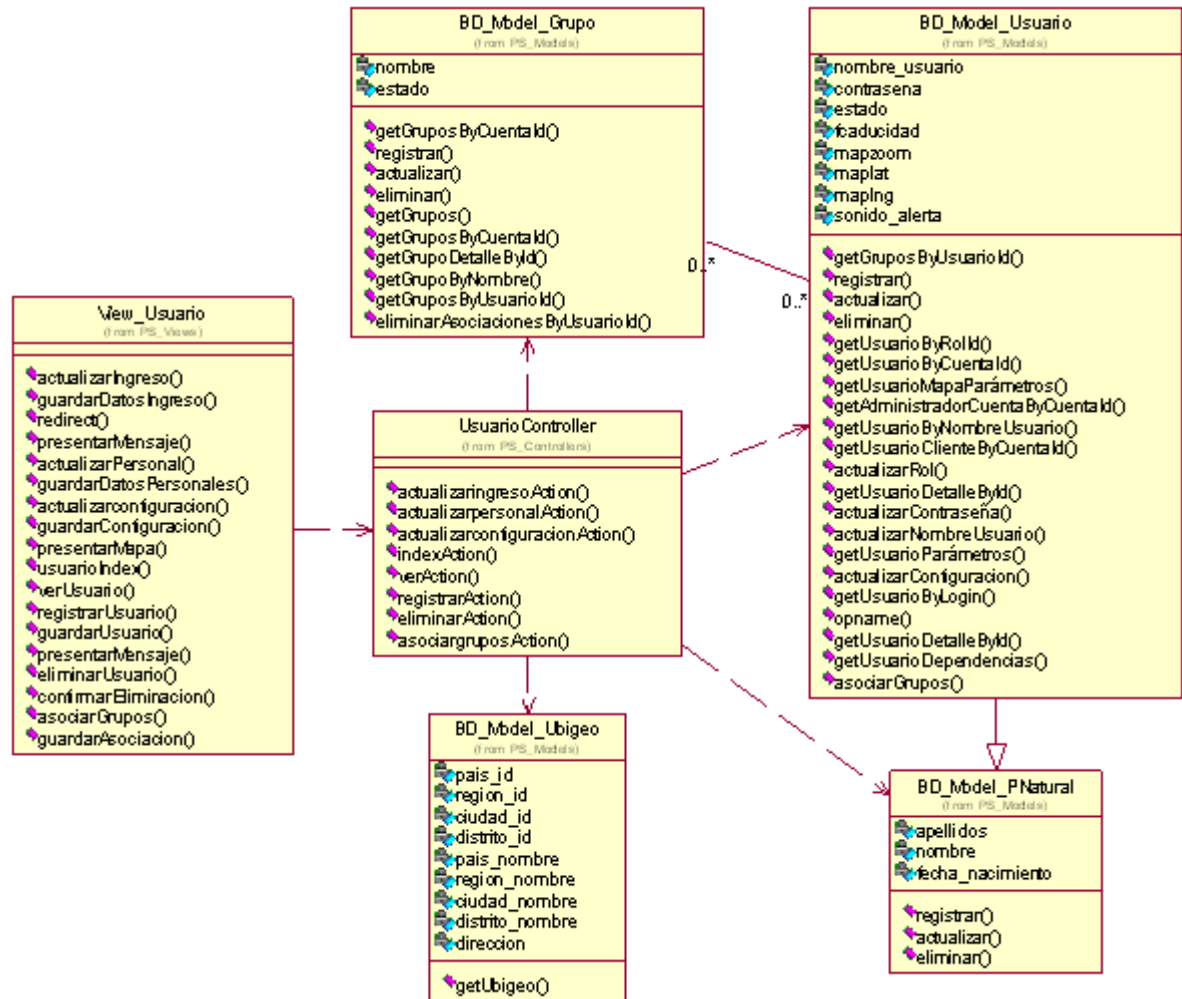


Figura 4.199 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

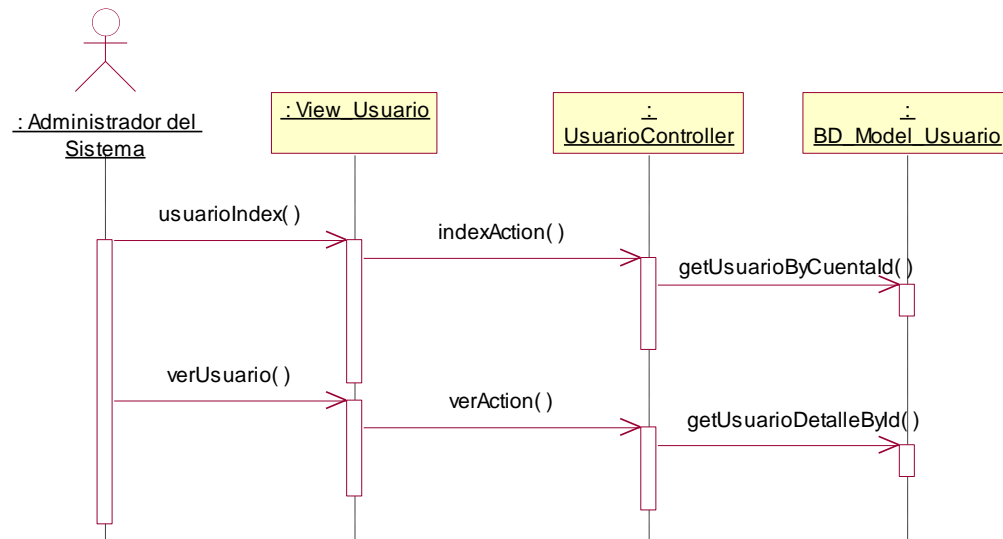


Figura 4.200 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Usuarios” del menú de Seguridad, este evento invoca a la clase View\_Usuario ejecutando la función usuarioIndex, la función usuarioIndex hace una llamada a la función indexAction de la clase UsuarioController que se encarga de obtener la lista de usuarios registrados en la cuenta cliente a través de la función getUsuarioByCuentaId de la clase BD\_Model\_Usuario, una vez que la función indexAction posee los registros, los entrega a la función usuarioIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verUsurio de la clase View\_Usuario, el cual hace una llamada a la función verAction de la clase UsuarioController, esta función realiza una llamada a la función getUsuarioDetalleById de la clase BD\_Model\_Usuario que devuelve el detalle del usuario por el id del usuario, una vez que la función posee el detalle del usuario, la información es presentada a través de la clase View\_Usuario.

## Diagrama de Secuencia – Registrar

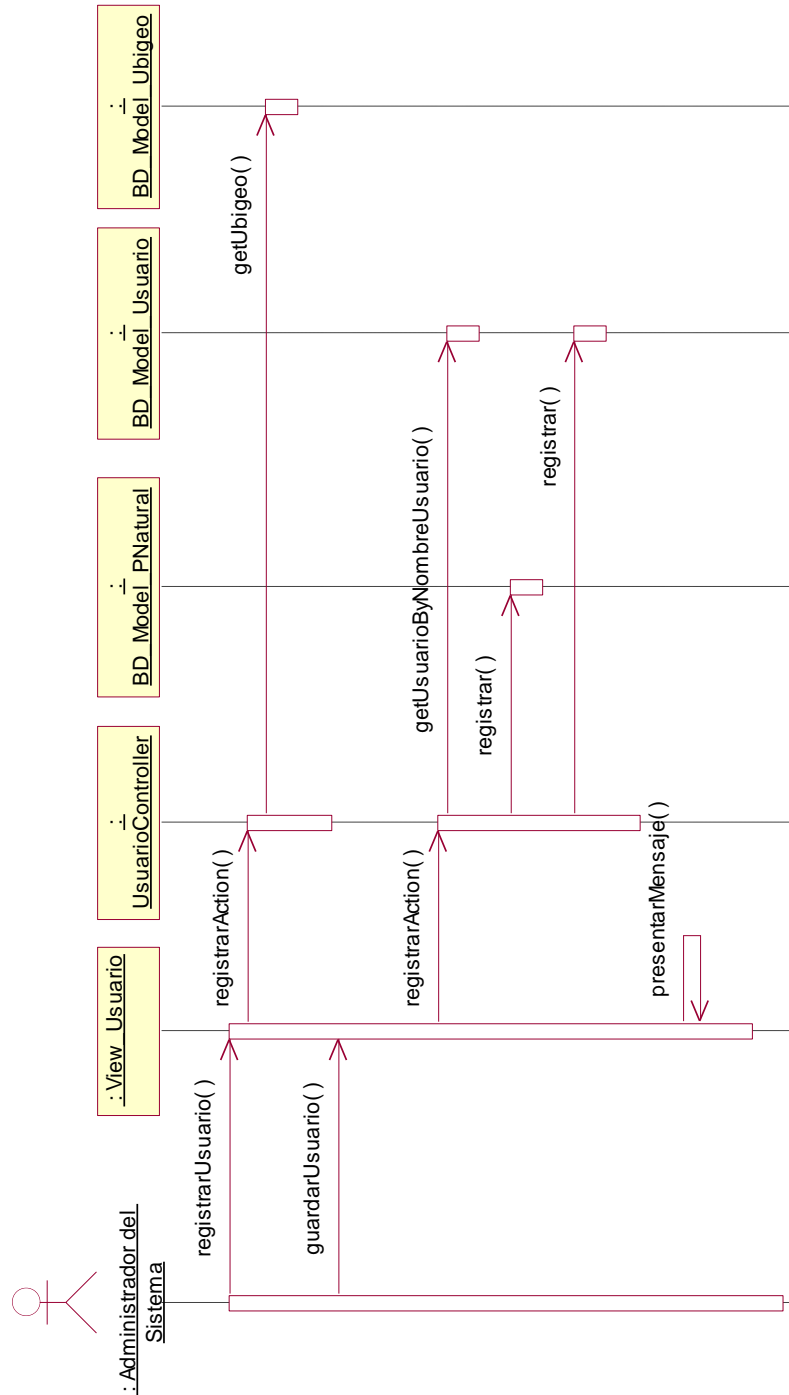


Figura 4.201 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de usuarios, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarUsuario de la clase View\_Usuario, la función registrarUsuario hace una

llamada a la función registrarAction de la clase UsuarioController que solicita la lista de departamentos, ciudades y distritos a la clase BD\_Model\_Ubigeo a través de la función getUbigeo, posteriormente invita a la función registrarUsuario presentar el formulario de registro de usuario. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarUsuario de la clase View\_Usuario, la función hace una llamada a la función registrarAction de la clase UsuarioController verifica que el nombre de usuario sea único, por eso llama a la función getUsuarioByNombreUsuario de la clase BD\_Model\_Usuario, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Usuario, y también se ejecuta la función registrar de la clase BD\_Model\_Pnatural, posteriormente la clase View\_Usuario puede presentar un mensaje indicando que la operación tuvo éxito.

### Diagrama de Secuencia – Eliminar

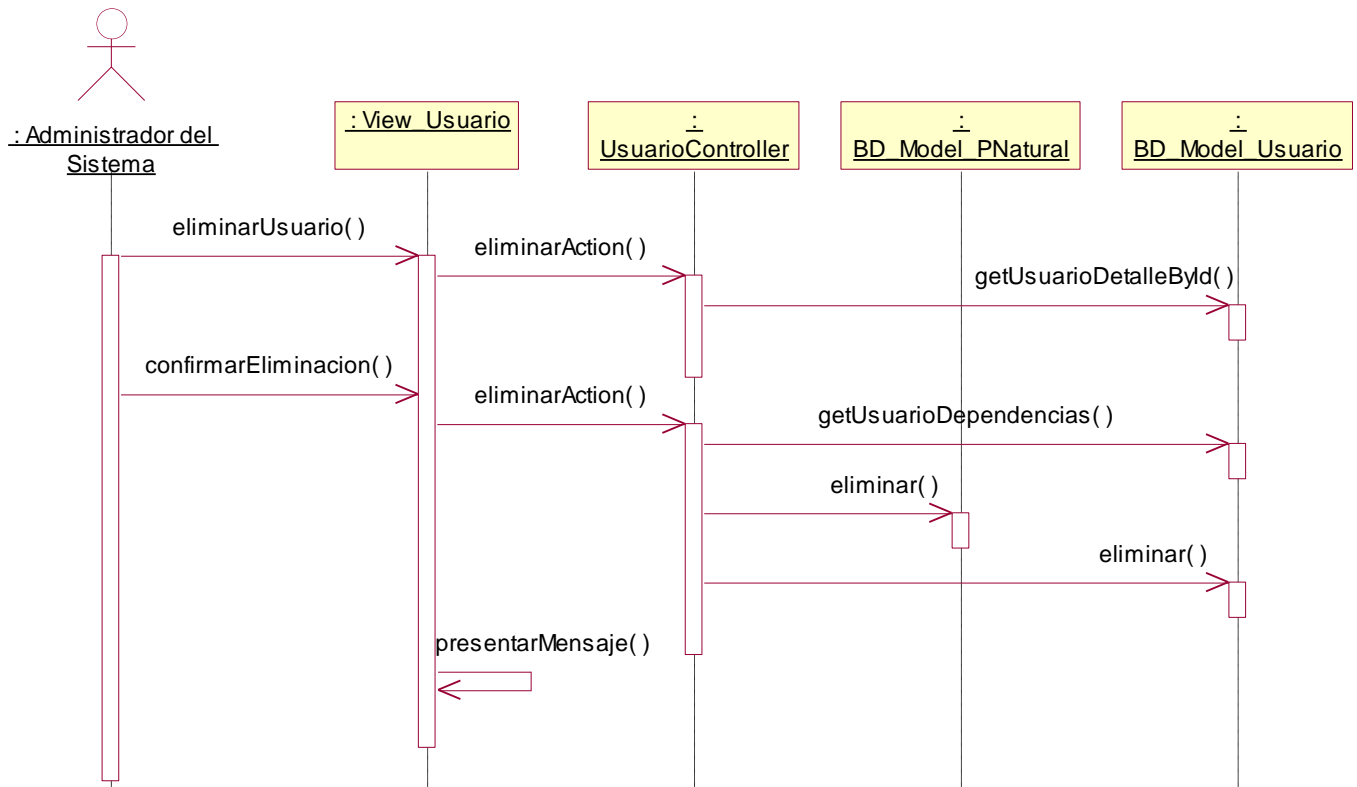


Figura 4.202 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de usuario, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarUsuario de la clase View\_Usuario, la función eliminarUsuario hace una llamada a la función eliminarAction de la clase UsuarioController el cual invoca a la función getUsuarioDetalleById de la clase BD\_Model\_Usuario a fin de obtener el

detalle del usuario a eliminar, una vez que la función eliminarAction tiene los datos, los pasa a la función eliminarUsuario y estos son presentados al usuario. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Usuario el cual invoca nuevamente a la función eliminarAction de la clase UsuarioController la cual primero verifica que el usuario no tenga registros asociados a través de la función getUsuarioDependencias de la clase BD\_Model\_Usuario, esta función se encarga de verificar que el usuario no haya atendido alertas, enviado comandos, activado equipo o creado otros usuarios, una vez que la función eliminarAction confirma que ese usuario no tiene registros asociados, ejecuta la función eliminar de la clase BD\_Model\_Usuario y la función eliminar de la clase BD\_Model\_Pnatural, y luego la clase View\_Usuario a través de la función presentarMensaje indica que la operación tuvo éxito.

### Diagrama de Secuencia – Asociar Flotas

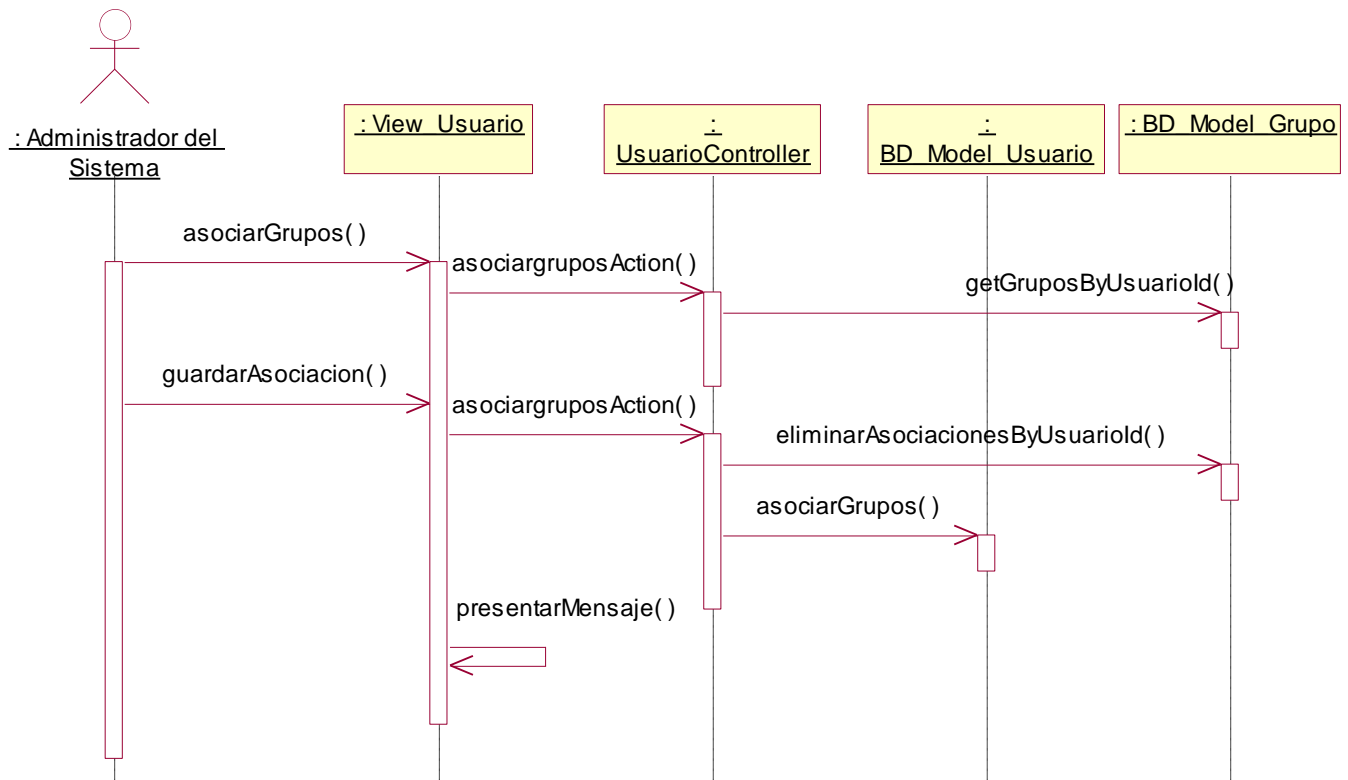


Figura 4.203 Diagrama de secuencia “Asociar Flotas”

**Descripción del diagrama de secuencia – Asociar Flotas:** El usuario en la lista de usuarios, hace clic en la opción “Flotas”, este evento ejecuta la opción asociarGrupos de la clase View\_Usuario, la función asociarGrupos hace una llamada a la función asociargruposAction de la clase UsuarioController el cual invoca a la función getGruposByUsuarioId de la cuenta BD\_Model\_Grupo a fin de obtener a las flotas que tiene asociada el usuario, una vez que la función

asociargruposAction tiene los datos, los pasa a la función asociarGrupos y estos son presentados al usuario. El usuario selecciona las flotas y confirma la asociación haciendo clic en el botón “Actualizar”, este evento ejecuta la función guardarAsociacion de la clase View\_Usuario el cual invoca nuevamente a la función asociargruposAction de la clase UsuarioController la cual primero elimina las asociaciones anteriores a través de la función eliminarAsociacionesByUsuarioId de la clase BD\_Model\_Grupo, luego ejecuta la función asociarGrupos de la clase BD\_Model\_Usuario a fin de guardar las nuevas flotas, y luego la clase View\_Usuario a través de la función presentarMensaje indica que la operación tuvo éxito.

### 4.3.9.6 Caso de Uso – Mantenimiento de Roles

#### Diagrama de Clases

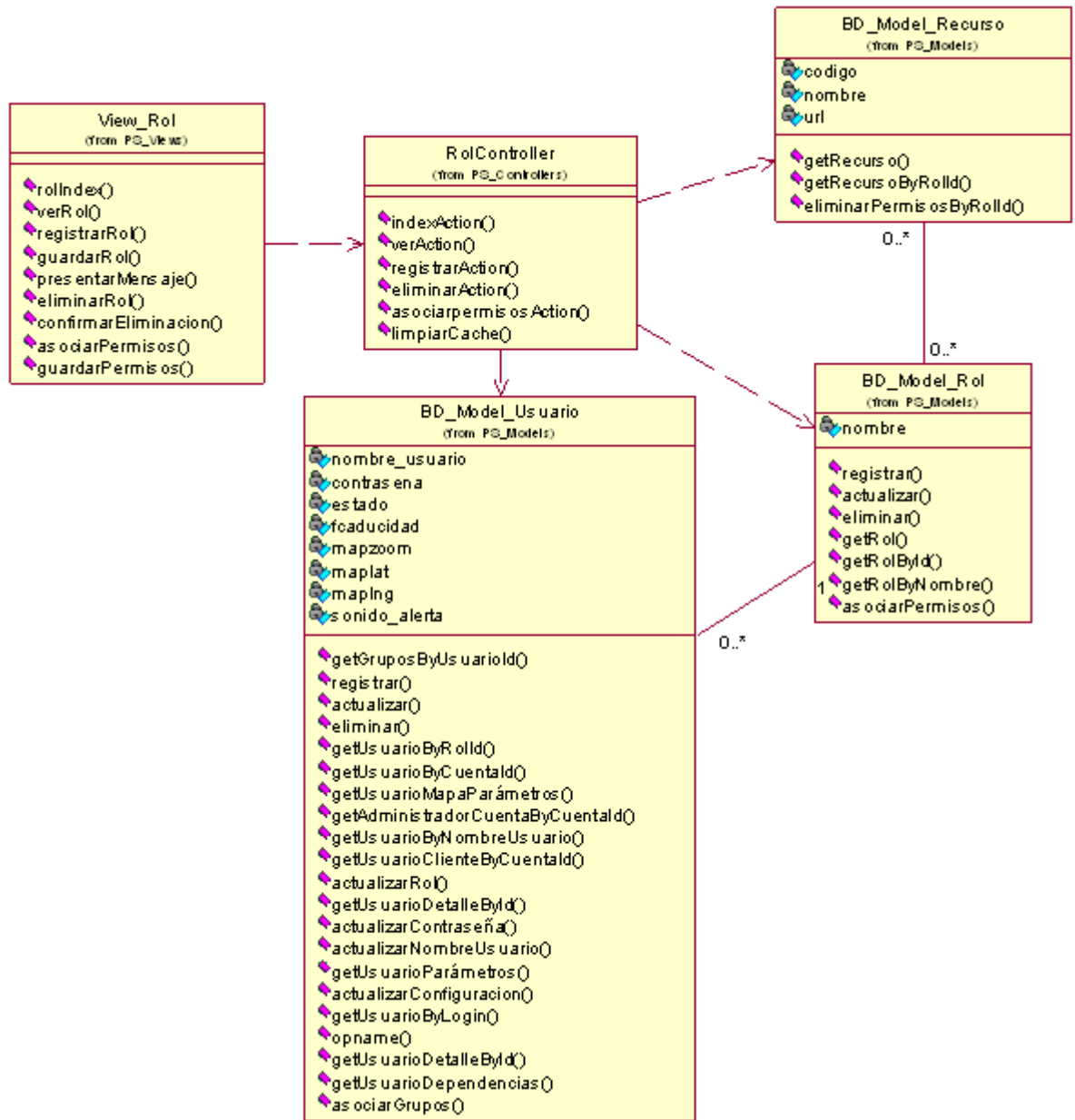


Figura 4.204 Diagrama de clases del caso de uso

## Diagrama de Secuencia –Listar

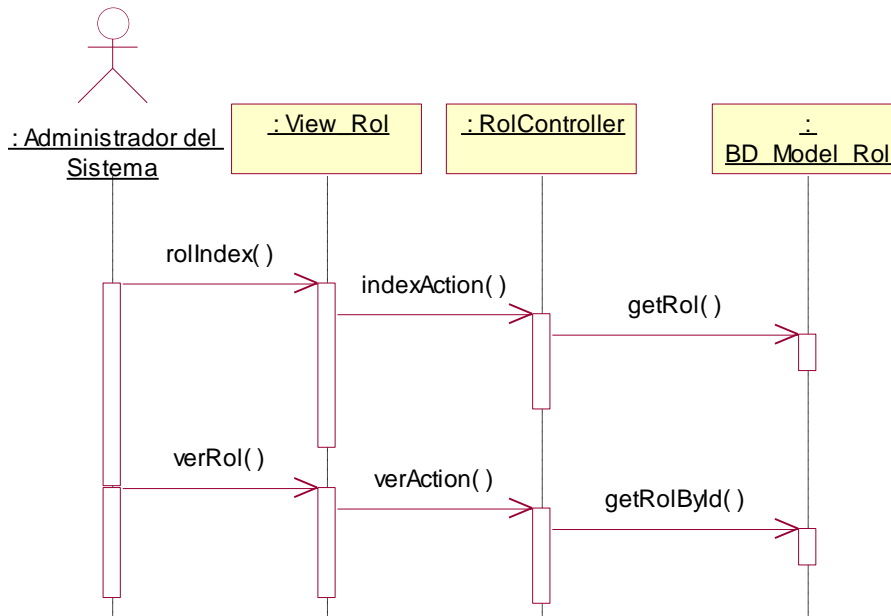


Figura 4.205 Diagrama de secuencia “Listar”

**Descripción del diagrama de secuencia – Listar:** El usuario al seleccionar la opción “Roles” del menú de Seguridad, este evento invoca a la clase View\_Rol ejecutando la función rolIndex, la función rolIndex hace una llamada a la función indexAction de la clase RolController que se encarga de obtener la lista de roles registrados en el sistema a través de la función getRol de la clase BD\_Model\_Rol, una vez que la función indexAction posee los registros, los entrega a la función rolIndex de forma paginada. El usuario cuando hace clic en la opción ver de un registro presentado, ejecuta el evento verRol de la clase View\_Rol, el cual hace una llamada a la función verAction de la clase RolController, esta función realiza una llamada a la función getRolById de la clase BD\_Model\_Rol que devuelve el detalle del rol por el id del rol, una vez que la función posee el detalle del rol, la información es presentada a través de la clase View\_Rol.



## Diagrama de Secuencia – Registrar

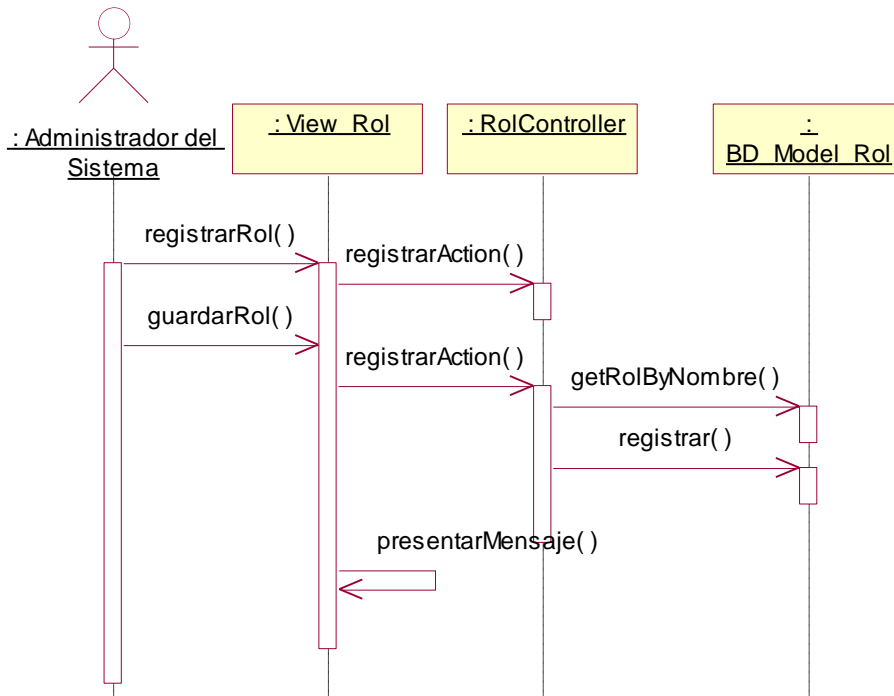


Figura 4.206 Diagrama de secuencia “Registrar”

**Descripción del diagrama de secuencia – Registrar:** El usuario en la lista de roles, hace clic en la opción “Registrar”, este evento ejecuta la opción registrarRol de la clase View\_Rol, la función registrarRol hace una llamada a la función registrarAction de la clase RolController, posteriormente invita a la función registrarRol presentar el formulario de registro de roles. El usuario ingresa los datos solicitados y hace clic en el botón “Registrar”, este evento ejecuta la función guardarRol de la clase View\_Rol, la función hace una llamada a la función registrarAction de la clase RolController verifica que el nombre del rol sea único, por eso llama a la función getRolByNombre de la clase BD\_Model\_Rol, y si cumple la condición se ejecuta la función registrar de la clase BD\_Model\_Rol, posteriormente la clase View\_Rol puede presentar un mensaje indicando que la operación tuvo éxito.

## Diagrama de Secuencia – Eliminar

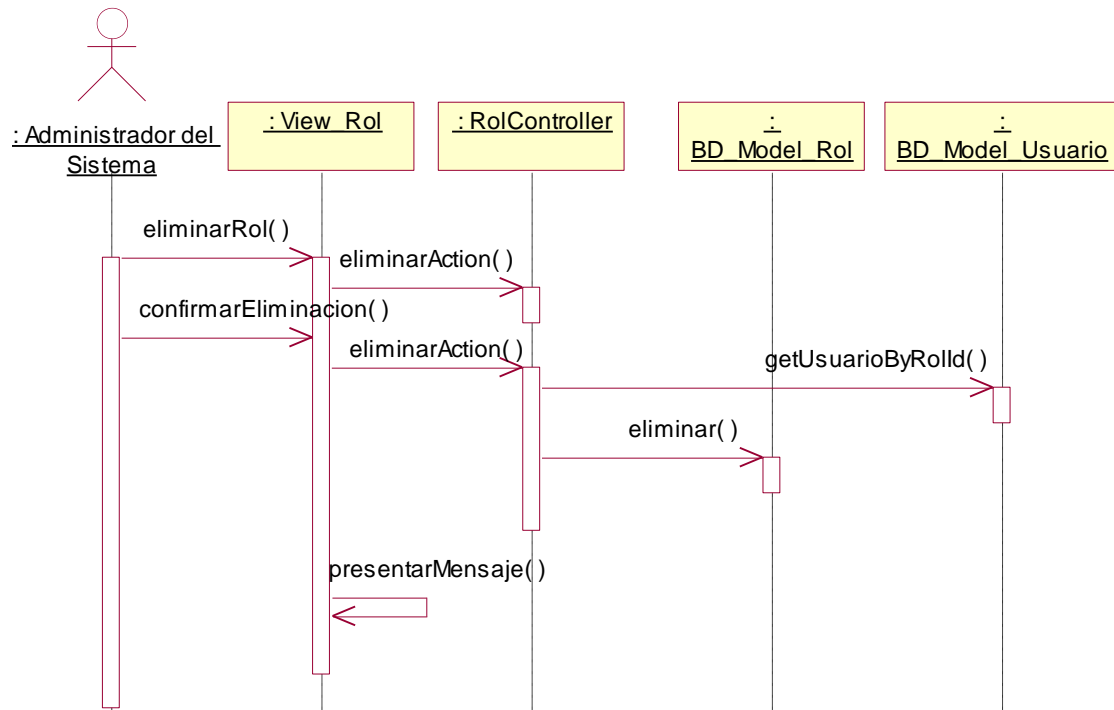


Figura 4.207 Diagrama de secuencia “Eliminar”

**Descripción del diagrama de secuencia – Eliminar:** El usuario en la lista de roles, hace clic en la opción “Eliminar”, este evento ejecuta la opción eliminarRol de la clase View\_Rol, la función eliminarRol hace una llamada a la función eliminarAction de la clase RolController, la función eliminarUsuario presenta la opción para confirmar la eliminación. El usuario confirma la eliminación haciendo clic en el botón eliminar, este evento ejecuta la función confirmarEliminación de la clase View\_Rol el cual invoca nuevamente a la función eliminarAction de la clase RolController la cual primero verifica que el rol no tenga usuarios asociados a través de la función getUsuarioByRolId de la clase BD\_Model\_Usuario, una vez que la función eliminarAction confirma que ese rol no tiene usuarios asociados, ejecuta la función eliminar de la clase BD\_Model\_Rol y luego la clase View\_Rol a través de la función presentarMensaje indica que la operación tuvo éxito.

## Diagrama de Secuencia – Asociar Recursos

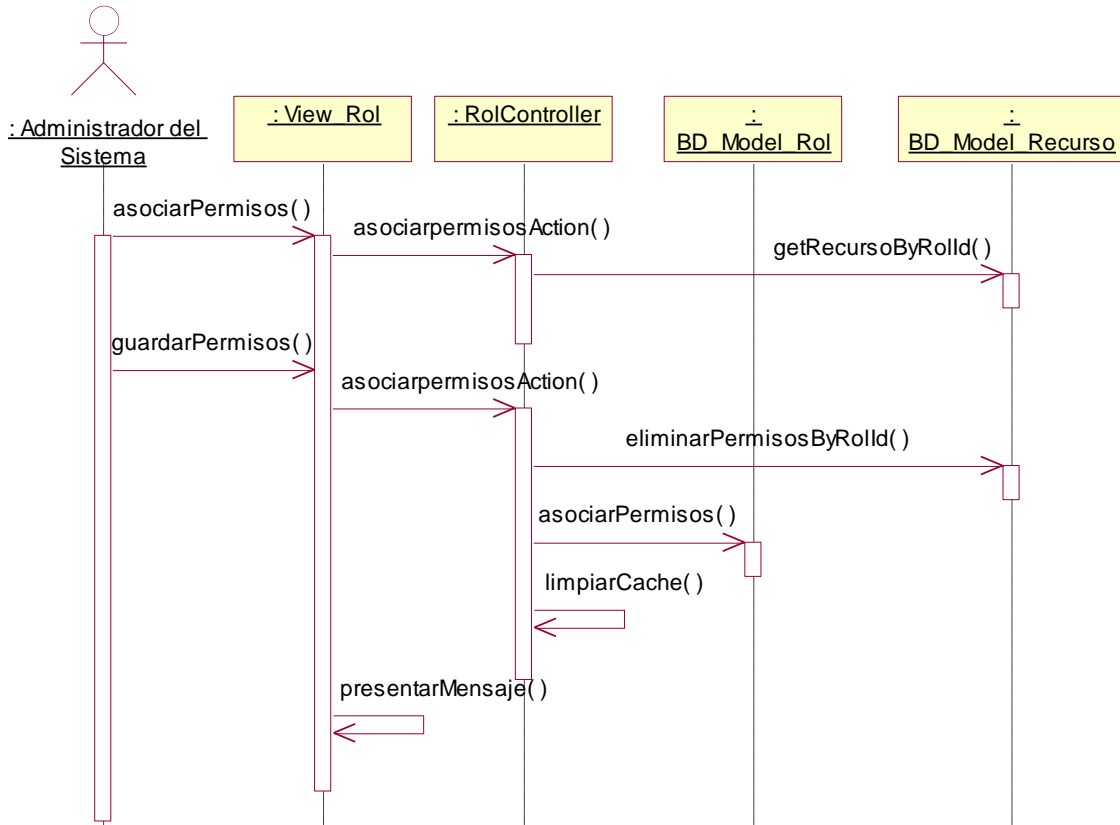


Figura 4.208 Diagrama de secuencia “Asociar Recursos”

**Descripción del diagrama de secuencia – Asociar Recursos:** El usuario en la lista de roles, hace clic en la opción “Permisos”, este evento ejecuta la opción asociarPermisos de la clase View\_Rol, la función asociarPermisos hace una llamada a la función asociarpermisosAction de la clase RolController el cual invoca a la función getRecursoByRolId de la cuenta BD\_Model\_Recurso a fin de obtener a los recursos que tiene asociado el rol, una vez que la función asociarpermisosAction tiene los datos, los pasa a la función asociarPermisos y estos son presentados al usuario. El usuario selecciona los permisos y sus actividades y confirma la asociación haciendo clic en el botón “Establecer Permisos”, este evento ejecuta la función guardarPermisos de la clase View\_Rol el cual invoca nuevamente a la función asociarpermisosAction de la clase RolController la cual primero elimina las asociaciones anteriores a través de la función eliminarPermisosByRolId de la clase BD\_Model\_Recurso, luego ejecuta la función asociarPermisos de la clase BD\_Model\_Rol a fin de guardar los nuevas permisos, limpia el directorio cache que contiene los permisos de los roles, y luego la clase View\_Rol a través de la función presentarMensaje indica que la operación tuvo éxito.

## **4.4 Modelo de Datos**

### **4.4.1 Diagrama de Clases Entidades de Diseño**

### **4.4.2 Diagrama del Diseño Físico de Datos**

### **4.4.3 Diccionario de Datos**

#### **4.4.2.1 Tablas**

Alerta: Almacena todas las alertas generadas por los equipos de localización

Card: Almacena todos los equipos de localización

CardData: Almacena toda la información transmitida por los equipos de localización

CardLastData: Almacena la última trama generada por cada equipo de localización.

Card\_Modelo: Almacena la información de los modelos de equipos de localización.

Card\_Modelo\_x\_Comando: Almacena la relación entre los modelos de equipos y los comandos del sistema.

Card\_Modelo\_x\_Sensor: Almacena la relación entre los modelos de equipos y los sensores del sistema:

Chofer : Almacena la información de los choferes del sistema

Ciudad: Almacena información de todas las ciudades del país.

Comando: Almacena la información de los comandos que el sistema puede enviar.

Comando\_Argumento: Almacena la información de los argumentos de los comandos.

Cuenta: Almacena la información de los clientes

Distrito: Almacena la información de todos los distritos del país.

Evento: Almacena la información de los eventos que el sistema puede detectar.

Geocerca: Almacena la información de las geocercas del sistema.

Geocerca\_Coordenadas: Almacena la información de las coordenadas de las geocercas en el eje x.y.

Geocerca\_Inout: Almacena la información de la unidades vehiculares que entran y salen de la geocerca.

Geocerca\_x\_grupo: Almacena la asociación de geocercas y flotas vehiculares.

Grupo: Almacena la información de las flotas vehiculares.

País: Almacena la información de los países.

Persona: Almacena la información de datos comunes de personas naturales y jurídicas.

Persona\_Juridica: Almacena la información de las personas jurídicas.

Persona\_Natural: Almacena la información de las personas naturales.

Poi: Almacena la información de los puntos de interés del sistema.

R\_Comando: Almacena la información de los comandos enviados.

Recordar: Almacena la información de los tokens de recuperación de contraseña.

Recurso: Almacena la información de funcionalidades y permisos del sistema.

Region: Almacena la información de las regiones del país.

Rol: Almacena la información de los roles del sistema.

Rol\_x\_Recurso: Almacena la asociación de roles, funcionalidades y recursos.

Sector\_Económico: Almacena la información de sectores económicos de los clientes.

Sensor: Almacena la información de sensores de los modelos de equipos de localización.

Tipo\_Carga: Almacena la información de los tipo de carga de las unidades vehiculares.

TipoDocumento: Almacena la información de los tipos de documentos de identificación de las personas.

TipoPoi: Almacena las categorías de los puntos de interés.

Unidad: Almacena la información de las unidades vehiculares.

Unidad\_Medida: Almacena la información de las unidades de medida de los sensores.

Usuario: Almacena la información de los usuarios del sistema.

Usuario\_x\_grupo: Almacena la asignación entre usuarios y flotas vehiculares para el monitoreo.

#### **4.4.2.2 Procedimientos Almacenados**

Fnc\_Analisis\_Trama: Este procedimiento almacenado analiza un lote de tramas generadas por los equipos de localización, ubicando las más recientes tramas en la tabla CardLastData y pasando las más antiguas a la tabla CardData. Así mismo detecta que unidades vehiculares están pasando por una geocerca y detecta si está generando una alerta del sistema.

Fnc\_Punto\_en\_Poligono: Este procedimiento almacenado detecta si un punto está atravesando un polígono, se utiliza a fin de determinar si un vehículo está pasando por una geocerca.

## 4.5 Modelo de Componentes

El modelo de componentes es la distribución física de los componentes del sistema, para lo cual se deberán detallar las principales características de cómo la aplicación será desplegada en el servidor de aplicaciones.

### 4.5.1 Web Server

El servidor apache 2.0 tiene una carpeta llamada htdocs, donde se encuentran por defecto las aplicaciones web, por lo general la ruta de este archivo es (en Centos) /var/www, por lo tanto la jerarquía de los directorios de la aplicación será la siguiente:

/var/application (Que contendrán todo el código fuente de la solución software)

/var/www/app/index.php (Archivo público de inicio de la aplicación)

/var/www/app/archivos (Directorio donde se almacenarán fotos, imágenes, y otros archivos multimedia.

La carpeta /var/application se encuentra fuera del alcance del directorio raíz de la aplicación a fin de proteger el código fuente de cualquier tipo de intromisión o ejecución de manera irregular. Así los permisos de las carpetas de este nivel son de nivel 755 y los archivos de nivel 775.

En la carpeta /var/www los permisos podrán ser todos de nivel 775

El archivo de configuración de la aplicación será el archivo config.ini que estará ubicado en la ruta /var/application/config.ini y contendrá los accesos a la base de datos y otros parámetros de configuración, como parámetros de conexión a otros servicios.

### 4.5.2 La aplicación

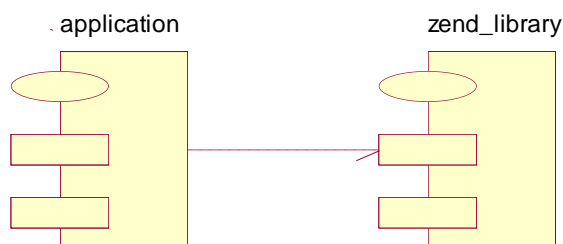


Figura 4.209 Diagrama de Componentes – Dependencia de Paquetes

Los siguientes paquetes se encontrarán en la ruta /var/application, y como podemos apreciar el paquete de “application” depende enteramente del paquete “zend\_library”, el paquete “application” contendrá enteramente las fuentes de la aplicación web y el paquete “zend\_library” contendrá la plataforma en las cuales nuestras clases y funciones tendrán soporte.

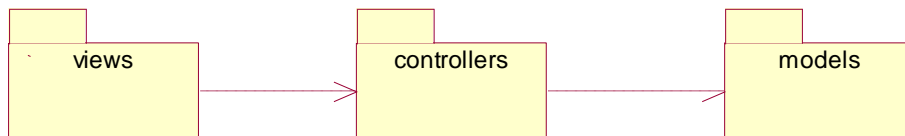


Figura 4.210 Vista interna del paquete “application”

El paquete “application” contiene las carpetas views, controller y models respectivamente, estas carpetas ilustran que la aplicación se basa en el patrón modelo-vista-controlador, y dentro de cada carpeta se ubicarán las clases respectivas.

#### 4.5.2.1 La aplicación – Carpeta “Models”

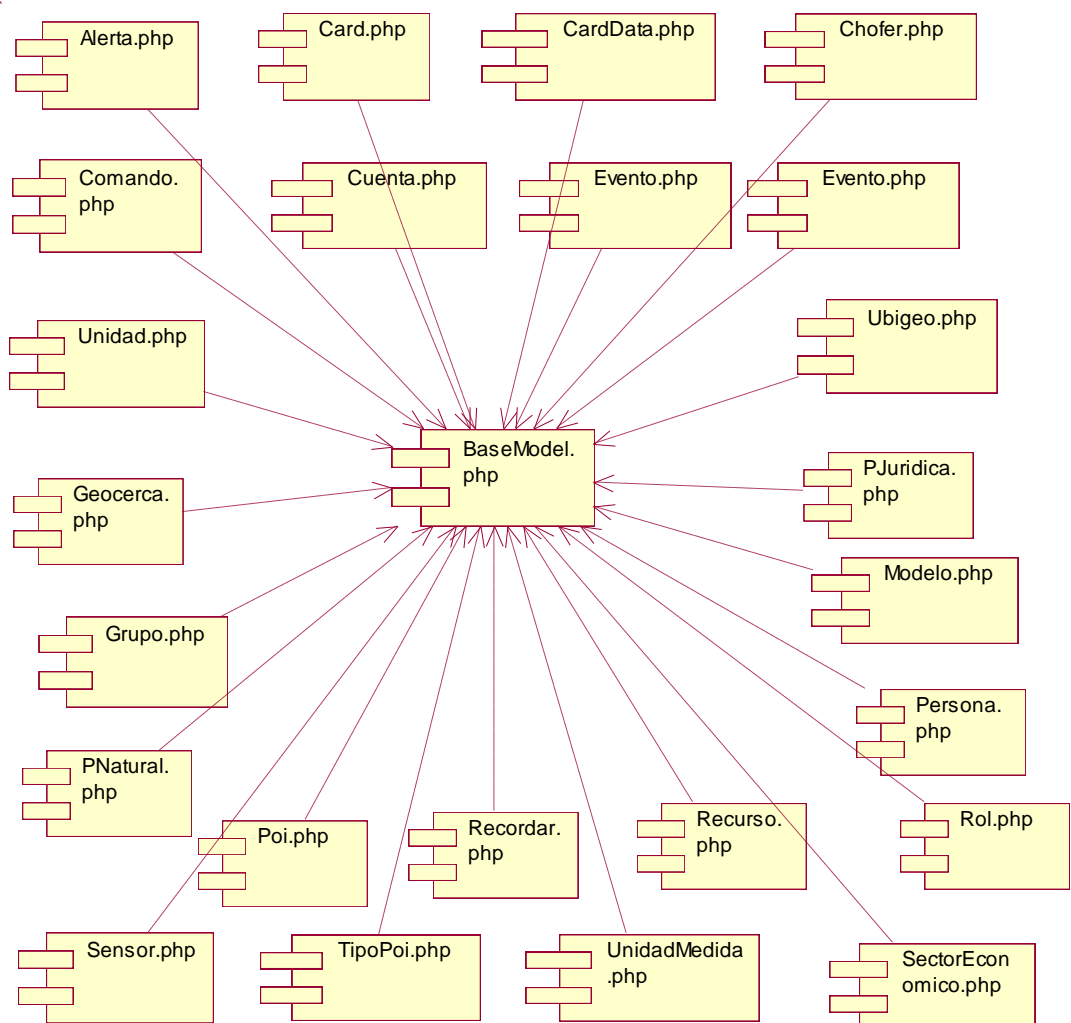


Figura 4.211 Archivos de la carpeta models

La carpeta models contiene archivos enteramente en PHP, los nombres de los archivos ilustran uno a uno el nombre de las clases de acceso a datos, es necesario resaltar que las funciones de conexión con la base de datos se encuentran en el archivo BaseModel.php, y todas los demás archivos de esta carpeta dependen de aquel archivo.

#### 4.5.2.2 La aplicación – Carpeta “Controllers”

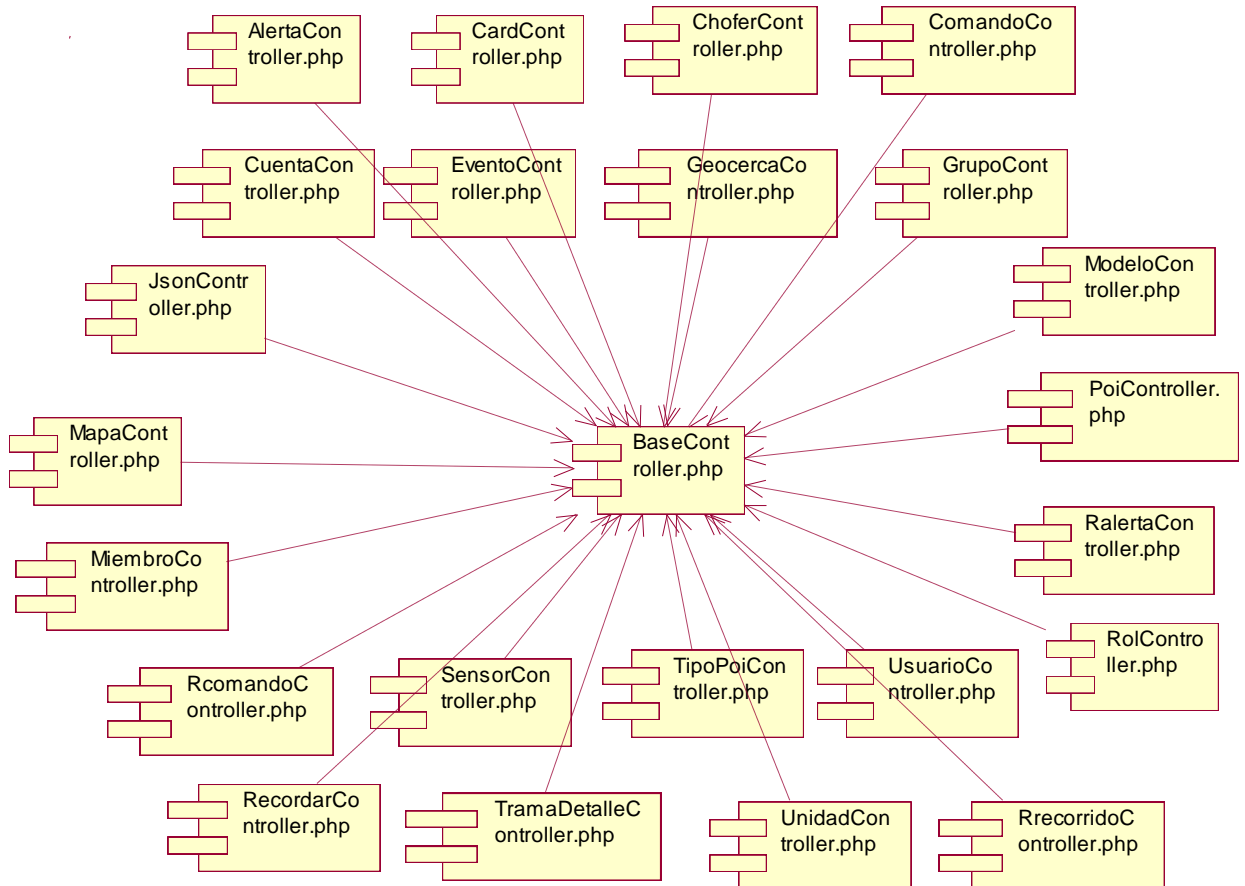


Figura 4.212 Archivos de la carpeta controllers

La carpeta controllers contiene archivos enteramente en PHP, los nombres de los archivos ilustran uno a uno el nombre de las clases de control, es necesario resaltar que las funciones que se reutilizan entre todas las clases de control se encuentran en el archivo BaseController.php, y todas los demás archivos de esta carpeta dependen de aquel archivo.



### 4.5.2.3 La aplicación – Carpeta “Views”



Figura 4.213 Carpetas de la carpeta views

La carpeta views está dividida en dos carpetas, la carpeta “js” y la carpeta “scripts”, en la carpeta “js” se almacenan los archivos de extensión javascript, los cuales en el diseño hacen llamadas tipo ajax, las clases están contenidas en 3 archivos principales: map.js , panel.js y markers.js.

El archivo maps.js contiene las clases que manejan el comportamiento del mapa, el archivo panel.js contiene las clases que manejan el comportamiento del panel de control y el archivo markers.js contiene las clases que manejan las geocercas, recorridos y agrupación de marcadores.



Figura 4.214 La carpeta “js”

La carpeta scripts contiene archivos que representan a las clases tipo VIEW de mantenimiento y descarga de archivos, todas las carpetas contenidas en la carpeta “scripts” contienen los archivos index.phtml, registrar.phtml, eliminar.phtml, que representaban en el diseño a las clases que daban inicio a las secuencias de listar, registrar y eliminar respectivamente. Además existen carpetas que contienen el archivo asociar.phtml que corresponden a la carpeta usuario, geocerca, grupo, modelo, y rol respectivamente.



**Figura 4.215** La carpeta “scripts”

## 4.6 Conclusiones

La arquitectura comprende diversas temáticas, desde la realización de casos de uso de análisis y diseño, hasta el diagrama de componentes, todos estos presentan un plano en el cual se deberá seguir en la programación del software. En el presente proyecto de investigación se ha descrito todas las funcionalidades del software junto con sus respectivas especificaciones, el objetivo de esto es tener un producto consecuente con la arquitectura planteada, de esta forma aseguramos, no solo la óptima calidad del software, además buscamos un mantenimiento adecuado en función al diseño inicial.

Existen diversas disciplinas en la ingeniería de software, no obstante, esta parte en especial es de suma importancia, por lo tanto se ha puesto especial interés en cada detalle a fin de garantizar resultados exitosos para el presente proyecto de investigación.

## CAPÍTULO V - DESARROLLO Y PRUEBAS

### 5.1 Introducción

En el presente capítulo presentaremos la plataforma tecnológica en la cual estará basada el software a desarrollar, esta plataforma tecnológica no solo permitirá la programación, sumando a esto exigirá el uso de mejores prácticas y el aseguramiento de la calidad del software, mediante el uso de herramientas de control y prueba. Así mismo describiremos las pruebas funcionales y de rendimiento a las que fue sometido el software desarrollado junto con sus resultados. Estas pruebas son constancia de que las funcionalidades inicialmente requeridas fueron satisfechas en su totalidad y cumplen todas las exigencias.

### 5.2 Desarrollo

#### 5.2.1 Plataforma tecnológica

##### Software de Desarrollo

Para el desarrollo de la solución software se utilizó la siguiente plataforma tecnológica: **Entorno de Desarrollo Integrado Netbeans 6:** IDE que posee un potente editor para PHP, que reconoce clases y funciones nativas de la versión de PHP seleccionada, así mismo se conecta automáticamente con un servicio de CVS para manejar versiones de las fuentes. Además posee un potente buscador de frases, y archivos que hacen la programación más ágil y sencilla.

**Firebug:** Debido a que se implementarán diversas funcionalidades en javascript, se utilizará una consola de control y seguimiento del comportamiento del navegador web, para esto firebug es una herramienta muy útil que permite identificar los errores, llamadas asíncronas, revisar analizar contenidos en formato JSON y analizar objetos HTML.

**BadBoy:** Es una poderosa aplicación diseñada para probar aplicaciones web con una plataforma cliente compleja, ya que graba el comportamiento del usuario utilizando la aplicación web como si fuera un simple navegador, y arroja una sencilla y definida secuencia de pruebas que luego puede ser usada por un ejecutor de pruebas como Jmeter.

##### Frameworks

La aplicación web se basará en el Framework de ZEND 1.10 que posee las librerías necesarias donde la aplicación web estará soportada en PHP 5.

Así mismo se utilizará el framework en javascript de Google Maps versión 3, que posee las funciones necesarias para presentar información geográfica en el mapa de Google.

### **Patrones**

Zend Framework promueve el uso del patrón Modelo Vista Controlador, por lo tanto para la programación utilizaremos el patrón MVC.

En el caso de los programas clientes en Javascript, estos serán tomados como programas de tipo presentación, dejando toda la lógica del negocio por lado del servidor.

## **5.3 Pruebas**

### **5.3.1 Plan de Pruebas del Proyecto**

#### **Propósito**

El propósito de este plan de pruebas para la aplicación web es tratar de cumplir los siguientes objetivos:

1. Identificar las funcionalidades más importantes del proyecto que deben de ser probados.
2. Listar los principales requisitos a probar.
3. Identificar las estrategias de prueba a utilizar o los tipos de prueba.
4. Listar los artefactos a entregar después de haber ejecutado las pruebas.

#### **Entorno**

El proyecto donde se define este plan de pruebas corresponde a la aplicación de monitoreo vehicular a desarrollarse, la arquitectura de la aplicación, corresponde a un aplicación web donde la mayoría de sus funcionalidades principales están basadas en Ajax.

#### **Alcance**

Se realizarán las siguientes pruebas:

Pruebas de caja negra: Su Objetivo es asegurar la funcionalidad requerida, incluyendo la navegación, entrada de datos, su procesamiento y recuperación.

Pruebas de rendimiento: Estudio del rendimiento de la herramienta con poca cantidad de procesamiento, con cantidad media, y con alta cantidad de procesamiento.

#### **Requerimientos para las Pruebas**

Se someterán a prueba los siguientes casos de uso de prueba con su respectiva correspondencia con los casos de uso del sistema:

##### **Paquete Consulta**

1. Ver Mapa
2. Ver Unidades
3. Ver Recorrido
4. Ver Alertas
5. Ver Comandos
6. Ver Detalle de la Posición
7. Atender Alerta
8. Enviar Comando

9. Buscar Unidad
10. Buscar Punto de Interés
11. Ver Punto de Interés
12. Seleccionar cliente en el Mapa

### **Paquete Configuración**

13. Mantenimiento de Equipos de Localización
14. Asociar Comando a Modelo de Equipo
15. Mantenimiento de Comandos
16. Mantenimiento de Eventos
17. Mantenimiento de Tipos de Puntos de Interés

### **Paquete Administración**

18. Mantenimiento de Clientes
19. Mantenimiento de Equipos de Localización
20. Mantenimiento de Unidades
21. Mantenimiento de Conductores
22. Mantenimiento de Geocercas
23. Mantenimiento de Puntos de Interés

### **Paquete Reportes**

24. Generación de Reporte de Recorrido
25. Generación de Reporte de Alertas
26. Generación de Reporte de Comandos

### **Paquete Seguridad**

27. Ingresar al Sistema
28. Salir del Sistema
29. Actualizar Cuenta
30. Mantenimiento de Usuarios
31. Mantenimiento de Roles

### **Sistema**

IP del Servidor: 10.0.37.149

Sistema Operativo: Ubuntu 11

Nombre de la base de datos: gis2

Configuración: Intel Corei3 3.3GHZ Memoria 4GB DD3 Disco 500GB

### 5.3.2 Ejecución de Casos de Uso de Pruebas

#### Paquete Consulta

CUP_001	Ver Mapa		
Desc.	Carga del Mapa de Monitoreo Vehicular		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	Se selecciona el menú "Ver Mapa"	La interfaz de usuario es cargada, donde se visualiza la barra de búsquedas, el panel de control y el mapa web.	La interfaz de usuario cargó, mostrando el panel de control, la barra de búsqueda y el mapa en google maps
2	Se calcula el tiempo de rendimiento	Deberá tomar menos de 6 segundos	El mapa cargó en 4 segundos, aunque varía en función a la velocidad de conexión a internet, puede llegar a 8 segundos

CUP_002	Ver Unidades		
Desc.	Ver la lista de unidades en el panel de control		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	Se selecciona el menú "Ver Mapa"	Repite el paso 1 de CUP_Ver Mapa y en el panel de control en la pestaña Mapa, lista las unidades asignadas al usuario.	Las unidades fueron listadas en la pestaña mapa del panel de control, la carga fue inmediata.
2	Visualizar el mapa web	Deberá mostrar las unidades como marcadores y las que están cercanas deberán estar agrupadas.	El mapa web muestra las unidades como marcadores y las unidades que transmiten el rumbo las muestra con un único romboide, las unidades próximas por nivel de zoom son agrupadas en íconos redondos.
3	Hacer clic en un marcador	Deberá mostrar una ventana de información con los datos básicos de la transmisión	Al hacer clic muestra los datos básicos de la transmisión en una ventana de información del tipo infoWindow.

CUP_003	Ver Recorrido		
Desc.	Visualizar el recorrido en el mapa web		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la ventana de información de una unidad, hacer clic en la opción "Ruta"	Las últimas 30 posiciones de la unidad son listadas en la pestaña Ruta del panel de control y el recorrido es visualizado como una línea en el mapa web y cada transmisión es un marcador,	La pestaña ruta lista las 30 últimas posiciones de la unidad seleccionada, esta lista es paginada de 30 en 30. Así mismo el recorrido de la unidad es visto como una línea, donde cada transmisión es un marcador.
2	Seleccionar otra unidad y hacer clic en la opción "Ruta"	La pestaña Ruta del panel de control da una opción para seleccionar entre las 2 unidades a fin de listar su recorrido, el recorrido de esta también es visualizado como una línea en el mapa web.	La pestaña ruta muestra una selección donde se puede escoger que unidad se desea listar su recorrido. El mapa web muestra el recorrido de ambas unidades.
3	Seleccionar la opción refrescar en el pestaña Mapa.	Los recorridos son borrados del mapa web.	Los recorridos son borrados del mapa web.

CUP_004	Ver Alertas		
Desc.	Listar las alertas en la pestaña de alertas		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	Hacer clic en la pestaña de alertas del panel de control.	Se listan las alertas de las unidades asignadas al usuario.	Las alertas de las unidades no atendidas son listadas en el panel de alertas.
2	Hacer clic en una alerta listada.	Se visualiza en una ventana la información básica de una alerta.	La ventana de información es abierta mostrando la información básica de la alerta.
3	Calcular el tiempo de carga.	La carga no deberá exceder los 4 segundos.	La carga es inmediata.

CUP_005	Ver Comandos		
Desc.	Listar los comandos enviados en la pestaña de comandos.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	Hacer clic en la pestaña de comandos del panel de control.	Se listan los comandos enviados y sus estados de las unidades.	Los comandos enviados de las unidades asignadas al usuario son listadas en el panel de comandos.
2	Hacer clic en un comando enviado.	Se visualiza en una ventana la información básica de un comando enviado.	La ventana de información es abierta mostrando la información básica de la alerta.
3	Calcular el tiempo de carga	La carga no deberá exceder los 4 segundos.	La carga es inmediata.

CUP_006	Ver Detalle de la Posición		
Desc.	Mostrar en una ventana emergente la información de la trama enviada.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En una ventana de información básica de un marcador, hacer clic en la opción "Detalle".	Una ventana emergente es abierta, la posición de la unidad en el mapa, el detalle de la trama, información del equipo, información de contacto e información de la alerta en caso de haberla.	La ventana emergente es abierta mostrando la unidad en el mapa, la información del equipo, la información de la trama, la foto de la unidad (en caso de haberla), y un formulario de atención de la alerta.
2	Hacer clic en una región fuera de la ventana.	La ventana emergente es cerrada.	La ventana emergente se cierra y se visualiza nuevamente la interfaz del mapa web.



CUP_007	Atender Alerta		
Desc.	Atender una alerta desde el panel de alertas.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el panel de alertas, hacer clic en el ID de una alerta y hacer clic en la opción "Detalle"	La ventana de información de la trama es abierta, y en la sección alerta un formulario requiriendo ingresar la atención de la alerta.	La ventana de información de la trama se abre, se visualiza el formulario de atención de la alerta mostrando una caja de texto donde ingresar la atención.
2	En el formulario de atención hacer clic en la opción "Atender Alerta"	La ventana de información se cierra y la lista de alertas pendientes de atención se actualiza y ya no muestra la alerta recientemente atendida.	La ventana de información se cierra y actualiza la lista de alertas del panel de alertas.

CUP_008	Enviar Comando		
Desc.	Enviar un comando a un equipo de localización.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el mapa web, hacer clic en una unidad, en su ventana de información hacer clic en la opción "Comando".	Una ventana emergente se abre donde se muestra un mensaje de advertencia.	Una ventana emergente se abre y se muestra un mensaje indicando que un comando enviado puede cambiar el comportamiento del equipo.
2	Hacer clic en el botón aceptar.	Una lista de comandos asociados al equipo es presentado.	La lista de comandos del equipo es presentada.
3	Seleccionar un comando y hacer clic en el botón siguiente.	Se presenta el detalle del comando a enviar junto con su lista de argumentos.	El detalle del comando a enviar es presentado y su lista de argumentos.
4	Llenar lista de argumentos y hacer clic en el botón "Enviar Comando Ahora"	El comando es enviado, y la ventana emergente se cierra, se actualiza la lista de comandos enviados.	La ventana emergente se cierra y en la lista de comandos enviados aparece el comando recientemente enviado junto con su estado.

CUP_009	Buscar Unidad		
Desc.	Buscar Unidad en la barra de búsqueda del mapa web.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la caja de texto de unidad, escribir la mitad del número de placa de una unidad vehicular.	La caja de texto muestra una lista de placas que coinciden con el texto ingresado.	Se muestra una lista de placas cuyos textos coinciden con el texto ingresado.
2	Seleccionar un ítem de la lista.	El mapa web se centra en función a la posición en latitud y longitud de la unidad seleccionada.	Se centra el mapa indicando la posición exacta de la unidad seleccionada.

CUP_010	Buscar Punto de Interés		
Desc.	Buscar un Punto de Interés en el mapa web.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la caja de texto de Lugar, ingresar la mitad del nombre de un punto de interés.	La caja de texto muestra una lista de puntos de interés cuyos nombres coinciden con el texto ingresado.	Se muestra una lista de puntos de interés cuyos nombres coinciden con el texto ingresado.
2	Seleccionar un ítem de la lista.	El mapa web se centra en función a la latitud y longitud del punto de interés ingresado.	Se centra el mapa en función al punto de interés seleccionado.

CUP_011	Ver Puntos de Interés		
Desc.	Ver los puntos de interés en el mapa.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el mapa web, hacer clic en el botón "Pto. Interés".	Una ventana emergente es mostrada, listando los tipos de puntos de interés junto con el botón "Ver en Mapa".	Una ventana muestra los tipos de puntos de interés.
2	Seleccionar los tipos de puntos de interés que deseamos visualizar y hacer clic en el botón "Ver en Mapa".	Los puntos de interés relacionados con los tipos de puntos de interés seleccionados son mostrados en el mapa web.	Se cargan los puntos de interés en el mapa web.
3	Calcular el tiempo de carga.	La carga de los puntos de interés no deberá exceder los 4 segundos.	Se toma alrededor de 3 segundos en mostrar los puntos de interés.

CUP_012	Seleccionar cliente en el Mapa		
Desc.	Ver los puntos de interés en el mapa.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la caja de texto de cliente, ingresar la mitad de un nombre de una cuenta cliente.	Se presenta una lista con nombre de cuentas clientes que coinciden con el texto ingresado.	Se listan los clientes cuyos nombres coinciden con el texto ingresado.
2	Seleccionar un ítem de la lista.	Las unidades, alertas, comandos y marcadores son actualizados en función al nuevo cliente seleccionado.	Toda la información del mapa web se carga con la nueva cuenta cliente.
3	Calcular el tiempo de carga.	La carga no deberá exceder los 4 segundos.	Se toma alrededor de 3 segundos en mostrar la nueva información.

### Paquete Configuración

CUP_001	Mantenimiento de Modelos de Equipos de Localización		
Desc.	Listar y registrar un modelo de equipo.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Modelo".	Se listan los modelos de equipos registrados en el sistema de forma paginada.	Se listan las modelos de los equipos.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de modelo de equipo.	El formulario es presentado.
3	Ingresar el nombre, marca, modelo, servicio, tecnología, seleccionar los sensores en cada puerto de entrada del modelo del equipo y hacer clic en la opción "Registrar".	Se registra el modelo y se presenta un mensaje de éxito.	Se presenta un mensaje indicando que el modelo fue registrado y mostrando el detalle del modelo.

CUP_002	Asociar Comando a Modelo de Equipo		
Desc.	Asociar Comandos a un Modelo.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la lista de modelos de equipos, hacer clic en el enlace "Comandos".	La lista de comandos es mostrada junto con un ícono indicando si el comando fue asociado con el modelo del equipo.	Se listan los comandos del sistema indicando cuales estaan asociados al modelo del equipo.
2	Hacer clic en la opción "Asignar Comandos".	Se presenta la misma lista de comandos pero con un caja de selección en la parte izquierda.	Se listan los comandos con la opción de escogerlos.
3	Seleccionar los comandos que irán asociados al modelo y hacer clic en la botón "Guardar".	Un mensaje de éxito es mostrado y listando los comandos seleccionados.	Se listan los mensajes seleccionados con un mensaje indicando que la operación fue exitosa.

CUP_003	Mantenimiento de Comandos.		
Desc.	Listar y Registrar un Comando.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Comandos".	Se listan los comandos registrados en el sistema de forma paginada.	Los comandos del sistema son listados.
2	Seleccionar la opción "Registrar".	Se presenta el formulario de registro de comando.	Se presenta el formulario de registro de comando.
3	Ingresar el código, el nombre, la descripción. Ingresar 3 argumentos y hacer clic en el botón "Guardar".	Se presenta el detalle del comando guardado junto con un mensaje de éxito.	Un mensaje indica que el comando fue registrado y se visualiza el detalle del comando ingresado.

CUP_004	Mantenimiento de Eventos		
Desc.	Listar y Registrar un Evento		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menu seleccionar la opción "Eventos".	Se listan los eventos registrados en el sistema de forma paginada.	Los eventos del sistema son listados.
2	Seleccionar la opción "Registrar".	Se presenta el formulario de registro de evento.	Se presenta un formulario solicitando los campos del evento.
3	Ingresar el ID, el nombre, el Cliente, la descripción, el tipo, si es alarma o no, la alerta, y la prioridad, hacer clic en la opción "Registrar".	Se presenta un mensaje de éxito junto con el detalle del evento registrado.	Un mensaje de éxito indica que el evento fue registrado satisfactoriamente, junto con el detalle del evento.

CUP_005	Mantenimiento de Tipos de Puntos de Interés		
Desc.	Listar y Registrar un Tipo de Punto de interés.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Tipos POI".	Se listan los tipos de puntos de interés registrados en el sistema de forma paginada.	Los tipos de puntos de interés del sistema son listados.
2	Seleccionar la opción "Registrar".	Se presenta el formulario de registro de tipo de punto de interés.	Un formulario solicita el nombre del tipo de punto de interés a registrar.
3	Ingresar el nombre del punto de interés y hacer clic en el botón "Registrar".	Se presenta un mensaje de éxito indicando que la operación fue exitosa, el nombre del tipo de punto de interés y la opción "Seleccionar Ícono".	Un mensaje de éxito es mostrado presentando el nombre del tipo de punto de interés.
4	Hacer clic en el enlace "Selecciona Ícono", seleccionar una imagen y aceptar la carga.	El ícono del tipo de punto de interés se actualiza.	Se refresca la página y el ícono cambia por la imagen seleccionada.

### **Paquete Administración**

CUP_001	Mantenimiento de Clientes		
Desc.	Listar y registrar un cliente.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Clientes".	Se listan los clientes registrados en el sistema de forma paginada.	Los clientes son listados, tanto personas naturales como jurídicas.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de clientes, indicando los campos obligatorios a ingresar.	El formulario de registro de personas jurídicas se presenta.

3	Ingresar el ruc, la razón social, el nombre corto, la fecha de expiración, el sector económico, el representante legal, los datos de la dirección fiscal y hacer clic en el botón "Registrar".	Se presenta un mensaje indicando que el cliente fue registrado satisfactoriamente y se presenta el formulario de registro de usuario cliente.	Se presenta un mensaje y un formulario requiriendo los datos del usuario administrador por parte del cliente.
4	Ingrese el correo electrónico, la contraseña, los nombres, los apellidos, el tipo de documento, el número de documento y haga clic en el botón "Registrar".	Se presenta un mensaje indicando que la operación tuvo éxito junto con los datos del usuario cliente registrado.	Un mensaje indica que el usuario fue registrado y se ve el detalle de sus datos.

CUP_003	Mantenimiento de Equipos de Localización		
Desc.	Listar y registrar equipos de localización en el sistema.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Cards".	Se listan los equipos de localización registrados en el sistema de forma paginada.	Se listan los equipos de localización que estan registrados en el sistema.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de Card.	Se presenta el formulario de registro y solicitando datos obligatorios.
3	Ingresar el Id del Card, en número de serie, el código de autenticación, el modelo del equipo, la fecha de expiración y su estado y hacer clic en el botón "Registrar".	Se presenta un mensaje de éxito indicando que el Card fue registrado satisfactoriamente, junto con el detalle del Card.	Un mensaje confirma el registro del card y muestra los datos registrados.

CUP_004	Mantenimiento de Unidades		
Desc.	Listar y registrar unidades vehiculares en el sistema.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Unidades".	Se listan las unidades asociadas a la cuenta cliente.	Las unidades de la cuenta cliente se listan.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de unidad vehicular.	El formulario de registro de unidad se presenta, exigiendo datos obligatorios.
3	Ingresar la flota, el número de serie, la clave de autenticación, el número de placa, el teléfono de contacto, la marca y hacer clic en el botón "Registrar".	Se presenta un mensaje de éxito indicando que la unidad fue registrada satisfactoriamente junto con los datos de la unidad recién ingresada.	Un mensaje de éxito confirma la operación y muestra los datos recién ingresados.

CUP_005	Mantenimiento de Conductores		
Desc.	Listar y registrar conductores.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Conductores".	Se listan los conductores asociados a la cuenta cliente de forma paginada.	Se listan los conductores del cliente.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de conductor.	Se presenta el formulario de registro de conductor solicitando datos obligatorios.
3	Ingresar el correo electrónico, nombres, apellidos, tipo de documento, número de documento, el estado, y hacer clic en la opción "Registrar".	Se presenta un mensaje de éxito indicando que el conductor fue registrado satisfactoriamente junto con los datos del conductor.	Un mensaje de éxito indica que la operación fue correcta y muestra los datos del conductor registrado.



CUP_006	Mantenimiento de Geocercas		
DESC.	Listar y registrar geocercas		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "Geocercas".	Se listan las geocercas de la cuenta cliente de forma paginada.	Se listan las geocercas de la cuenta cliente.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de geocercas.	El formulario se presenta indicando datos obligatorios.
3	Ingresar el nombre de la geocerca, el color de su borde, el color de su sombra y hacer clic en el botón "Registrar".	Se presenta un mensaje de éxito indicando que la geocerca fue registrada, y se presenta un mapa solicitando el ingreso de las coordenadas.	Un mensaje confirma la operación y redirige al mapa de ingreso de coordenadas.
4	Dibujar un cuadrado haciendo clic cuatro veces en lados opuestos del mapa y hacer clic en la opción "Guardar Coordenadas".	Se presenta un mensaje indicando que las coordenadas fueron guardadas y se muestra en un mapa el polígono con las coordenadas registradas.	Un mensaje indica que las coordenadas fueron ingresadas y se muestra la geocerca en un mapa.
5	Volver a la lista de geocercas y hacer clic en el enlace "Flotas"	La lista de flotas de la cuenta cliente se presenta indicando que flotas estan asociadas a la geocerca.	Se listan las flotas de la cuenta cliente y cada flota indica si está asociada con la geocerca.
6	Hacer clic en el enlace "Actualizar Flotas"	Se presenta un formulario indicando por cada flota si la geocerca será exclusiva, prohibida, de control y un rango horario.	Se presenta la lista de flotas con la opción de escogerlas, seleccionar el tipo de geocerca, el rango de horas.
7	Hacer clic en el botón "Guardar"	Un mensaje se presenta indicando que las flotas fueron correctamente asociadas, mostrando la pantalla anterior pero con los datos actualizados.	Se presenta el mensaje de confirmación de la asociación y se listan las flotas asociadas.

CUP_007	Mantenimiento de Puntos de Interés		
DESC.	Listar y registrar Puntos de Interés		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar la opción "POI".	Se listan los puntos de interés asociados a la cuenta cliente de forma paginada.	Se listan los puntos de interés del cliente.
2	Hacer clic en la opción "Registrar".	Se presenta el formulario de registro de punto de interés.	El formulario requiere campos obligatorios.
3	Ingresar el nombre y el tipo de punto de interés y hacer clic en la opción "Registrar".	Se presenta un mensaje indicando que el punto de interés fue registrado y se muestra un mapa donde con un marcador se puede indicar la posición del punto de interés.	El mensaje de éxito de la operación se muestra y se muestra un mapa donde se puede indicar la posición del punto de interés con un marcador.
4	Mover el marcador hasta la posición deseada, y hacer clic en el botón "Actualizar".	Un mensaje indica que la posición fue actualizada y muestra en un mapa al punto de interés en la posición indicada.	El mensaje de éxito se muestra y un mapa indica la posición del punto de interés.

### **Paquete Reportes**

CUP_001	Generacion_Reporte_Recorrido		
DESC.	Generar Reporte de Recorrido.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar "Reporte de Recorrido"	Se presenta un formulario donde se indica los parámetros del reporte.	El formulario se presenta solicitando la flota y la placa.
2	Seleccionar la flota y la placa, hacer clic en la opción "Siguiente".	Se presenta un reporte en formato HTML paginado, con las opciones de exportar a CSV, KML, PDF, XLS.	Se listan los registros de las tramas de forma paginada, junto con las opciones de exportación.

3	Hacer clic en la opción KML	Se descarga el archivo KML y se abre el Google Earth donde se visualiza el recorrido de la unidad	El navegador pregunta si se desea abrir o guardar el archivo, al abrir se abre el Google Earth y muestra el recorrido de las tramas.
---	-----------------------------	---	--

CUP_002	Generacion_Reporte_Alertas		
DESC.	Generar Reporte de Alertas		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar "Reporte de Alertas"	Se presenta un formulario donde se indica los parámetros del reporte.	Se presenta el formulario solicitando la flota.
2	Seleccionar la flota, la placa, seleccionar como alerta el botón de pánico y hacer clic en "Siguiete".	Se presenta un reporte en formato HTML paginado con la opciones de exportación en XML, CSV, XLS Y PDF.	Se listan las alertas de forma paginada con opciones de exportación.
3	Hacer clic en la opción PDF, y en la ventana de información hacer clic en abrir.	Se abre un archivo PDF listando las alertas de las unidades seleccionadas.	Se puede visualizar el reporte de alertas en formato PDF.

CUP_003	Generacion_Reporte_Comandos		
DESC.	Generar Reporte de Comandos		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú seleccionar "Reporte de Comandos"	Se presenta el formulario de reporte de Comandos.	Se presenta el formulario solicitando la flota.
2	Seleccionar la flota, la placa, y el tipo de comando y hacer clic en el botón "Siguiete".	Se presenta un reporte en formato HTML paginado con las opciones de exportación en XML, CSV, XLS y PDF.	Se listan los comandos junto con su estado y las opciones de exportación.
3	Hacer clic en la opción PDF, y en la ventana de información hacer clic en abri.	Se abre un archivo PDF listando los comandos y su estado.	Se puede visualizar el reporte de comandos en formato PDF.

### Paquete Seguridad

CUP_001	Ingresar al Sistema		
DESC.	Iniciar Sesión en el Sistema		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la pantalla de inicio hacer clic en el enlace "Acceder".	Se presenta el formulario de acceso al sistema, solicitando el correo electrónico y la contraseña.	El formulario de acceso se presenta solicitando el correo y la contraseña.
2	Ingresar un correo electrónico válido con una contraseña inválida y hacer clic en el botón "Acceder".	Se presenta un mensaje indicando que el correo electrónico y/o la contraseña son incorrectos.	Un mensaje de error indica que los datos ingresados no coinciden.
3	Ingresar un correo válido con una contraseña válida y hacer clic en el botón "Acceder".	Se redirige a la pantalla de inicio al sistema con las opciones de menú cargadas en función al rol del usuario.	Se dirige a la pantalla de bienvenida del sistema donde se muestra el menú de opciones.

CUP_002	Salir del Sistema		
DESC.	Cerrar la Sesión del Sistema		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En la pantalla de inicio hacer clic en el enlace "Salir"	Se redirige nuevamente a la pantalla de inicio requiriendo el acceso al sistema nuevamente.	Se destruye la sesión y se presenta la ventana de acceso al sistema.

CUP_003	Actualizar Cuenta		
DESC.	Actualizar los datos personales y de ingreso al Sistema		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	Hacer clic en el enlace "Cuenta" del menú de opciones.	Se presentan los datos del usuario, sus datos de acceso, y su configuración en el mapa.	La pantalla detalla los datos de usuario, tanto personales, como acceso y consulta en el mapa.
2	Hacer clic en la opción "Actualizar datos de ingreso"	Se presentan 2 formularios donde el usuario puede cambiar su contraseña y su correo electrónico.	Dos formulario requieren cambiar de contraseña y actualizar correo electrónico.

3	Introducir una nueva contraseña, confirmarla y hacer clic en el botón "Actualizar".	Se presenta la pantalla inicial de datos del usuario junto con un mensaje indicando el éxito de la operación.	Un mensaje de éxito se presenta con la pantalla inicial.
4	Hacer clic en la opción "Actualizar datos personales"	Se presenta un formulario donde el usuario puede actualizar sus datos de persona natural.	Un formulario requiere los datos personales del usuario.
5	No realizar ningún cambio y hacer clic en la opción "Actualizar"	Se presenta la pantalla inicial de datos del usuario junto con un mensaje indicando el éxito de la operación.	Un mensaje de éxito se presenta con la pantalla inicial.
6	Hacer clic en la opción actualizar configuración del mapa.	Se presenta un formulario preguntando si desea activar el sonido de las alertas, e indicar la posición inicial de consulta en el mapa.	Un formulario permite seleccionar si se desea activar la alarma sonora y un mapa web deja indicar con un marcador la posición inicial de consulta en el mapa.
7	Hacer clic en el botón "Actualizar".	Se presenta la pantalla inicial de datos del usuario junto con un mensaje indicando el éxito de la operación.	Un mensaje de éxito se presenta con la pantalla inicial.

CUP_004	Mantenimiento de Usuarios		
DESC.	Listar, Registrar usuarios del sistema.		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú de opciones hacer clic en "Usuario".	Se deberán listar los usuarios del sistema de forma paginada.	Se listan los usuarios del sistema.
2	Hacer clic en la opción "Registrar".	Se deberá presentar el formulario de registro de usuarios	Se presentó el formulario de registro indicando los campos obligatorios.
3	Ingresar los datos obligatorios y hacer clic en el botón registrar.	Se deberá presentar un mensaje confirmando el éxito de la operación.	Se presenta un mensaje de éxito indicando que el registro fue correcto.

CUP_004	Mantenimiento de Roles		
DESC.	Listar, Registrar roles del sistema		
#	Pasos	Resultados Esperados	Resultados Obtenidos
1	En el menú de opciones hacer clic en "Rol".	Se deberán listar los roles del sistema de forma paginada.	Se listan los roles del sistema.
2	Hacer clic en la opción "Registrar".	Se deberá presentar el formulario de registro de roles	Se presentó el formulario de registro indicando los campos obligatorios.
3	Ingresar los datos obligatorios y hacer clic en el botón registrar.	Se deberá presentar un mensaje confirmando el éxito de la operación.	Se presenta un mensaje de éxito indicando que el registro fue correcto.

### 5.3.3 Pruebas de Rendimiento

A fin de ejecutar las pruebas de rendimiento se simularon las siguientes condiciones:

- 1 Tamaño en GigaBytes de la Base de datos en ambiente normal: 10 GB
- 2 Tamaña en GigaBytes de la Base de datos para las pruebas de carga: 30 GB
- 3 Número de consultas promedio al sistema que realiza un cliente: 7
- 4 Número de consultas al sistema para las pruebas de carga: 18 hilos
- 5 Número de equipos transmitiendo en ambiente normal: 1000 equipos
- 6 Número de equipos transmitiendo en ambiente de carga:3000 equipos

Se capturará la siguiente información:

- 1 Uso de Memoria
- 2 Uso de CPU (4 CPUs)
- 3 Media de Uso de CPU
- 4 Número de clientes simultáneos
- 5 Número de Muestras
- 6 Media
- 7 Desviación

Debido a que los usuarios permanecerán la mayor parte del tiempo utilizando el CUS Ver Mapa. La funcionalidad de esta caso de uso está basado en AJAX, y de manera asíncrona ejecuta llamadas al servidor a través de diferentes hilos, el rendimiento de la aplicación se definirá en función a como el sistema soporte todas esas llamadas recurrentes a fin de presentar la información solicitada.

Direcciones web a consultar de manera simultánea:

<a href="http://10.0.37.149/trunk/www/pe/mapa3">http://10.0.37.149/trunk/www/pe/mapa3</a>
<a href="http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/0/grupos//ixp/30/page/1/estado/0/nocache/8288">http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/0/grupos//ixp/30/page/1/estado/0/nocache/8288</a>
<a href="http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/0/grupos//ixp/30/page/1/estado/0/nocache/66874">http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/0/grupos//ixp/30/page/1/estado/0/nocache/66874</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidades/cuenta/0">http://10.0.37.149/trunk/www/pe/json/unidades/cuenta/0</a>
<a href="http://10.0.37.149/trunk/www/pe/json/cuenta/">http://10.0.37.149/trunk/www/pe/json/cuenta/</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidadgrupo/cuenta/90/grupos//nocache/98004">http://10.0.37.149/trunk/www/pe/json/unidadgrupo/cuenta/90/grupos//nocache/98004</a>
<a href="http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/2801">http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/2801</a>
<a href="http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/28771">http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/28771</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidades/cuenta/90">http://10.0.37.149/trunk/www/pe/json/unidades/cuenta/90</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/30/page/1/nocache/18751">http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/30/page/1/nocache/18751</a>
<a href="http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/96333">http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/96333</a>
<a href="http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/21487">http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/21487</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/999/page/2/nocache/47324">http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/999/page/2/nocache/47324</a>
<a href="http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/999/page/3/nocache/56263">http://10.0.37.149/trunk/www/pe/json/unidadrecorrido/card_id/605/ixp/999/page/3/nocache/56263</a>
<a href="http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/38913">http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/38913</a>
<a href="http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/61922">http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/61922</a>
<a href="http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/48452">http://10.0.37.149/trunk/www/pe/json/alerta/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/48452</a>
<a href="http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/23764">http://10.0.37.149/trunk/www/pe/json/comandosmapa/cuenta/90/grupos//ixp/30/page/1/estado/0/nocache/23764</a>

#### Resultados de la PRUEBA 01

Clientes simultáneos	1 cliente activo
Número Muestras	48 consultas
Media	1967 milisegundos
Desviación	8186 milisegundos

Memoria	652 MegaByte
CPU 1	57% de capacidad de procesamiento
CPU 2	41% de capacidad de procesamiento
CPU 3	14% de capacidad de procesamiento
CPU 4	13% de capacidad de procesamiento
Media	31.25% de capacidad de procesamiento

#### Resultados de la PRUEBA 02

Clientes simultáneos	3 clientes activos
Número de Muestras	144 consultas
Media	1253 milisegundos
Desviación	4692 milisegundos

Memoria	501 MegaByte
CPU 1	71 % de capacidad de procesamiento
CPU 2	74 % de capacidad de procesamiento
CPU 3	87 % de capacidad de procesamiento
CPU 4	86 % de capacidad de procesamiento
Media	79.5 % de capacidad de procesamiento

#### Resultados de la PRUEBA 03

Clientes simultáneos	6 clientes activos
Número de Muestras	288 consultas
Media	999 milisegundos
Desviación	3061 milisegundos

Memoria	650 MegaBytes
CPU 1	92 % de capacidad de procesamiento
CPU 2	91 % de capacidad de procesamiento
CPU 3	98 % de capacidad de procesamiento
CPU 4	98 % de capacidad de procesamiento
Media	94.75 % de capacidad de procesamiento

#### 5.4 Conclusiones

La plataforma tecnológica y las pruebas presentadas en el presente capítulo postulan el software desarrollado como un producto de calidad, ya que ha sido desarrollado con herramientas que exigen la utilización de mejores prácticas, sumado a esto cumple todos los requisitos funcionales y no funcionales. Si bien este documento está casi al final del proceso de desarrollo de software, es de suma importancia ya que garantiza el correcto funcionamiento del producto desarrollado y presenta el software como representación tangible de los objetivos planteados en la visión del proyecto, junto con todos los requerimientos que hicieron posible modelar esta solución.



## **CAPÍTULO VI - GESTIÓN DEL PROYECTO**

### **6.1 Introducción**

En este capítulo presentaremos como se planeó, organizó y coordinó el presente proyecto de investigación y desarrollo. Así mismo se presentan todos los factores que hacen que el presente proyecto sea viable desde el punto de vista técnico y económico, ya que se presentan de manera específica los costos en recursos tecnológicos, y humanos. Así mismo se presenta el cronograma del proyecto junto con los principales entregables del proyecto, así mismo se toma especial cuidado en la gestión de riesgos y en la gestión de la solicitud de cambios a fin de garantizar de manera adecuada el alcance del proyecto. Todos estos entregables pertenecientes a la gestión del proyecto garantizan que el proyecto de desarrollo es realizado en función a las mejores prácticas planteadas en la ingeniería de software cumpliendo todos los objetivos planteados inicialmente en la visión y alcance del proyecto.

### **6.2 Viabilidad del proyecto**

#### **6.2.1 Viabilidad técnica**

##### **Plataforma actual del negocio**

A fin de que un sistema de monitoreo vehicular sea implementado, es necesario contar con una infraestructura que pueda soportar cualquier evento que pueda interrumpir el servicio de monitoreo.

En la actualidad la TELCO cuenta con una estación terrena, donde se realiza la conexión con los diversos satélites que maneja la compañía, esta estación terrena posee la infraestructura necesaria para asegurar la comunicación entre las diversas oficinas de la TELCO y sus clientes, es capaz de soportar movimientos telúricos de grado 9 en la escala de Richter y cuenta con un suministro de energía eléctrica para soportar cualquier ausencia de energía hasta ser restaurada.

Así mismo, la TELCO cuenta con una infraestructura de redes basadas en fibra óptica a fin de optimizar la comunicación entre sus diversas instalaciones y entregar de manera oportuna la información solicitada.

En este marco de seguridad asegura el correcto funcionamiento y servicio continuo de un sistema de monitoreo, junto con los respectivos equipos de hardware, hace posible la puesta en marcha del sistema producto de este proyecto de investigación.

## 6.2.2 Viabilidad económica

A fin de sustentar la viabilidad económica, se detallaran los recursos y sus respectivos costos, a fin de determinar el ROI.

### Recursos humanos

**Responsable del Proyecto por parte del área de Desarrollo:** Es el responsable manejar el proyecto, desde los requerimientos, la implementación y la puesta en marcha. Los roles que representa este recurso son:

Gerencia del Proyecto (por parte del área de TI)

Arquitecto de Software

Ingeniero de pruebas

Las habilidades o skills de este recurso deberán ser: Conocimiento en gestión de proyectos, diseño de arquitecturas SOA y experiencia en implementar casos de uso de prueba.

**Analista Desarrollador:** Es el responsable de recolectar los requisitos de los stakeholders, convierte los requisitos en casos de uso del sistema, casos de uso de análisis, casos de uso de diseño y componentes, los roles que representa este recurso son:

Analista de Requerimientos

Ingeniero de Casos de Uso (análisis y diseño)

Ingeniero de Componentes

Los skills o habilidades de este recurso deberán ser: Experiencia en gestión de requerimientos (Definición y manejo de la trazabilidad de requerimientos), experiencia en análisis y diseño de sistemas, experiencia en programación de software basados en arquitectura SOA.

**Programador en PHP:** Es el responsable de programar los scripts en PHP en función a los diagramas de diseño del sistema y los casos de uso del sistema. Los skills o habilidades que deberá contar este recurso deberá ser: Experiencia avanzada en PHP 5, experiencia implementación de proyectos basados en Zend Framework, conocimientos avanzados de Ajax, conocimientos en JQuery y manejo de XHTML con CSS.

**Administrador de Base de Datos PostgreSQL:** Es el responsable de implementar el diseño físico de la base de datos, tomando en cuenta cada punto del diagrama, así mismo deberá tener conocimientos en PostgreSQL a fin de programar procedimientos almacenados, deberá tener conocimientos avanzados en sistema UNIX, experiencia en manejo de cron tareas y administración de redes. Así mismo será responsable de instalar los principales componentes software que tendrá el servidor de aplicaciones.

**Ingeniero Electrónico:** Es responsable de utilizar el software y equipos de la TELCO a fin de enviar las tramas al servidor de aplicaciones, administra y maneja el servidor de envío de tramas y comandos.

### **Recursos de hardware**

1 Laptop Hp Professional con procesador Intel core i7 – asignada al responsable del proyecto.

1 PC HP con procesador Intel core i3 – asignada al analista desarrollador.

2 PCs HP con procesador Intel core i3 – asignada a los programadores.

1 PC HP con procesador Intel core i3 – asignada al administrador de la base de datos.

1 PC HP con procesador Intel core i3 –servirá como ambiente de desarrollo, manejo de versiones, pruebas funcionales y de stress.

1 HP Server (ver especificación en la sección 6.4.1) – servirá como servidor de aplicaciones

**Observación:** El servidor de envío de tramas y comandos es uso exclusivo de la TELCO y también sirve a otros sistemas de control, por lo tanto no se cuenta como recurso de la solución.

### **Recursos de software**

Todos los equipos utilizarán software libre con coste cero a excepto de la PC del analista desarrollador que contará con las siguientes características de software:

1. Windows XP Sp2
2. Rational Rose – IBM
3. Microsoft Office 2007
4. Avast antivirus

Los otros equipos contarán con la siguiente característica de software (excepto del servidor de aplicaciones):

1. Ubuntu 11 – Desktop
2. Open Office
3. Visual Paradigm – Modelador UML, abre archivos .mdl (de Rational Rose)
4. Inkscape - Editor gráfico
5. Open Project
6. Rabbit SVN
7. Xampp server (PHP + Apache y Mysql)
8. PgAdmin – Herramienta de administración de PostgreSQL

### **Sobre los muebles e inmuebles y otros medios de comunicación**

Los costos de inmuebles, muebles para computadoras, conexión a internet, luz, y otros servicios no serán contados dentro del presente análisis debido a que estos costos fueron absorbidos de manera puntual por el área de desarrollo, no obstante se calculó que estos costos representarán el 5% del costo total.

### **Tiempo de implementación del proyecto**

El proyecto en total, tuvo una duración de 3 meses, desde la concepción, diseño, implementación y puesta en marcha.

### **Costo de los recursos**

Recurso Humano	Mes 1	Mes 2	Mes 3	Sub-Total
Responsable del Proyecto	\$ 1,500.00	\$ 1,500.00	\$ 1,500.00	\$ 4,500.00
Analista Desarrollador	\$ 800.00	\$ 800.00	\$ 800.00	\$ 2400.00

Programador 1		\$ 700.00	\$ 700.00	\$ 2,100.00
Programador 2		\$ 700.00		\$ 700.00
Administrador de Base de Datos		\$ 1,200.00	\$ 1,200.00	\$2,400.00

**Sub-Total \$ 12,100.00**

Recurso Hardware / Software	Cantidad	Precio U.	Sub-Total
Laptop Hp Professional con procesador Intel core i7	1	\$ 1,500.00	\$ 1,500.00
PC HP con procesador Intel core i3	5	\$ 800.00	\$ 4,000.00
Servidor de Aplicaciones	1	\$ 5,000.00	\$ 5,000.00
Windows XP – Professional	1	\$ 200.00	\$ 200.00
Rational Rose	1	\$ 445.00	\$ 445.00
Microsoft Office	1	\$ 170.00	\$ 170.00

**Sub-Total \$ 11,315.00**

#### Cálculo del Costo total del Proyecto

Costo de Recursos Humanos	\$12,100.00
Costo de Recursos de Hardware/Software	\$11,315.00
Costos de Inmuebles, muebles, Administrativos y otros (%5)	\$1,190.75
<b>Costo total del Proyecto</b>	<b>\$24,585.75</b>

Costo total del proyecto: \$24 585.75 dólares americanos.

#### Cálculo del retorno de la inversión

La TELCO estima ofrecer el servicio de monitoreo vehicular a sus clientes que posean equipos de localización, a fin de afrontar el costo del proyecto, sumará un cargo adicional por el uso del servicio de monitoreo vehicular por cada unidad con dispositivo de localización, para esto el estimado en 3 años es la siguiente:

- Año 1: 500 unidades
- Año 2: 1000 unidades
- Año 3: 1500 unidades

El cargo adicional anual por cada unidad vehicular que utilice el servicio de monitoreo vehicular será de \$30 dólares, para lo cual se espera obtener:

- Año 1: 500 unidades \* \$30 = \$15 000 dólares americanos
- Año 2: 1000 unidades \* \$30 = \$30 000 dólares americanos
- Año 3: 1500 unidades \* \$30 = \$45 000 dólares americanos

Según el estimado, a fines del segundo año la inversión en el proyecto será retornada, se estima a fines del segundo año y no a mediados de este, debido a que existe un costo por ventas y un costo de administración del software que será variable en función a las unidades que utilicen el sistema.

**Tiempo estimado de Retorno de inversión: 2 años**

Ingresos medios = [Año 1 (\$15 000) + Año 2 (\$30 000) + Año 3 (\$45 000) ] /3

**ROI = [(\$90 000 - \$24 585.75) / \$24 585.75] = 2.660%**

### **6.2.3 Viabilidad legal**

La implementación del presente proyecto es totalmente factible desde el punto legal, bajo la salvedad que es de obligación de la TELCO entregar todas las transmisiones emitidas por los dispositivos de localización al Ministerio de Transportes y Comunicaciones cada 5 minutos, para cumplir dicha exigencia, existe un proceso externo al sistema de monitoreo que lee la base de datos de transmisiones y envía las tramas requeridas en el formato solicitado (XML) por el Ministerio de Transportes y Comunicaciones.

## 6.3 Organización del proyecto

### 6.3.1 Organigrama del proyecto

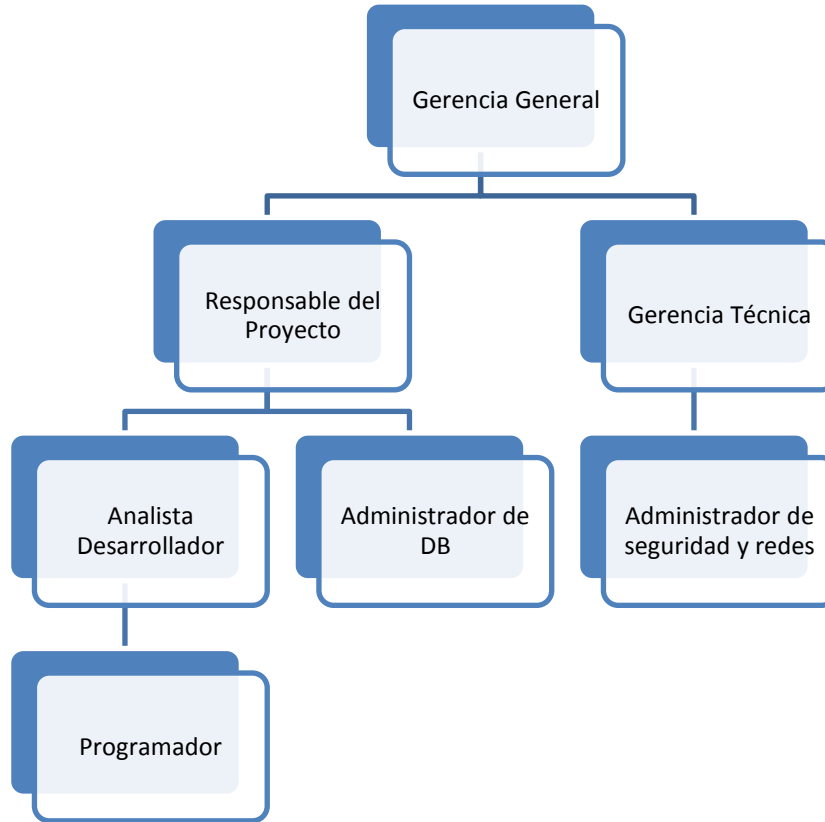


Figura 6.1 Organigrama del Proyecto

### 6.3.2 EDT del proyecto

El EDT del proyecto presenta las actividades y documentación presente en el proyecto y organizado por las disciplinas de la ingeniería del software.

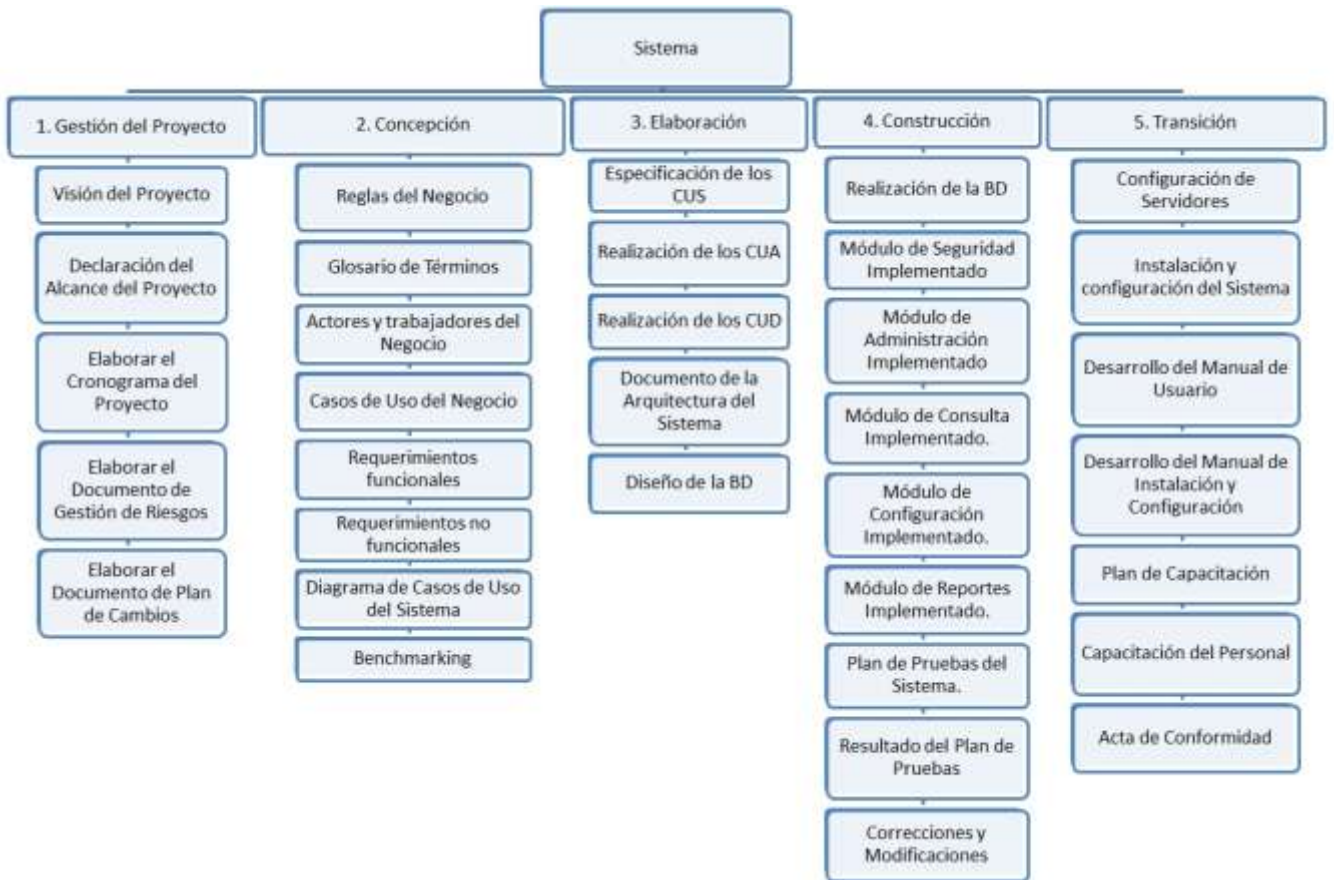


Figura 6.2 EDT del Proyecto

## 6.4 Cronograma de Ejecución del Proyecto

## 6.5 Gestión de riesgos del proyecto

### Propósito

El propósito de la gestión de riesgos es anticipar los riesgos que de alguna forma puedan afectar los tiempos, programación o calidad del software a desarrollar, y proponer estrategias para evitar o minimizar el riesgo.

### Lista de riesgos predecibles

ID	Tipo	Riesgo	Descripción	Probabilidad	Efecto	Indicador
RP1	Personal	Rotación de Personal	Stakeholders del Proyecto se retiran de la empresa	Media	Severo	Número de Requerimientos que cambian
RP2	Personal	Stakeholders no capacitados	Stakeholders sin experiencia en los procesos del negocio son asignados como parte del proyecto	Alta	Catastrófico	Número de requerimientos que no son necesarios
RP3	Personal	El Personal del Proyecto se enferma	Uno o más miembros del proyecto se enferman y no se pueden realizar las actividades planificadas.	Media	Severo	Número de actividades no implementadas
RP4	Tecnológico	No disponibilidad del Hardware	El Hardware requerido para el desarrollo del proyecto no es entregado a tiempo	Media	Catastrófico	Número de actividades no implementadas
RP5	Tecnológico	No disponibilidad de GPSs para realizar pruebas	Los GPSs y los automóviles para realizar las pruebas no están disponibles en las fechas indicadas	Alta	Severo	Número de funcionalidades no probadas
RP6	Requerimientos	Cambio de Requerimientos	Se solicitan cambios de requerimientos a mitad del desarrollo del software	Alta	Severo	Número de nuevos requerimientos
RP7	Requerimientos	Retraso en la aceptación de especificaciones	EL documento de especificaciones no es aceptado por los Stakeholders y no es posible desarrollar los requerimientos	Alta	Severo	Número de requerimientos no implementados



RP8	Organizacional	Reducción del Presupuesto	La TELCO reduce el presupuesto inicialmente aprobado.	Baja	Severo	Falta de liquidez para pagar al personal
-----	----------------	---------------------------	---	------	--------	--

ID	Riesgo	Estrategia
RP1	Rotación de Personal	Tener otra persona que cumpla las mismas funciones pero que esté involucrada en el tema y las actividades que se estén trabajando.
RP2	Stakeholders no capacitados	Realizar una entrevista a los stakeholders al inicio del proyecto y pedirles que describan los procesos en los que están involucrados, seguido a eso comparar dichas descripciones con otro personal que labora en la TELCO.
RP3	El Personal del Proyecto se enferma.	Tener otra persona que cumpla las mismas funciones pero que esté involucrada en el tema y las actividades que se estén trabajando.
RP4	No disponibilidad del Hardware	Solicitar el hardware más urgente al inicio del proyecto y solicitar cotizaciones a los proveedores, en caso que los responsables de la compra demoren, ejecutar las órdenes de compra inmediatamente.
RP5	No disponibilidad de GPSs para realizar pruebas	Solicitar a la TELCO permiso para probar el software con un cliente piloto que tenga GPSs y autos.
RP6	Cambio de Requerimientos	Solicitar la firma de los stakeholders cuando aprueben las especificaciones y pedir amplitud de tiempo y recursos en caso de cambios.
RP7	Retraso en la aceptación de especificaciones	Considerar el retraso de aceptación de especificaciones como un retraso del proyecto y solicitar recursos en tiempo y dinero.
RP8	Reducción del Presupuesto	Solicitar la asignación del total del presupuesto al inicio del proyecto.

### **Lista de riesgos no predecibles**

<b>ID</b>	<b>Tipo</b>	<b>Riesgo</b>	<b>Descripción</b>	<b>Probabilidad</b>	<b>Efecto</b>
RN1	Natural	Desastre Natural	Catástrofe natural que afecte recursos físicos y tecnológicos	Baja	Catastrófico
RN2	Organizacional	Problemas financieros en la organización	La TELCO determina no invertir en el proyecto de investigación por tener otras necesidades financieras.	Baja	Severo

<b>ID</b>	<b>Riesgo</b>	<b>Estrategia</b>
RN1	Desastre Natural	Ejecutar el proyecto en la estación satelital de la TELCO que soporta terremotos de grado 9 y otros desastres naturales.
RN2	Problemas financieros en la organización	Presentar el proyecto de investigación como un proyecto de inversión a la TELCO, donde obtendrá beneficios a mediano plazo.

## **6.6 Plan de cambios del negocio**

### **6.6.1 Procedimiento de solicitud de cambios**

#### **Nomenclatura de los cambios**

La nomenclatura para las Solicitudes de Cambio estará definida por un valor alfanumérico con tres secciones como sigue: SC\_TTS\_9999 , donde SC significa: Solicitud de Cambio, las TTS corresponderán a las siglas que identificarán al proyecto. Finalmente, 9999 corresponde al número consecutivo de Solicitud de Cambio para el proyecto, que estará en el rango de 0001 a 9999.

#### **Documentación**

Las solicitudes de cambio se harán mediante la Plantilla de Solicitudes de Cambio (Ver Anexo).

#### **Comunicación y seguimiento**

Una vez que la solicitud del cambio haya sido realizada, se comunicará al responsable del proyecto y al sponsor del proyecto acerca del cambio solicitado, el responsable del proyecto tendrá 2 días para determinar el impacto en tiempo y recursos que tomará el cambio solicitado, la “Plantilla de Solicitud de Cambio” será enviada vía correo electrónico al responsable del proyecto, con copia al sponsor del proyecto, así mismo el estado del cambio solicitado será comunicado vía correo electrónico.

### 6.6.2 Estado de un cambio solicitado

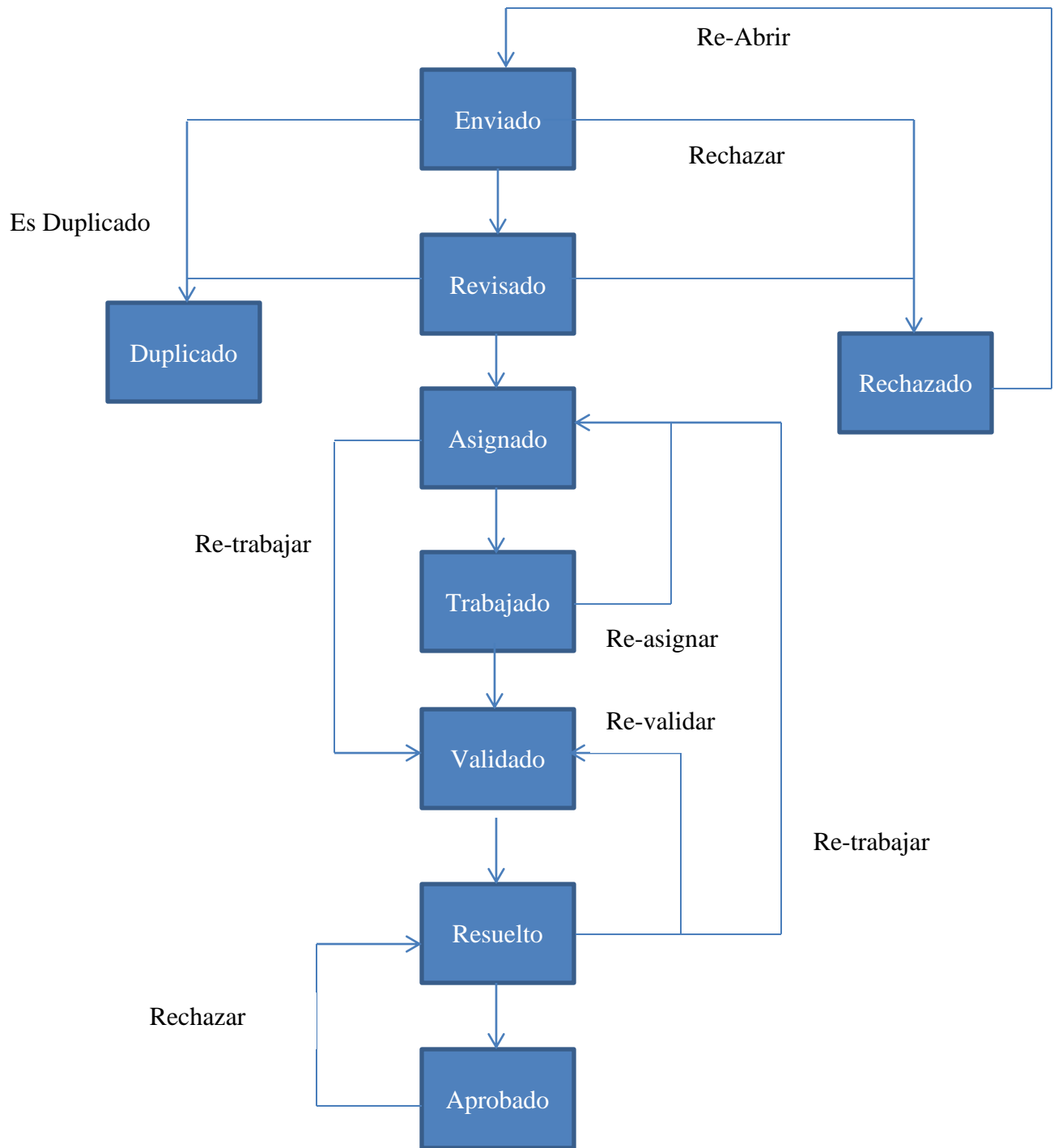


Figura 6.3 Diagrama de estados de un cambio solicitado

### 6.6.3 Matriz de acción y responsabilidades

Acción	Descripción	Responsable
Enviar	Consiste en enviar la solicitud de cambio mediante el procedimiento de solicitud de cambio, toda solicitud tendrá como estado inicial enviada.	Usuario Final, Stakeholder, Sponsor del Proyecto Responsable del Proyecto
Revisar	Consiste en revisar la solicitud de cambio y determinar si es técnicamente factible y no es el duplicado de otra solicitud de cambio.	Responsable del Proyecto Analista Desarrollador
Duplicado	Consiste en determinar que una solicitud de cambio es duplicado de otra ya enviada.	Responsable del Proyecto Analista Desarrollador
Rechazar	Cuando una solicitud de cambio no pueda ser implementada por alguna razón, esta será rechazada explicando la razón.	Responsable del Proyecto
Asignada	Es la designación de la implementación del cambio a algún miembro del equipo de desarrollo.	Responsable del Proyecto
Trabajada	Es la implementación o codificación del cambio solicitado, lo que significa que el cambio se está realizando.	Analista Desarrollador Programador Administrador de BD
Validada	Consiste en pasar la etapa de pruebas de la solicitud de cambio en función a los requerimientos nuevos.	Analista Desarrollador Stakeholder Usuario Final Responsable del Proyecto
Resuelto	Consiste en determinar que una solicitud de cambio ha sido completamente implementada en función a los requerimientos establecidos, y pasa a producción.	Stakeholder Usuario Final Responsable del Proyecto
Aprobado	Consiste en dar por aprobado el cambio implementado.	Stakeholder Usuario Final Responsable del Proyecto

## **6.7 Constancia de aceptación del cliente sobre el proyecto**

La constancia de aceptación del cliente sobre el proyecto es un documento que certifica que el proyecto fue entregado en función a los requerimientos solicitados, cumpliendo los objetivos funcionales planteados inicialmente.

### **Constancia de Aceptación del Proyecto de Tecnología de Información**

**Referencia:** Área de Desarrollo y Proyectos

**Responsable:** Rubén Bocanegra Ureta

**Proyecto:** Sistema de rastreo satelital vehicular

**Entrega de:** Entrega del proyecto implementado al 100%

**Fecha:** 01 Septiembre del 2011

Mediante la aceptación de la presente constancia el representante de la TELCO certifica que el proyecto “Sistema de rastreo satelital vehicular” fue entregado en su totalidad en función a los requerimientos planteados, ser sometidos a las pruebas respectivas y cumplir los objetivos funcionales inicialmente planteados.

**Rubén Bocanegra Ureta**

Área de Desarrollo TI

## **6.8 Conclusiones**

La gestión del proyecto presentada en el presente capítulo abarca los principales entregables a tomarse en cuenta en un proyecto de desarrollo serio y de calidad. Si bien el proyecto de investigación fue desarrollado de manera independiente por la TELCO y por lo tanto existen otros documentos que han sido obviados, los más importantes han sido presentados e ilustran la forma en que fue gestionado el proyecto. Desde la viabilidad técnica, sustentación de costos con un ROI de 2.660% (ver punto 6.2.2) y gestión de riesgos, se ha tomado especial interés en hacer de la gestión un pilar fundamental del presente proyecto, ya que salvaguarda el documento inicial de visión (Ver Capítulo 1) hace factible el desarrollo de todas las funcionalidades inicialmente planteadas. La gestión del proyecto si bien es una disciplina más de la ingeniería de software, cuida que el proyecto en todo su ciclo de vida cumpla las exigencias requeridas de manera ordenada y objetiva.

# **CAPÍTULO VII - CONCLUSIONES Y RECOMENDACIONES**

## **7.1 Introducción**

El presente capítulo, es la parte final de la presente investigación aplicada donde se hace un recuento del trabajo de investigación, análisis, diseño e implementación que se hizo en este proyecto, y producto de ese recuento de experiencias podemos llegar a conclusiones que se contrastan con los objetivos que dieron inicio al presente proyecto de investigación.

## **7.2 Conclusiones**

1. En la presente investigación, se llegó a conocer de manera más profunda los términos y conceptos que engloba el servicio de monitoreo vehicular, así mismo se conceptualizó las principales definiciones del proceso de monitoreo vehicular. Por lo tanto el objetivo específico de la presente investigación que era conocer los conceptos más importantes del servicio de monitoreo vehicular fue logrado. (Ver sección 1.3.1 del Capítulo 1)
2. Así mismo el objetivo específico de conocer las funcionalidades más representativas de las aplicaciones de monitoreo vehicular en diversas industrias, fue alcanzado, ya que se dieron a conocer las aplicaciones de monitoreo vehicular más representativas y se preparó una comparativa de funcionalidades. (Ver sección 3.5.2.5 del Capítulo 3)
3. Y por último la necesidad de modelar una solución software en plataforma web, que cumpla con las exigencias más importantes del servicio de monitoreo vehicular fue también alcanzado, ya que se concibió en el Capítulo 2 y 3, una solución web que abarque las funcionalidades que se exigen en los procesos del negocio.
4. Por lo tanto el objetivo principal de esta investigación aplicada fue alcanzado, debido a que producto del trabajo realizado a través del proceso de desarrollo de la presente investigación, se implementó una solución web de monitoreo vehicular que da servicio a múltiples clientes sin necesidad de instalaciones personales, solo con el uso de un navegador web, los usuarios pueden gozar de las más representativas funcionalidades que una solución de monitoreo puede brindar de manera inmediata. Actualmente el sistema cuenta con 95 clientes y 200 unidades transmitiendo todos los días, siendo la principal herramienta de monitoreo de muchas de ellas.

## 7.3 Recomendaciones

1. A raíz de la presente investigación, se ha detallado en el proceso de desarrollo de software de la presente solución, que posee la mayoría de las funcionalidades de las aplicaciones más reconocidas del mercado en diferentes industrias, pero esto no habría sido posible sin un correcto proceso de desarrollo, por lo tanto, se recomienda siempre llevar a cabo el proceso de desarrollo de software, desde el modelado del negocio hasta las pruebas a fin de garantizar un producto de calidad y que cumpla las expectativas de los usuarios.
2. Se ha demostrado que una aplicación de monitoreo en plataforma web puede utilizarse inmediatamente en un centro de control, independientemente de la cantidad de unidades que se desee monitorear, por lo tanto se recomienda tomar en cuenta soluciones en plataforma web a fin de ahorrar tiempo de implantación y ejecutar tareas inmediatas.
3. Así mismo, se ha dado a conocer cómo implementar una solución de monitoreo vehicular, no obstante, la presente investigación puede servir como plataforma para otro tipo de soluciones como monitoreo de rutas, gestión logística, entre otros. Por lo tanto se recomienda tomar en cuenta el diseño de la presente solución software como fundamento para soluciones de mayor envergadura que cumplan requisitos específicos. En esta última recomendación podemos considerar como valioso el aporte de agrupar las unidades vehiculares por flotas y clientes, asignándole permisos de monitoreo a los usuarios del sistema por cada flota, el modelado de clases de tipo entidad y la base de datos pueden llegar a ser la materia prima para soluciones de mayor complejidad o especialización en el mercado.

## BIBLIOGRAFÍA

[ABSOLUTE SOFTWARE 11] ABSOLUTE SOFTWARE Inc. Absolute Software – Official Website <http://www.absolutesw.com>, 2011

[BAO-YEN 00] JAMES BAO-YEN TSUI Fundamentals of Global Positioning System Receivers A Software Approach John Wiley & Sons, Inc. 2000.

[BELL 05] A. T. BELL Telecommunications Billing Virtual Book Worm. 2005

[BONHAM-CARTER 06] GRAEME F. BONHAM-CARTER Geographic Information Systems for Geoscientists: Modeling with GIS Elsevier Butterworth – Heinemann Linacre House, Jordan Hill Oxford OX2 8DP. 2006

[CRAWFORD 03] WILLIAM CRAWFORD J2EE design patterns O'REILLY. 2003

[DERNIEAME 99] JEAN CLAUDE DERNIAME Software process: principles, methodology, and technology Springer. 1999.

[GLOBALSTAR 11] GLOBALSTAR Inc Official Website [www.globalstar.com](http://www.globalstar.com), 2011

[GPS INSIGHT 11] GPS INSIGHT LLC Tracking System Product Brochure [www.gpsinsight.com](http://www.gpsinsight.com), 2011

[IRIDIUM 11] [IRIDIUM COMMUNICATIONS Inc Official Website www.iridium.com](http://www.iridium.com), 2011

[JABLONSKI 04] STEFAN JABLONSKI Guide to web application and platform architectures Springer. 2004.

[JACOBSON 00] IVAR JACOBSON The road to the unified software development process Cambridge University Press. 2000.

[KAPLAN 06] ELLIOT D. KAPLAN Understanding GPS: Principles and applications. Second Edition British Library Cataloguing in Publication Data. 2006

[LARMAN 04] CRAIG LARMAN UML y Patrones: Introducción al análisis y diseño orientado a objetos. Prentice Hall. 2004.

[IU 08] LI LIU Apparatus and method of map engine clustering automation Patent Application Publication - United States. 2008

[LONGLEY 05] PAUL A. LONGLEY Geographic Information Systems and Science Wiley. 2005.

[LOUIS 02] P. J. LOUIS Telecom management crash course: managing and selling Telecom services and products McGraw-Hill Professional. 2002.



[MEDAGAMA 08] MICHAEL MEDAGAMA GIS/GPS/GPRS and Web based Framework for Vehicle Fleet Tracking The Institution of Engineers, Sri Lanka ENGINEER - Vol. XXXXI, No. 05, pp. 28-33. 2008

[MILLER 06] CHRISTOPHER C. MILLER A Beast in the Field: The Google Maps Mashup as GIS/2 Texas A&M International University, USA. 2006

[POZO-RUZ 00] A. POZO-RUZ Sistema de Posicionamiento Global (GPS): Descripción, Análisis de errores, aplicaciones y futuro. Departamento de Tecnología Electrónica. Universidad de Málaga. 2000.

[RAMOS-BOSCH 08] PERE RAMOS – BOSCH Improvements in autonomous GPS navigation of Low Earth Orbit Satellites Departamento de Física y Matemáticas aplicadas de la Universidad Politécnica de Catalunya, España. 2008.

[REIF 97] JOE REIF Services -- The export of 21th Century: A guidebook for US service exporters Northern California District Export Council - USA. 1997.

[RODRIGUEZ-SOLANO 99] ROBERTO RODRIGUEZ-SOLANO SUAREZ Las Técnicas GPS como herramienta en la gestión ambiental Observatorio Medioambiental - España. 1998.

[SANTIAGO QUINTANA 08] SANTIAGO MANCEBO QUINTANA Libro SIG: aprendiendo a manejar los SIG en la gestión ambiental. Madrid - España. 2008. [SCOPE TECHNOLOGIES 11] SCOPE TECHNOLOGIES Scope Techonolgies Holdings Limited - Official Website <http://scopetechnology.com>. 2011

[SHKLAR 03] SHKLAR LEON Web Application architecture: principles, protocols and practices Jhon Wiley and Sons. 2003.

[SPOT 11] Globalstar Inc. Find me Spot - Official Website <http://www.findmespot.com>, 2011

[TAT-JEN 07] TAT JEN CHAM Advances in multimedia modeling: 13<sup>th</sup> International Multimedia Modeling Conference, January 9 -12, Singapore. 2007

[THANGAVELU 07] ARUNKUMAR THANGAVELU Location and Identification and Vehicle Tracking using VANET (VETRAC) IEEE – ICSCN- MIT Campus, Anna University, Chennai, India. 2007

[ZAMAN 02] KHAWAR ZAMAN Developing enterprise Java applications with J2EE and UML Addison-Wesley Professional. 2002

## GLOSARIO

**Altitud:** La altitud es la distancia vertical a un origen determinado, considerado como nivel cero, para el que se suele tomar el nivel medio del mar.

**Aplicación Web:** En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

**Canal de Comunicación:** Un canal de comunicación es el medio de transmisión por el que viajan las señales portadoras de la información emisor y receptor. Es frecuente referenciarlo también como canal de datos.

**Conexión Satelital:** Tecnología basada en el GPS, la cual le permite la comunicación desde un área geográfica inaccesible, (una jungla o una reserva forestal) mediante un teléfono satelital.

**Geocerca:** Es un área o perímetro virtual en una zona geográfica

**Google Maps:** Google Maps es el nombre de un servicio gratuito de Google. Es un servidor de aplicaciones de mapas en 2D. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero.

**Ingeniería de Software:** Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software.

**Latitud:** La latitud es la distancia angular entre el ecuador y un punto determinado del planeta medida a lo largo del meridiano que pasa por ese punto.

**Longitud:** Distancia expresada en grados, entre el meridiano de un punto y otro tomado como referencia en el Ecuador.

**Método Dúplex:** En el que dos estaciones A y B, actúan como fuente y colector, transmitiendo y recibiendo información simultáneamente. Permite la transmisión en ambas direcciones y de forma simultánea. Por ejemplo una conversación telefónica.

**Método Semi-dúplex:** Es aquel en el que una estación A en un momento de tiempo, actúa como fuente y otra estación correspondiente B actúa como colector, y en el momento siguiente, la estación B actuará como fuente y la A como colector. Permite la transmisión en ambas direcciones, aunque en momentos diferentes. Un ejemplo es la conversación entre dos radioaficionados, pero donde uno espera que el otro termine de hablar para continuar el diálogo.

**Método Simplex:** Es aquel en el que una estación siempre actúa como fuente y la otra siempre como colector. Este método permite la transmisión de información en un único sentido.

**Open Source:** Código fuente abierto software libre, se refiere a un programa cuyo código fuente está disponible al público general, gratis, para usar y modificar.

**Operador de Telecomunicaciones:** Un operador de telecomunicaciones es cualquier

empresa legalmente constituida que cuenta con las licencias, permisos y garantías exigidas por Ley para ejercer actividades de telecomunicaciones con consumidores finales personas físicas u otras personas jurídicas (administraciones, empresas, organismos, etc.).

**Punto de Interés:** Es una marca visualmente fácil de identificar en un mapa, la cual representa la posición de un lugar de relevancia.

**Rastreo vehicular automatizado:** El sistema de Rastreo Vehicular Automatizado (RVA) o AVL, acrónimo de Automatic Vehicle Location, se aplica a los sistemas de localización remota en tiempo real, basados generalmente en el uso de un GPS y un sistema de transmisión que es frecuentemente un módem inalámbrico. El sinónimo europeo es Telelocalización.

**RMI:** Invocación de métodos remotos (Remote Method Invocation), consiste en que un objeto acceda a un método (una de las funcionalidades) de otro objeto remoto (que esté situado en otro punto de una red).

**RTP:** son las siglas de Real-time Transport Protocol (Protocolo de Transporte de Tiempo real). Es un protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real.

**Sistema de posicionamiento global:** Es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión.

**Stakeholders: Son** los interesados directos e indirectos de una empresa que teniendo algún tipo de interés en las operaciones empresariales, le brindan su apoyo y ante los cuales la organización es responsable, tales como los accionistas, proveedores, clientes, usuarios de sistemas, entre otros.

**Trama:** En monitoreo de unidades vehiculares una trama, es una unidad de envío de datos, que equivale a una cadena de caracteres donde detalla la posición actual del vehículo junto con otra información anexa.

**Wi-Fi:** Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: un ordenador personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica. Dicho punto de acceso tiene un alcance de unos 20 metros (65 pies) en interiores y al aire libre una distancia mayor. Pueden cubrir grandes áreas la superposición de múltiples puntos de acceso.

## SIGLARIO

**AJAX:** Asynchronous JavaScript And XML.

**API:** Application Programming Interface. **BD:**

Base de Datos

**CPU:** Central Processing Unit.

**CUA:** Caso de Uso de Análisis. **CUD:**

Caso de Uso de Diseño.

**CUN:** Caso de Uso del Negocio. **CUS:**

Caso de Uso del Sistema. **CSV:** Comma-

Separated Values. **FAA:** Agencia de

Aviación Federal **GB:** Gigabyte.

**GIS:** Geographic Information System

**GPRS:** General Packet Radio Service (GPRS) o servicio general de paquetes vía radio.

**GPS:** Global Positioning System.

**IDE:** Integrated Development Environment.

**HTML:** HyperText Markup Language.

**IU:** Interfaz de Usuario.

**JSON:** JavaScript Object Notation.

**KML:** Keyhole Markup Language.

**M.T.C.:** Ministerio de Transportes y Comunicaciones.

**MVC:** Model-View-Controller.

**PC:** Personal Computer.

**PDF:** Portable Document Format

**PHP:** Personal Home Pages.

**OSIPTEL:** Organismo Supervisor de Inversión Privada en Telecomunicaciones.

**ROI:** Return On Investment.

**TELCO:** Telecommunications Company.

**TI:** Tecnologías de Información.

**WIFI:** Wireless Fidelity.

**WWW:** World Wide Web.

**XML:** Extensible Markup Language.