

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 8

Análisis Diseño e Implementación de la Capa de Presentación de los Sub-Módulos Persona y Vehículo pertenecientes al Subsistema de Análisis de Información del SIIPOL

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

Autor: José Carlos Guevara Turruelles.
Tutor: Ing. Osdalme Fuente Colina.
Co-Tutor: Ing. Irina Cancela Nieto.

Ciudad de la Habana, Mayo de 2009

*Si piensas que vales lo que sabes, estás muy equivocado.
Tus conocimientos de hoy no tienen mucho valor más allá de un par de años.
Lo que vales es lo que puedes llegar a aprender, la facilidad con la que te adaptas
a los cambios que esta profesión nos regala tan frecuentemente.*

José M. Aguilar

DECLARACIÓN DE AUTORÍA

Declaro que soy autor de este trabajo y autorizo a la Facultad 8 de la Universidad de las Ciencias Informáticas así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2009.

AUTOR

José Carlos Guevara Turruelles

TUTOR

Ing. Osdalme Fuentes Colina

CO-TUTOR

Ing. Irina Cancela Nieto

AGRADECIMIENTOS

Si agradezco a todos los que han realizado algún aporte que me haya ayudado a llegar a esta meta, tendría que elaborar otro documento. No podría pasar por alto a mi madre, gracias mamá por tanto esfuerzo y sacrificio, esta es una de las recompensas. Yanet, que sería de mi sin ti, hermanita tengo mucho por lo que agradecerte. Paipa no te quedas atrás, a ti también tengo mucho que agradecer. Osdalme, tremendo tutor, gracias por toda la ayuda. Irina gracias también por ser una buena co-tutora. Diana, Miriam, Vladimir, Susana y Eric, gracias por ser otra familia para mí, por todo el apoyo y hospitalidad que me han brindado. Gracias piquete por estar presentes en las buenas y en las malas, no voy a mencionar nombres porque tampoco alcanzaría el documento, pero todos ustedes están aquí.

DEDICATORIA

*A mi familia,
esa que es un tanto pequeña en número, pero grande a la vez,
quien me ha dado todo lo necesario para alcanzar este logro.*

RESUMEN

El Cuerpo de Investigaciones Científicas, Penales y Criminalísticas de la República de Venezuela es una de las instituciones más importantes que lucha contra la criminalidad existente en ese país, utilizan el Sistema Integrado de Información Policial para gestionar digitalmente la información que manejan, pero este software ya se encuentra obsoleto y no sufraga las necesidades actuales de este cuerpo policial. El Proyecto CICPC es un contrato realizado entre las Repúblicas de Cuba y Venezuela para elaborar un sistema informático que cumpla con los requerimientos obtenidos durante el estudio del funcionamiento de dicha institución. En el presente trabajo se analizó, diseñó e implementó una fracción de la aplicación que cumplió con los requerimientos relacionados a la gestión de la información de personas y vehículos en la División de Información Policial y se generaron los artefactos correspondientes a cada uno de los flujos de desarrollo mencionados.

ABSTRACT

The CICPC of the Bolivarian Republic of Venezuela is one of the most important institutions that fight against crime at the Republic, using the “Sistema Integrado de Información Policial” to manage digital information, but this software is now obsolete and do not cover the current needs of this police force. CICPC Project is a contract made between Cuba and Venezuela to develop an informatics system according to the requirements collected during the study of that institution. In this investigation was analyzed, designed and implemented a fraction of the application that met the requirements related to information management of people and vehicles in the Division of Police Information and artifacts that have been generated for each workflow.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	4
1.1. INTRODUCCIÓN.....	4
1.2. SOFTWARE DE GESTIÓN POLICIAL.....	4
1.2.1 SISTEMA INTEGRADO DE INFORMACIÓN POLICIAL.....	5
1.2.2 DATAPOLICE.....	5
1.2.3 SISTEMA TERRITORIAL DE EMERGENCIAS Y GESTIÓN POLICIAL (STEGPOL).....	5
1.3. METODOLOGÍA, LENGUAJES Y HERRAMIENTAS DE DESARROLLO.....	6
1.3.1 EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....	7
1.3.2 LENGUAJE UNIFICADO DE MODELADO (UML).....	10
1.3.3 VISUAL PARADIGM PARA UML.....	10
1.3.4 JAVA PARA SOLUCIONES EMPRESARIALES (JAVA 2 ENTERPRISE EDITION O J2EE).....	11
1.3.5 JAVA SERVER FACES (JSF).....	12
1.3.6 JAVASCRIPT.....	14
1.3.7 ECLIPSE.....	15
1.4. ARQUITECTURA TÉCNICA.....	15
1.5. CONCLUSIONES.....	16
CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	18
2.1. INTRODUCCIÓN.....	18
2.2. PREVIO AL ANÁLISIS Y DISEÑO.....	18
2.2.1 INTRODUCCIÓN AL SIIPOL.....	18
2.2.2 SUB-MÓDULO PERSONA.....	18
2.2.3 SUB-MÓDULO VEHÍCULO.....	21
2.2.4 DISEÑO DE LAS INTERFACES DE USUARIO.....	23
2.3. MODELO DE ANÁLISIS.....	25
2.3.1 INTRODUCCIÓN AL FLUJO DE ANÁLISIS.....	25
2.3.2 CLASES DEL ANÁLISIS.....	25
2.3.3 DIAGRAMAS DE CLASES DEL ANÁLISIS.....	26
2.4. MODELO DE DISEÑO.....	27
2.4.1 INTRODUCCIÓN AL FLUJO DE DISEÑO.....	27
2.4.2 DIAGRAMA DE PAQUETES.....	28
2.4.3 DIAGRAMAS DE CLASES DEL DISEÑO.....	29

2.4.4	DESCRIPCIONES DE LAS CLASES.....	38
2.4.5	REALIZACIONES DE CASOS DE USO.....	50
2.4.6	DIAGRAMA DE CLASES PERSISTENTES.....	53
2.5	MODELO DE IMPLEMENTACIÓN	55
2.5.1	INTRODUCCIÓN AL FLUJO DE IMPLEMENTACIÓN.....	55
2.5.2	DIAGRAMAS DE SUBSISTEMAS DE IMPLEMENTACIÓN.....	55
2.5.3	DIAGRAMA DE COMPONENTES.....	56
2.5.4	ESTÁNDAR DE CODIFICACIÓN.....	57
2.6	CONCLUSIONES.....	58
3	VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	60
3.1	INTRODUCCIÓN.....	60
3.2	TIPOS Y NIVELES DE PRUEBA.....	60
3.2.1	PRUEBAS DE CAJA NEGRA.....	60
3.2.2	NIVELES DE PRUEBAS	61
3.3	RESULTADOS OBTENIDOS	62
3.4	CONCLUSIONES.....	63
	CONCLUSIONES	64
	RECOMENDACIONES	65
	REFERENCIAS BIBLIOGRÁFICAS	66
	BIBLIOGRAFÍA	67
	GLOSARIO DE TÉRMINOS	69
	ANEXOS.....	70
	ANEXO 1: DIAGRAMAS DE CLASES DEL ANÁLISIS DE LOS CU MÁS SIGNIFICATIVOS.	70
	ANEXO 2: REALIZACIÓN DE CASOS DE USO MÁS SIGNIFICATIVOS DEL ANÁLISIS.	71
	ANEXO 3: DIAGRAMAS DE CLASES DEL DISEÑO.....	73
	ANEXO 4: REALIZACIONES DE LOS CASOS DE USO MÁS SIGNIFICATIVOS DEL DISEÑO.	87
	ANEXO 5: DIAGRAMAS DE COMPONENTES DE IMPLEMENTACIÓN.....	91

ÍNDICE DE FIGURAS

Capítulo 1

FIG. 1.1 FASES Y FLUJOS DE TRABAJO DEL RUP.	9
FIG. 1.2 MODELO DE RELACIÓN ENTRE LOS TÉRMINOS DE JSF.....	13
FIG. 1.3 DIAGRAMA DE DESPLIEGUE.	16

Capítulo 2

FIG. 2.1 DIAGRAMA DE CU DEL SISTEMA PERTENECIENTES AL SUB-MÓDULO PERSONA.	19
FIG. 2.2 DIAGRAMA DE CU SIGNIFICATIVOS DEL SISTEMA PERTENECIENTES AL SUB-MÓDULO VEHÍCULO.	21
FIG. 2.3 DISEÑO DE INTERFAZ DE USUARIO PARA EL CU GESTIONAR PERSONA.	23
FIG. 2.4 DISEÑO DE INTERFAZ DE USUARIO PARA EL CU GESTIONAR VEHÍCULO.	24
FIG. 2.5 ESTEREOTIPO DE CLASES INTERFAZ.	25
FIG. 2.6 ESTEREOTIPO DE CLASES DE CONTROL.	25
FIG. 2.7 ESTEREOTIPO DE CLASES ENTIDAD.	26
FIG. 2.8 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU GESTIONAR PERSONA.	26
FIG. 2.9 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU CONSULTAR PERSONA.	26
FIG. 2.10 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU GESTIONAR VEHÍCULO.	27
FIG. 2.11 DIAGRAMA DE CLASES DEL ANÁLISIS DEL CU VER RELACIONES DE VEHÍCULO.	27
FIG. 2.12 DIAGRAMA DE PAQUETES DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO.	28
FIG. 2.13 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTE AL CU GESTIONAR PERSONA.	30
FIG. 2.14 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTE AL CU GESTIONAR PERSONA.....	31
FIG. 2.15 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTE AL CU CONSULTAR PERSONA.....	32
FIG. 2.16 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTE AL CU CONSULTAR PERSONA.	33
FIG. 2.17 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTE AL CU GESTIONAR VEHÍCULO.....	34
FIG. 2.18 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTE AL CU GESTIONAR VEHÍCULO.	35
FIG. 2.19 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTE AL CU VER RELACIONES DE VEHÍCULO.	36
FIG. 2.20 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTE AL CU VER RELACIONES DE VEHÍCULO.....	37
FIG. 2.21 DIAGRAMA DE CONTRATO ENTRE PAQUETES DEL CU GESTIONAR PERSONA: MODIFICAR.	51
FIG. 2.22 DIAGRAMA DE CONTRATO ENTRE PAQUETES DEL CU CONSULTAR PERSONA.....	52
FIG. 2.23 DIAGRAMA DE CONTRATO ENTRE PAQUETES DEL CU VER RELACIONES DE VEHÍCULO.	53
FIG. 2.24 DIAGRAMA DE CLASES PERSISTENTES.....	54
FIG. 2.25 DIAGRAMA DE SUBSISTEMAS DE IMPLEMENTACIÓN.	55
FIG. 2.26 DIAGRAMA DE COMPONENTES DE IMPLEMENTACIÓN PERTENECIENTE AL SUB-MÓDULO PERSONA.....	56

Capítulo 3

FIG. 3.1 GRÁFICO DE COLUMNAS DE PRUEBAS REALIZADAS A LA APLICACIÓN.	62
FIG. 3.2 GRÁFICO DE LÍNEAS DE PRUEBAS REALIZADAS A LA APLICACIÓN.	63

Anexo 1

ANEXO 1.1 DIAGRAMA DE CLASES DEL ANÁLISIS PERTENECIENTE AL CU VER RELACIONES DE PERSONA.	70
ANEXO 1.2 DIAGRAMA DE CLASES DEL ANÁLISIS PERTENECIENTE AL CU CONSULTAR VEHÍCULO.	70

Anexo 2

ANEXO 2.1 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR PERSONA: INCLUIR.	71
ANEXO 2.2 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR PERSONA: MODIFICAR.	71
ANEXO 2.3 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR PERSONA: VER.	71
ANEXO 2.4 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU CONSULTAR PERSONA.	71
ANEXO 2.5 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR VEHÍCULO: INCLUIR.	72
ANEXO 2.6 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR VEHÍCULO: MODIFICAR.	72
ANEXO 2.7 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU GESTIONAR VEHÍCULO: VER.	72
ANEXO 2.8 DIAGRAMA DE COMUNICACIÓN PERTENECIENTE AL CU CONSULTAR VEHÍCULO.	72

Anexo 3

ANEXO 3.1 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU VER RELACIONES DE PERSONA.	73
ANEXO 3. 2 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU VER RELACIONES DE PERSONA.	73
ANEXO 3.3 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU GESTIONAR RELACIONES DE PERSONA.	74
ANEXO 3.4 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU GESTIONAR RELACIONES DE PERSONA.	75
ANEXO 3.5 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR DATOS DE PERSONA SAIME.	76
ANEXO 3.6 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR DATOS DE PERSONA SAIME.	76
ANEXO 3.7 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR HOJA DE VIDA DE PERSONA.	77
ANEXO 3.8 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR DATOS DE PERSONA SAIME.	77
ANEXO 3.9 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU VER HISTÓRICO GENERAL DE PERSONA.	78
ANEXO 3.10 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU VER HISTÓRICO GENERAL DE PERSONA.	78
ANEXO 3.11 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR VEHÍCULO MANEJADO.	79
ANEXO 3.12 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR VEHÍCULO.	80
ANEXO 3.13 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR LUGARES RELACIONADOS A VEHÍCULO.	80
ANEXO 3.14 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR LUGARES RELACIONADOS A VEHÍCULO.	81

ANEXO 3.15 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR DATOS DE VEHÍCULO INTTT. ...	82
ANEXO 3.16 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR DATOS DE VEHÍCULO INTTT.....	82
ANEXO 3.17 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU CONSULTAR PROPIETARIOS DE VEHÍCULO INTTT.	83
ANEXO 3.18 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU CONSULTAR PROPIETARIOS DE VEHÍCULO INTTT.	83
ANEXO 3.19 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU REGISTRAR RECUPERACIÓN DE VEHICULO.	84
ANEXO 3.20 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU REGISTRAR RECUPERACIÓN DE VEHICULO.	85
ANEXO 3.21 DIAGRAMA DE CLASES DEL DISEÑO WEB PERTENECIENTES AL CU VALIDAR SERIAL DE CARROCERÍA.....	86
ANEXO 3.22 DIAGRAMA DE CLASES DEL DISEÑO PERTENECIENTES AL CU VALIDAR SERIAL DE CARROCERÍA.	86

Anexo 4

ANEXO 4.1 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU GESTIONAR PERSONA: INCLUIR.	87
ANEXO 4.2 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU GESTIONAR PERSONA: VER.	87
ANEXO 4.3 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU CONSULTAR PERSONA.	88
ANEXO 4.4 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU GESTIONAR VEHÍCULO: INCLUIR.	88
ANEXO 4.5 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU GESTIONAR VEHÍCULO: VER.....	89
ANEXO 4.6 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU GESTIONAR VEHÍCULO: MODIFICAR.	89
ANEXO 4.7 DIAGRAMA DE SECUENCIA PERTENECIENTE AL CU CONSULTAR VEHÍCULO.....	90

Anexo 5

ANEXO 5.1 DIAGRAMA DE COMPONENTES DE IMPLEMENTACIÓN WEB PERTENECIENTE AL SUB-MÓDULO PERSONA.....	91
ANEXO 5.2 DIAGRAMA DE COMPONENTES DE IMPLEMENTACIÓN PERTENECIENTE AL SUB-MÓDULO VEHÍCULO.....	92
ANEXO 5.3 DIAGRAMA DE COMPONENTES DE IMPLEMENTACIÓN WEB PERTENECIENTE AL SUB-MÓDULO VEHÍCULO.	93

INTRODUCCIÓN

La República Bolivariana de Venezuela se encuentra inmersa en un proceso de transformaciones sociales que abarca diferentes esferas, entre ellas: la salud, la educación y la seguridad ciudadana. A esta última se le ha dado una gran importancia debido al alto índice de criminalidad existente. Varios estudios han arrojado cifras muy preocupantes en los índices de criminalidad en las ciudades venezolanas. Una somera revisión de las tendencias recientes en la criminalidad oficial registrada en Venezuela muestra un incremento destacado en los rubros de delitos violentos. De acuerdo con las estadísticas oficiales, entre 1990 y 2000, la tasa de homicidios pasó de 13 a 33 por cada cien mil habitantes; del año 2000 al 2002 se incrementó esta cifra en un 21%. ⁽¹⁾ Por esta razón se ha comenzado a realizar una serie de cambios en el Ministerio del Poder Popular para las Relaciones del Interior y Justicia, al que pertenece el Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC) tratando por todos los medios posibles de mejorar la situación actual. Para esto, a propósito de los convenios de la Alternativa Bolivariana para América Latina y el Caribe (ALBA) se han formalizado contratos entre la República de Cuba y la República Bolivariana de Venezuela. Dichos acuerdos están enmarcados en diferentes esferas productivas, dentro de las cuales se incluye la producción de software. La Universidad de las Ciencias Informáticas lleva la delantera en esta rama de la producción y a ella se le han asignado la materialización de varios productos de software y la prestación de servicios informáticos. Entre ellos se encuentra el Sistema de Investigación e Información Policial (SIIPOL) que se ha creado para colaborar en la lucha del gobierno venezolano contra la criminalidad y como software sustituto al que dispone actualmente el CICPC.

El CICPC es la institución que se encarga de investigar y registrar toda la información referente a los delitos que se realizan en la República Bolivariana de Venezuela. Cuenta actualmente con un Sistema Integrado de Información Policial que lleva muchos años de funcionamiento. La infraestructura que lo soporta y las técnicas con que fue desarrollado son actualmente obsoletas, motivo por el cual la aplicación no brinda la rapidez necesaria para estos tipos de sistemas; carece de muchas funcionalidades precisas para facilitar el trabajo de los funcionarios y no cuenta con áreas de trabajo definidas, esto trae como resultado que los funcionarios tienen acceso a información que no pertenece a su departamento; las búsquedas se realizan con pocos criterios, presenta una interfaz de usuario que dista mucho de las

normas y los requerimientos visuales de una aplicación moderna (cuenta con una consola de fondo negro y letras verdes), que dificulta la interacción de los funcionarios con el software.

El proceso de Análisis de Información se encarga de la gestión de la información recabada durante las investigaciones pertinentes ante un hecho delictivo; la información está dada por los datos referentes a las armas, objetos, personas y vehículos que tienen alguna relación con el delito investigado, además controla todo lo relacionado con los funcionarios pertenecientes a la entidad. Actualmente el CICPC cuenta con la División de Información Policial, que manipula los datos utilizando el Sistema Integrado de Información Policial. Para que un funcionario ajeno a la División tenga acceso a la información que esta maneja tiene que solicitarla a través de una comunicación. La información no está organizada convenientemente y el volumen de datos que se maneja es insuficiente, no se comunica en tiempo real con los servicios externos proporcionados por el Instituto Nacional de Tráfico y Transporte Terrestre (INTTT) y el Servicio Autónomo de Identificación, Migración y Extranjería (SAIME) que le permita validar, con información actualizada, los datos ingresados a la aplicación. Por lo antes expuesto se plantea el siguiente problema.

Problema Científico: ¿Cómo garantizar una eficiente gestión y visualización de la información que se contempla en los Sub-Módulos Persona y Vehículo del Subsistema de Análisis de Información desde el punto de vista de la interfaz de usuario?

Objeto de Estudio: El Análisis de Información en los procesos de Gestión Policial.

Campo de Acción: La lógica de presentación para la gestión de personas y vehículos en el Sistema de Investigación e Información Policial.

Objetivo General de la Investigación: Analizar, diseñar e implementar los Sub-Módulos Vehículo y Persona del Subsistema de Análisis de Información del Sistema de Investigación e Información Policial, desde el punto de vista de lógica de presentación de acuerdo con los requerimientos funcionales y no funcionales de la aplicación.

Objetivos Específicos de la Investigación:

- Elaborar el marco teórico conceptual.
- Analizar la capa de presentación de los Sub-Módulos Vehículo y Persona.

- Diseñar la capa de presentación de los Sub-Módulos Vehículo y Persona.
- Implementar la capa de presentación de los Sub-Módulos Vehículo y Persona.

Ideas a Defender: Si se realiza una implementación robusta de los Sub-Módulos Persona y Vehículo del Subsistema de Análisis de Información desde el punto de vista de la interfaz de usuario entonces se garantizará una eficiente gestión y visualización de la información en dichos módulos.

Tareas de la Investigación:

- Realizar un detallado análisis de las descripciones de los Casos de Uso de los Sub-Módulos Persona y Vehículo del Subsistema de Análisis de Información del Sistema de Investigación e Información Policial.
- Realizar un estudio profundo de las tecnologías y herramientas actuales para la producción de software.
- Confeccionar los Diagramas de Clases del análisis de la capa de lógica de presentación y los diagramas de comunicación entre estas clases.
- Confeccionar los Diagramas de Clases del diseño de la capa de presentación y los diagramas de Contrato entre Paquetes que se adapten a los requerimientos funcionales y no funcionales de los Sub-Módulos Vehículo y Persona.

El documento fue elaborado con la siguiente estructura:

- Capítulo 1. Fundamentación Teórica.
- Capítulo 2. Análisis, Diseño e Implementación de la Propuesta de Solución.
- Capítulo 3. Validación de la Solución Propuesta.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. INTRODUCCIÓN

Este capítulo contiene la base teórica para la elaboración del software propuesto. Describe los Sistemas de Gestión Policial y estudia sistemas similares dentro del campo de acción. Argumenta el uso de las metodologías, herramientas y tecnologías a utilizar en el proceso de desarrollo del Sistema de Investigación e Información Policial.

1.2. SOFTWARE DE GESTIÓN POLICIAL

Cada día son más los documentos a procesar en las entidades policiales debido a que los índices de criminalidad van en aumento. Se convierte en un proceso muy engorroso manejar manualmente el grueso de información que se genera relacionada con algún delito. Un Sistema de Gestión Policial es un software creado para automatizar todos los procesos posibles que se desarrollan en una institución policial, en su mayoría, estos procesos se encuentran relacionados con la información recabada en las escenas de crímenes, denuncias, experticias y del personal perteneciente a la entidad, además de los procesos de búsquedas y reportes que tienen una gran utilidad. El manejo de la información es muy importante en estos sistemas, pues es el bien más preciado de estas instituciones. También se le brinda mucha atención a la seguridad, debido a que las entidades policiales la requieren, por lo tanto durante el desarrollo de estos sistemas se le presta mucha atención a la confianza de las contraseñas y al manejo de perfiles de acceso.

La creación del SIIPOL ayudará a la lucha que actualmente desarrolla el gobierno de la República Bolivariana de Venezuela contra el crimen. Dada la importancia que tiene la creación de un software robusto, rápido y seguro, surge la necesidad de investigar la existencia de otros Sistemas de Gestión Policial en el mundo o en el marco nacional para acumular conocimientos sobre estos, de esta forma desarrollar una aplicación robusta y verificar la necesidad de la creación del Sistema de Investigación e Información Policial.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

1.2.1 SISTEMA INTEGRADO DE INFORMACIÓN POLICIAL

El Sistema Integrado de Información Policial es el software con que actualmente cuenta el Cuerpo de Investigaciones Científicas Penales y Criminalísticas de la República Bolivariana de Venezuela para dar soporte informático a los procesos que se llevan a cabo en esta entidad.

Este fue desarrollado con Adabas-Natural un lenguaje de programación que actualmente es obsoleto y servidores de base datos SUN 6500 el cual ya está retirado del mercado. El sistema es insuficiente pues no abarca los procesos necesarios incidiendo negativamente en el buen desempeño de los funcionarios que lo manejan y por tanto también influye en el correcto desarrollo de los procesos policiales llevados a cabo por el CICPC.

1.2.2 DATAPOLICE

El software de Gestión Policial Datapolice se distingue por su versatilidad a la hora de aprovechar al máximo las posibilidades de comunicaciones actuales. Con la instalación del mismo la plantilla de la Policía contará desde el primer momento con plena comunicación en tiempo real con los datos obrantes en sus bases de datos. Ningún otro programa existente utiliza la plataforma empleada por Datapolice. Puede ser utilizado a través de ordenadores portátiles, PDAs e incluso teléfonos móviles. Además de contar con las ventajas anteriores, el programa Datapolice, cuenta con la posibilidad de compartir los datos con el resto de las Estaciones de Policía locales. El programa funciona como una página web, por lo tanto será extremadamente fácil de utilizar por cualquier usuario sin experiencia en el manejo del mismo, lo que redundará en un importante ahorro en formación de los agentes. Datapolice ofrece una plataforma rápida y versátil y plenamente probada en otros Cuerpos Policiales. El programa es plenamente configurable al cliente y se adapta a cualquier norma de estandarización de formatos de documentos existente.⁽²⁾

Este es un software creado con fines comerciales y está orientado a un proceso general, tiene como inconvenientes que no se ajusta a los procesos específicos de cada entidad.

1.2.3 SISTEMA TERRITORIAL DE EMERGENCIAS Y GESTIÓN POLICIAL (STEGPOL)

El STEGPOL es un Sistema de Información Geográfica con tecnología de punta sobre una Plataforma Nacional Común de Información aplicada al Sistema de Emergencias Nacionales y al Sistema Territorial de Gestión Policial que operan previamente en Chile. La idea principal del Proyecto STEGPOL es integrar

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

a las diferentes entidades como Carabineros, Investigaciones, Ministerio del Interior y Municipios, entre otros, en una Plataforma Nacional Común de Información que permita el intercambio de datos y que sirva de apoyo a la gestión operacional regional o comunal, donde dichas Instituciones estén interconectadas entre sí, en especial con diferentes Divisiones o Departamentos de Carabineros, tales como Jefaturas de Zona, Prefecturas, Comisarías, Sub-Comisarías, Tenencias y por último los Retenes, más el apoyo directo y coordinado entre Carabineros y el área de Seguridad de los Municipios. Hoy día se han dado todas las condiciones para poder interconectar un gran número de Comisarías, y que todas tengan acceso a una misma Plataforma de Mapas Digitalizados del país, además de las otras Instituciones relacionadas al tema de seguridad y sus respectivas unidades especializadas. A través de la Intranet del Estado o vía comunicación en Banda Ancha – Internet, se puede acceder al STEGPOL montado en la web con diferentes claves de acceso restringido tanto para los usuarios operadores encargados de ingresar o modificar datos en línea, como para aquellos que solo tengan acceso a consultar. ⁽³⁾

Al realizar un análisis de los sistemas que facilitan la gestión policial en los ámbitos nacional e internacional se evidencia la necesidad de la creación del Sistema de Investigación e Información Policial, ya que estos software manejan una lógica general relacionadas con los procesos policiales, pero el CICPC necesita de una herramienta que se adecue a los procesos propios que este órgano realiza. Durante el análisis de los procesos del subsistema de Análisis de Información se han encontrado muchos de estos procesos de naturaleza única, además, los datos a manejar así como su organización avalan la creación de este sistema.

1.3. METODOLOGÍA, LENGUAJES Y HERRAMIENTAS DE DESARROLLO

El proceso de desarrollo del software tiene como principal objetivo la producción eficaz y eficiente de un producto de software. Los proyectos de desarrollo de software tienen muchas semejanzas a cualquier otro proyecto de ingeniería, pero en su construcción hay varios desafíos adicionales relacionados con la naturaleza del producto a obtener. Con el auge de la industria del software se ha creado la necesidad de mejorar las metodologías y las herramientas con las que se desarrolla. Cada vez son más extensos y complicados los procesos a automatizar y el tiempo que se le da al desarrollador es prácticamente el mismo.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Una metodología de desarrollo es un proceso detallado y completo, donde se precisan los roles, artefactos, actividades e involucrados, también se definen prácticas y técnicas adecuadas. Existen varios tipos de metodologías:

- Estructuradas: Dirigidas esencialmente a proyectos que desarrollan con lenguajes de 3ra o 4ta generación.
- Orientadas a Objetos: Son utilizadas para desarrollar con lenguajes Orientados a Objetos. El Proceso Unificado de Desarrollo de Software (Rational Unified Process - RUP) es una de estas metodologías.
- Tradicionales (no ágiles): Guiadas por una fuerte planificación, en estas la etapa de Análisis y Diseño se realiza con gran intensidad.
- Ágiles: Se caracterizan por ser incremental, cooperativas y adaptables.

Se debe ofrecer mucha atención a la hora de escoger la metodología a utilizar, ya que esta será la guía durante todo el proceso de desarrollo. Se tendrá en cuenta para escoger la metodología el tamaño del software y el tiempo de elaboración del que se dispone.

Las herramientas de desarrollo son todos aquellos instrumentos que usa el desarrollador para crear un software. Existen muchas y pueden ser: herramientas de modelado, editores, IDEs (Entornos de Desarrollo Integrados), herramientas para la gestión de la configuración, herramientas de análisis de código y otras. Estos instrumentos deben ser seleccionados tomando en cuenta muchos factores, entre los que se encuentran: las tecnologías con que cuenta el desarrollador, el nivel de integración y el dominio que tenga el equipo de desarrollo sobre el manejo de estas herramientas.

En este caso, las metodologías y herramientas fueron seleccionadas por el cliente, el equipo de Arquitectura y la dirección del Proyecto CICPC. El cliente seleccionó el lenguaje Java y el gestor de bases de datos Oracle. A continuación se muestra una descripción con las principales características y ventajas que proporcionan cada una de estas herramientas y metodologías por separado. Todas ellas han sido seleccionadas por las facilidades de integración que brindan, costos y conocimientos acumulados por otros proyectos de la Universidad de las Ciencias Informáticas.

1.3.1 EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

El Proceso Unificado de Desarrollo de Software junto al Lenguaje Unificado de Modelado (UML), forman la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

orientados a objetos. Constituye una guía rectora que decide quién hace qué, cuándo y cómo lo hace. RUP dirige las actividades que se realizan por rol, les da un orden, especifica qué artefactos deberían ser desarrollados y puede además monitorear y medir los productos y actividades de un proyecto, teniendo en cuenta la calidad del producto final. RUP se caracteriza por dividir el ciclo de vida de la producción del software en 4 fases: ⁽⁴⁾

- Inicio o Conceptualización: Donde se determina la visión del proyecto, o sea se comprende el entorno y se determina el alcance del producto.
- Elaboración: En esta etapa se determinan los cimientos de la arquitectura y se analiza el dominio del problema.
- Construcción: En esta fase se obtiene la capacidad operacional inicial del producto.
- Transición: Se obtiene el release o liberación del producto y se pone en manos de los usuarios finales.

Los elementos característicos del RUP son: ⁽⁴⁾

- Actividades: Son los procesos que se llegan a realizar en cada iteración.
- Trabajadores: Son las personas o entidades involucradas en cada proceso.
- Artefactos: Un artefacto puede ser un modelo o un elemento de modelo, un documento, en fin todo lo que puede ser generado en el proceso.

Sus características principales son: ⁽⁴⁾

- Dirigido por Casos de Uso: Los Casos de Uso guían el proceso de desarrollo pues los modelos que se obtienen representan la realización de los mismos.
- Centrado en arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo.
- Iterativo e Incremental: Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. En el caso de una iteración de la fase elaboración, se centra la atención en el Análisis y Diseño, a la vez que se refinan los requerimientos y se obtiene un producto con un determinado nivel, que irá creciendo incrementalmente en cada iteración.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Cuenta con 9 flujos de trabajo, 6 principales y 3 de soporte los cuales son: ⁽⁴⁾

- Modelamiento del Negocio.
- Captura de Requerimientos del Sistema.
- Análisis y Diseño.
- Implementación.
- Prueba.
- Despliegue.
- Gestión de Configuración y Cambios.
- Gestión de Proyectos.
- Entorno.

Donde los 3 últimos constituyen los flujos de soporte.

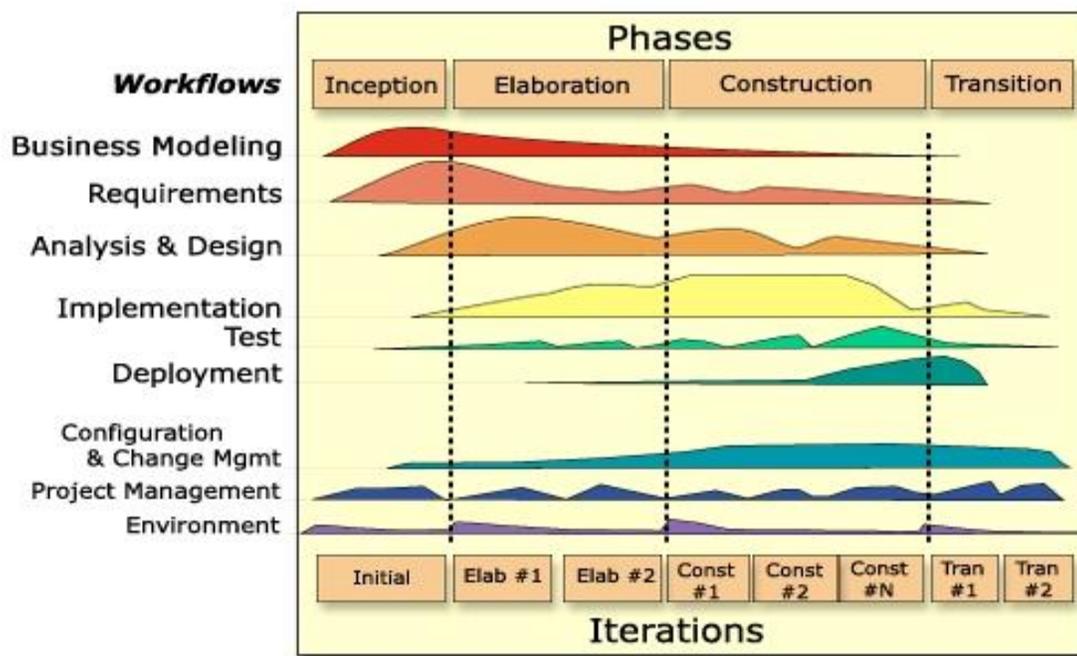


Fig. 1.1 Fases y Flujos de Trabajo del RUP.

Esta metodología es usada en proyectos de gran envergadura que impliquen elevados tiempos de desarrollo y que posean un equipo de desarrollo grande.

1.3.2 LENGUAJE UNIFICADO DE MODELADO (UML)

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es actualmente el lenguaje de modelado más conocido. UML ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Su objetivo principal es el de la comunicación entre quien tiene la idea y el desarrollador. Es importante resaltar que UML especifica, no describe métodos ni procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. RUP lo utiliza como lenguaje de modelado, pero puede ser usado además por otras metodologías. El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. En UML 2.0, la última versión disponible, hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente: ⁽⁵⁾

- Diagramas de Estructura: Enfatizan en los elementos que deben existir en el sistema modelado; incluye, entre otros, Diagrama de clases, Diagrama de Componentes, Diagrama de Objetos y Diagrama de Paquetes.
- Diagramas de Comportamiento: Enfatizan en lo que debe suceder en el sistema modelado; ejemplo son el Diagrama de Actividades y Diagrama de Casos de Uso.
- Diagramas de Interacción: Son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado; aquí se puede ver el Diagrama de Secuencia y Diagrama de Colaboración.

El uso de UML y el trabajo continuo que se ha venido realizando sobre él lo convierten en una práctica que asegura la especificación de los procesos en el desarrollo del software, y es muy favorable su uso como vía de comunicación y documentación.

1.3.3 VISUAL PARADIGM PARA UML

Visual Paradigm es una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) que soporta, completamente, el ciclo de vida del desarrollo de software: Análisis y Diseño Orientados a Objetos, Construcción, Pruebas y Despliegue. Visual Paradigm para UML

soporta un conjunto de lenguajes, tanto para la generación de código como para la realización de ingeniería inversa en: Java, C++, CORBA IDL, PHP, XML Schema, Ada y Python. Además, apoya la generación de código C#, VB.NET, Object Definition Language (ODL), ActionScript, Delphi, Perl. La ingeniería inversa en ocasiones también soporta clases Java, .NET dll y .exe, JDBC, y los archivos de mapeo Hibernate. Visual Paradigm posibilita además generar código Java a partir de modelos y también generar modelos a partir de código Java, cualquier cambio que se realice en el código puede verse reflejado en el modelo y viceversa. Permite exportar e importar archivos de Rational Rose (.MDL / .CAT). Aprovecha al máximo la interoperabilidad de productos ya que permite la importación y exportación de modelos de proyectos de/a XML abierto. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos. ⁽⁶⁾

1.3.4 JAVA PARA SOLUCIONES EMPRESARIALES (JAVA 2 ENTERPRISE EDITION O J2EE)

Los orígenes del lenguaje de programación Java se remontan desde el año 1991 con la creación de un proyecto para el control de consolas de televisión. James Gosling, el padre de Java, al no querer utilizar C++ creó el lenguaje Java, que en sus principios era llamado OAK. Los objetivos al diseñar Java pretendían *crear un lenguaje sencillo, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutral, portátil, de gran rendimiento, multitarea y dinámico*. ⁽⁷⁾ Es una plataforma de programación (parte de la Plataforma Java) que define un estándar para desarrollar aplicaciones empresariales con arquitectura multicapas, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. La plataforma J2EE aprovecha muchas de las características de la Plataforma Java 2 Standard Edition (J2SE), como Write Once (Escribir una vez), Run Anywhere (se ejecuta en cualquier lugar), la portabilidad, el API de JDBC para acceso a bases de datos, tecnología CORBA para la interacción con los recursos empresariales existentes y un modelo de seguridad que protege los datos incluso en aplicaciones que se ejecutan en servidores conectados a Internet. Partiendo de esta base, la J2EE agrega pleno apoyo a los componentes Enterprise JavaBeans, Servlet, Portlets, JavaServer Pages y varias tecnologías de servicios web. Teniendo esto, es posible para el desarrollador crear una Aplicación Empresarial portable entre plataformas, escalable e integrable con tecnologías anteriores. Esta plataforma brinda otros servicios, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, permitiéndole a los desarrolladores concentrarse más en la lógica de negocio de los

componentes en lugar de tareas de mantenimiento de bajo nivel. El estándar J2EE incluye las especificaciones completas y pruebas para garantizar la portabilidad de aplicaciones a través de la amplia gama de sistemas empresariales existentes capaces de apoyar a la plataforma. Además, la especificación asegura la Interoperabilidad de Servicios Web (WS-I).

1.3.5 JAVA SERVER FACES (JSF)

Un framework define una estructura que puede servir para organizar y desarrollar un proyecto de software. Suelen incluir soporte de programas, bibliotecas y un lenguaje interpretado que ayudan a desarrollar e incluir los demás componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. JavaServer Faces (JSF) es un framework para aplicaciones Java. Esta tecnología ha sido diseñada para ser flexible, aprovecha las UI (Interfaz de Usuario) estándares y los conceptos web existentes sin limitar al desarrollador a un lenguaje de marcado, protocolo o dispositivo cliente. JSF presenta una arquitectura basada en componentes, cuenta con una serie de componentes UI o widgets (botones, cajas de chequeo, campos de texto), todos ellos reutilizables que agilizan el proceso de desarrollo. Dichos componentes son orientados a eventos (cambio del valor de un campo, clickeando un botón) que son enviados al servidor para ser procesados. JSF tiene un conjunto de términos que forman la base conceptual de las características que proporciona: ⁽⁸⁾

- Componentes UI: Objetos que se encuentran en el lado del servidor que proveen una funcionalidad específica para que el cliente interactúe con el software. Estos son JavaBeans con propiedades, métodos y eventos organizados en un árbol de componentes que se muestran como una página.
- Renderer: Funciona como un traductor, genera el código HTML para que los componentes UI de JSF sean reconocidos por el navegador web cliente, de esta forma el usuario puede interactuar con la aplicación.
- Validator (Validador): Se encarga de comprobar que la información suministrada por el usuario sea correcta. Un solo componente puede tener varias validaciones.
- Backing beans (Beans de Respaldo): JavaBeans especializados que se encargan de recoger la información de los componentes UI e implementa los métodos que escuchan los eventos lanzados por estos elementos.

- Converter (Convertidor): Convierte el valor de un componente a una cadena de caracteres y viceversa, para que este sea mostrado. Al componente solo se le puede asignar un convertidor.
- Events and listeners (Eventos y oyentes): JSF usa el modelo event/listener. Los componentes UI generan eventos y los listeners los capturan para procesarlos.
- Messages (Mensajes): Es la información enviada como respuesta al usuario por alguna acción. Generalmente está relacionada con las validaciones o errores en la implementación que impidan el correcto funcionamiento de la aplicación.
- Navigation (Navegación): Característica que le permite moverse de una página a otra. JSF tiene un poderoso sistema de navegación integrado con event listeners especializados.

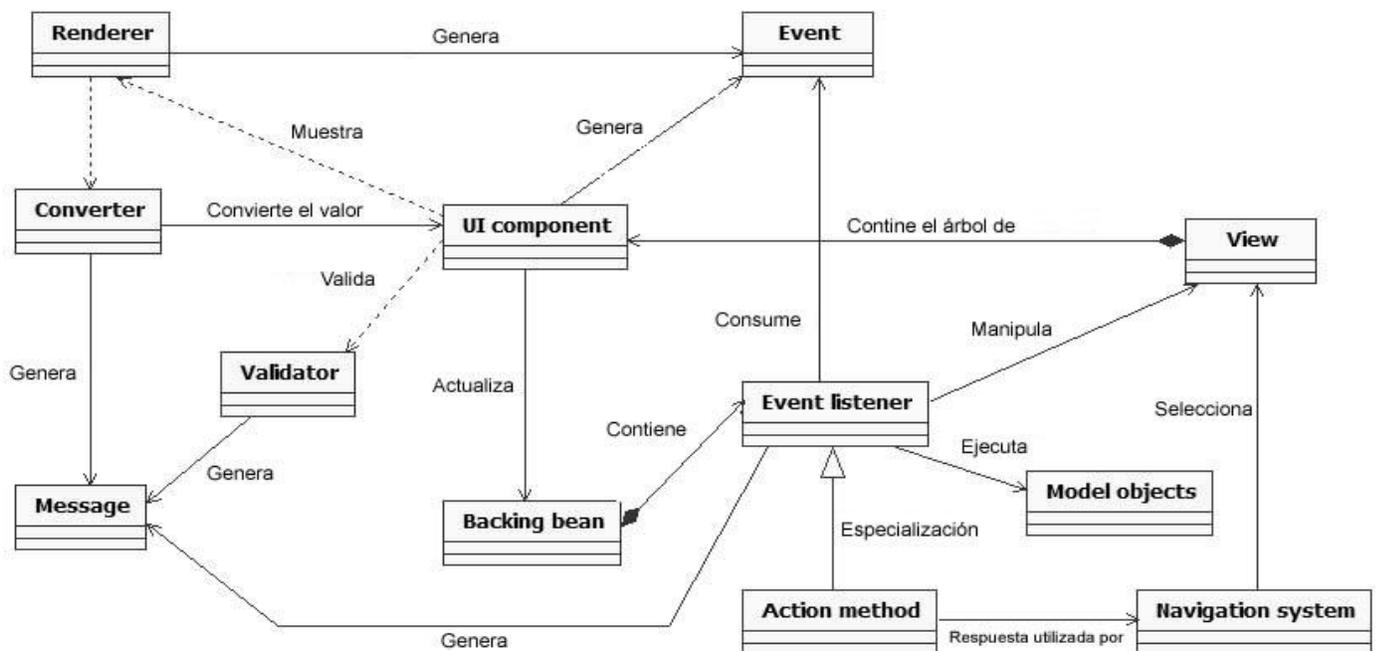


Fig. 1.2 Modelo de Relación entre los términos de JSF.

JSF tiene numerosas implementaciones de estas características que son de uso general, también permite desarrollar cualquiera con especificaciones propias del desarrollador tales como: validadores, beans de respaldo, convertidores y componentes de presentación.

Una de las grandes ventajas de que JSF sea una especificación es que no limita al desarrollador a un solo fabricante, este puede escoger la implementación que sea más ventajosa para su proyecto teniendo en cuenta: número de componentes que proporciona, rendimiento, soporte, precio y política. MyFaces es una

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

de estas implementaciones elaborada por Apache MyFaces Project perteneciente a Apache Software Foundation. Esta implementación brinda:

- Implementación de JSF (MyFaces API, Módulos de implementación de MyFaces).
- Varias librerías de componentes que contienen UI widgets para la construcción de aplicaciones web con JSF, una de ellas es MyFaces Tomahawk.
- Módulos de integración con otras tecnologías y estándares.

RichFaces es una librería con numerosos componentes para JavaServer Faces y a la vez un framework open source avanzado que integra las capacidades que brinda AJAX (JavaScript asíncrono y XML). Además, utiliza el framework JSF incluyendo su ciclo de vida, validaciones y convertidores. Todas estas capacidades son fácilmente integrables con MyFaces.

1.3.6 JAVASCRIPT

JavaScript es el lenguaje de programación Web del lado del cliente más extendido. Con este lenguaje script se pueden generar páginas dinámicas en función de las preferencias del usuario, validar los datos introducidos en un formulario o modificar el contenido de la página. Fue creado por Netscape para su navegador 2.0 con la versión 1.0 de JavaScript y ya marcha por la 1.5; está basado en objetos. Por cuestiones de seguridad se ha impedido que JavaScript pueda leer y escribir a disco, ejecutar otros programas y conectarse a otra computadora o enviar correo, producto que pudieran introducirse virus de esta forma cuando un servidor envía una página a un cliente. De la misma forma en el navegador se puede configurar para no ejecutar código script.

JavaScript está basado en objetos, no está orientado a objetos, pues no permite la herencia. Sus principales características son:

- Manejado por eventos: JavaScript puede responder a eventos como el movimiento del mouse y la carga de una página web.
- Independiente de cualquier plataforma: Los programas de JavaScript están diseñados para ejecutarse dentro de documentos HTML. Son independientes de cualquier plataforma o sistema operativo.
- Permite desarrollo rápido: El navegador web y el código HTML manejan la mayoría de las características como formas, cuadros, y otros elementos de Interfaz Gráfica del Usuario (GUI).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Esto hace que los programadores de JavaScript no tienen que preocuparse de crear o manejar estos elementos en sus aplicaciones.

- Fácil de aprender: No incluye complejas reglas sintácticas.

1.3.7 ECLIPSE

La construcción de aplicaciones web es un proceso muy largo y difícil, se necesita estar al tanto de muchos detalles. Partiendo que una aplicación web incorpora una inmensa diversidad de recursos que están altamente integrados incluyendo: lenguajes de marcado, imágenes, script, Java Applets, controles ActiveX, servlets, entre otros, lo que dificulta en algunos casos validar la integración de estos. Esta variedad de recursos vincula una gran diversidad de especialistas quienes necesitan herramientas específicas por su rol. Se hace necesario durante el proceso de desarrollo realizar pruebas al código implementado. Los desarrolladores necesitan centrarse en la producción y minimizar en lo posible desviar la atención por trabas propias del instrumento con el que se trabaja. En muchos casos es necesario realizar numerosas integraciones, este proceso ofrece dificultades ya que algunas herramientas tienen mecanismos de integración propios.

La plataforma Eclipse es un entorno de desarrollo integrado de código abierto que reduce el costo de integración ya que provee un gran número de servicios, API's y frameworks que lo hacen un escalable y efectivo instrumento integrado. El workbench de eclipse provee un punto de integración central que permite al desarrollador realizar proyectos usando una lista de herramientas heterogéneas que al mismo tiempo muestran una vista común de toda la aplicación en perspectivas personalizables. Facilita al máximo la gestión de proyectos colaborativos mediante el control de versiones CVS (Concurrent Versions System, Sistema Concurrente de Versiones), es posible también con subversión exportar e importar proyectos. Es permitido añadir nuevas funcionalidades al editor a través de nuevos plugins (módulos) para programar en otros lenguajes además de Java como C/C++, PHP, JavaScript, entre otros.

1.4. ARQUITECTURA TÉCNICA

El desarrollo de aplicaciones web ha alcanzado gran auge en los últimos años debido a las facilidades que brindan los navegadores web como clientes ligeros ayudando a disminuir el costo de las estaciones de trabajo. Los sistemas son cada vez mayores y muy complejos, requiriendo mayor seguridad, mantenibilidad y gran velocidad de respuesta. Esencialmente estas aplicaciones se basan en un servidor

de aplicaciones que escucha las peticiones de uno o varios clientes a través de una red interna o internet basándose principalmente en un protocolo TCP/IP o HTTP.

Para el desarrollo del SIIPOL se propone la estructura que se muestra en el siguiente Diagrama de Despliegue.

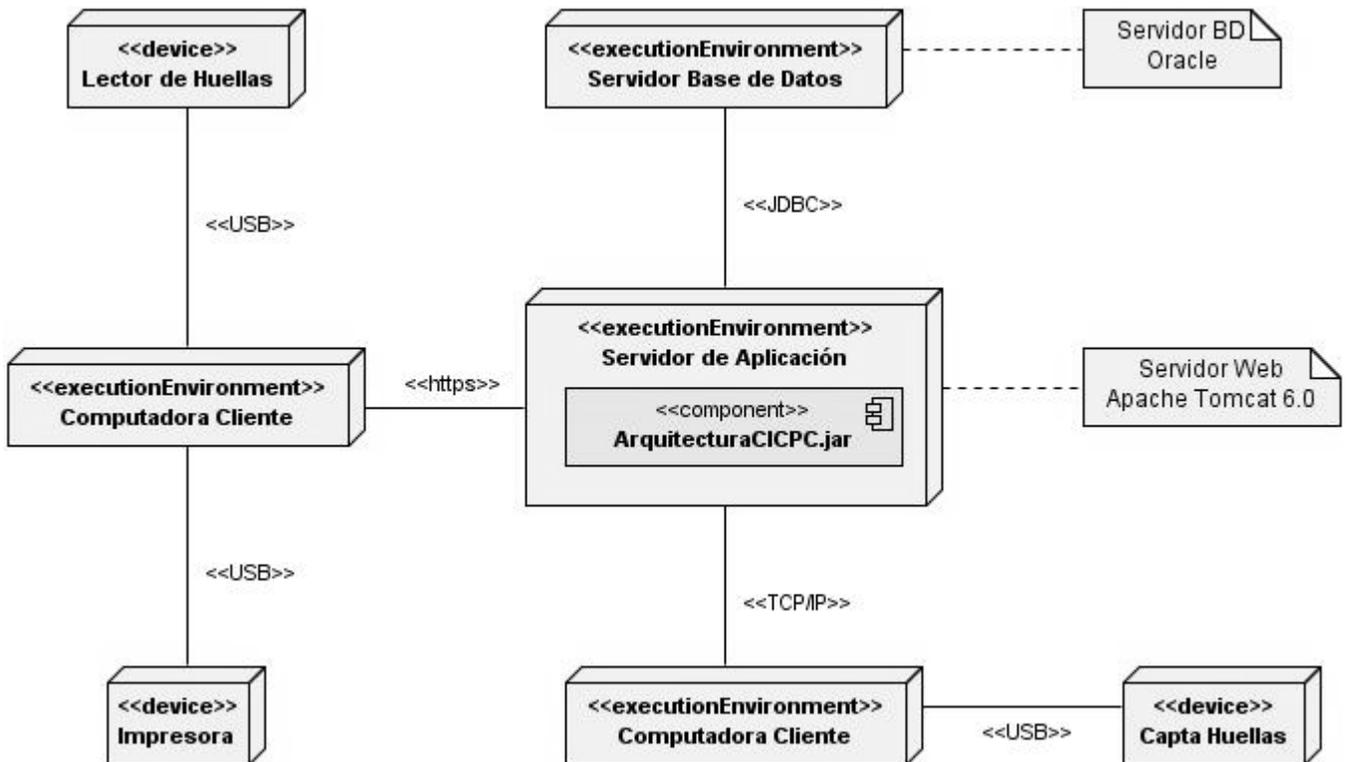


Fig. 1.3 Diagrama de Despliegue.

1.5. CONCLUSIONES

Las herramientas y metodologías detalladas anteriormente están a la delantera en cuestiones de desarrollo de software a nivel mundial. Brindan la integración necesaria para facilitar el proceso de desarrollo del SIIPOL. Se utilizará como metodología RUP, que controlará todo el proceso de producción del software y como lenguaje de modelado UML. Visual Paradigm para UML la herramienta de modelado por la buena integración con el Eclipse para la generación de código a partir de diagramas y viceversa. Java el lenguaje de programación pues fue a petición del cliente. El Eclipse será el IDE para desarrollar,

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

dada la variedad de herramientas que brinda y la posibilidad de personalizar las vistas en que estas se muestran. JSF el framework de Java para la implementación de la Capa de Lógica de Presentación por las facilidades que brinda al ser orientado a componentes y la reusabilidad de los mismos, en conjunto a la librería de componentes para JSF RichFaces; JavaScript el lenguaje del lado del cliente para brindar soporte a JSF en los casos que sea necesario. Utilizando todas estas herramientas integradas y como metodología RUP se obtendrá como resultado un Sistema de Información Policial que cumpla con los requerimientos del cliente.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN

2.1. INTRODUCCIÓN

Se tratarán en este capítulo los temas relativos al Análisis y Diseño de los Sub-Módulos Persona y Vehículo. Se mostrará el resultado obtenido a partir de la Ingeniería de Requerimientos y serán descritos brevemente los procesos que se llevan a cabo en cada uno de los flujos que se desarrollarán. Se mostrarán diagramas generados durante el proceso de Análisis y Diseño de los Casos de Uso más significativos de los Sub-Módulos Persona y Vehículo.

2.2. PREVIO AL ANÁLISIS Y DISEÑO

2.2.1 INTRODUCCIÓN AL SIIPOL

El SIIPOL está compuesto por varios módulos funcionales inter-relacionados, cada uno con una función y objetivos específicos. El sub-sistema de Análisis de Información pertenece a estos módulos y en él se gestiona toda la información referente a objetos, armas, personas y vehículos que es recabada en escenas de crímenes, investigaciones relacionadas a estos o indagaciones propias de la institución policial. También la información relacionada con los funcionarios de la entidad es manipulada por este subsistema.

Análisis de Información fue concebido principalmente para servir de base a los demás subsistemas. Cuenta con 7 sub-módulos: Búsqueda, Personas, Objetos, Vehículos, Armas, Armas Orgánicas, y Funcionarios. Exceptuando el manejo de los funcionarios y otras pocas funcionalidades como validación de algunos datos y consultas, los otros servicios que brinda este subsistema son subprocesos de otros subsistemas, los objetos, armas, personas y vehículos no representan por si solos una evidencia, únicamente tienen valor asociándose a algún proceso investigativo, es decir, formando parte de un Acta Procesal, Denuncia, Notificación o Solicitud llegan a ser de utilidad.

2.2.2 SUB-MÓDULO PERSONA

El Sub-Módulo Persona fue creado para mantener actualizados los datos obtenidos de uno o varios individuos durante un proceso investigativo, además de permitir adquirir toda la información posible

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

correspondiente a estos y sus relaciones con otros elementos tales como vehículos y armas. Tiene una gran cantidad de Casos de Uso (CU), en la imagen siguiente se muestra un modelo con los CU del sub-módulo.

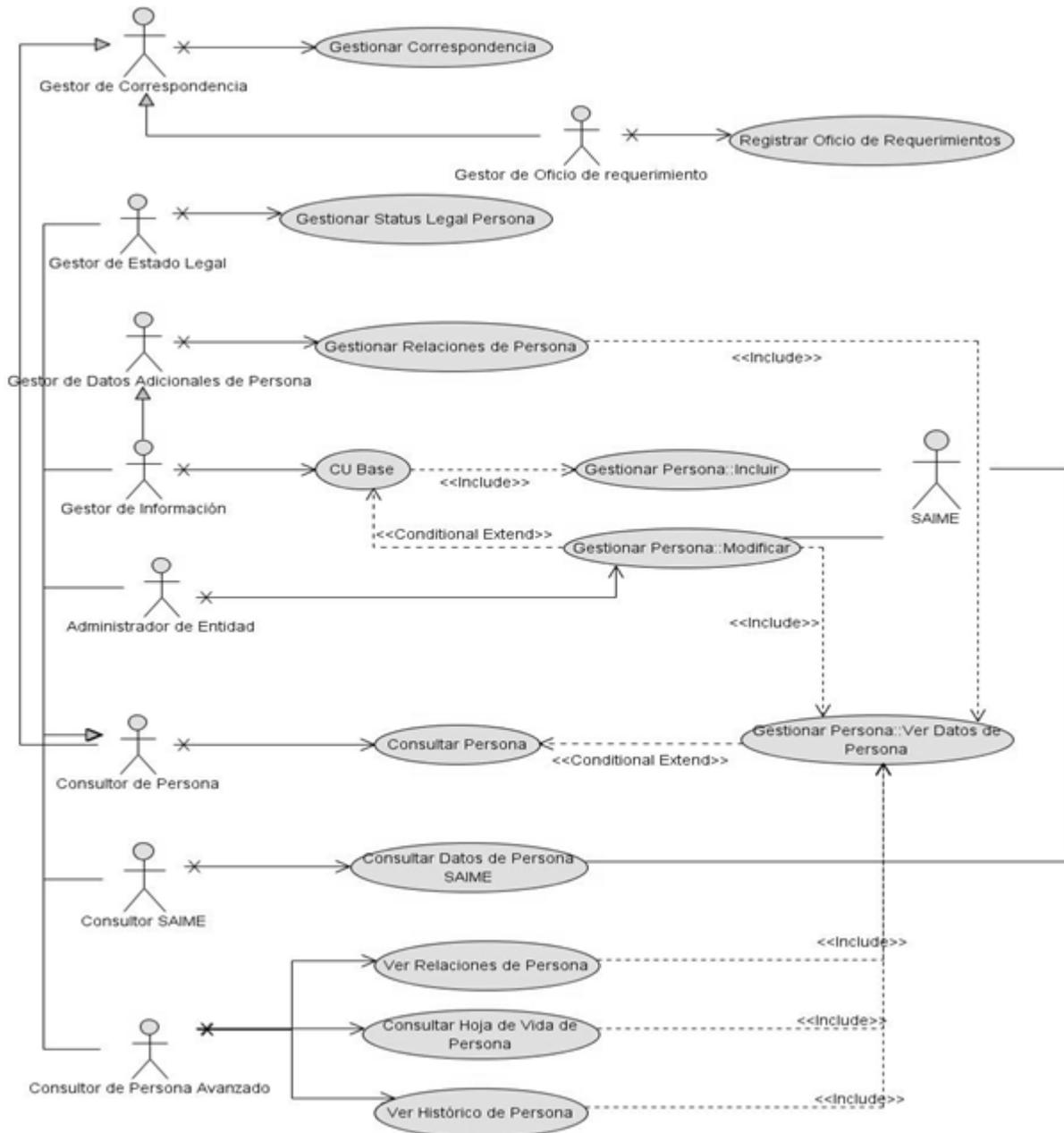


Fig. 2.1 Diagrama de CU del Sistema pertenecientes al sub-módulo Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

A continuación se presentan las descripciones de los Casos de Uso más significativos del Sub-Módulo Persona.

Tabla 1 Descripción del CU Gestionar Persona.

Nombre del CU	Gestionar Persona.
Objetivo	Crear, ver, o modificar una persona.
Actor (es)	Caso de Uso Base.
Descripción	El Caso de Uso inicia cuando el CU Base solicita realizar una operación sobre una persona. En caso de crear una persona el sistema brinda la posibilidad de introducir los datos conocidos de la persona, permitiendo validar con SAIME la información introducida. En caso de modificar los datos de la persona, el sistema muestra todos los datos conocidos de la misma y brinda la posibilidad de añadir nuevos datos o modificar los ya existentes, actualizando el registro en el sistema y llevando un registro histórico de los cambios ocurridos. En caso de ver los datos de la persona, el sistema muestra todos los datos conocidos de la misma.

Tabla 2 Descripción del CU Consultar Persona.

Nombre del CU	Consultar Persona.
Objetivo	Buscar y listar de manera ordenada un resumen de los datos de personas coincidentes con uno o varios criterios de búsqueda.
Actor (es)	Consultor de Persona.
Descripción	El Caso de Uso inicia cuando el Consultor de Persona accede a la opción de consultar una persona. El sistema muestra diferentes criterios para la búsqueda de la persona, el Consultor introduce los criterios deseados y el sistema devuelve las coincidencias encontradas. El Caso de Uso termina.

2.2.3 SUB-MÓDULO VEHÍCULO

En este sub-módulo se encuentran los Casos de Uso que permiten mantener los datos del vehículo actualizados y manejar toda la información correspondiente al vehículo así como su relación con otros elementos dentro del sistema. En la Figura 2.2 se muestra el Diagrama de los CU del Sistema.

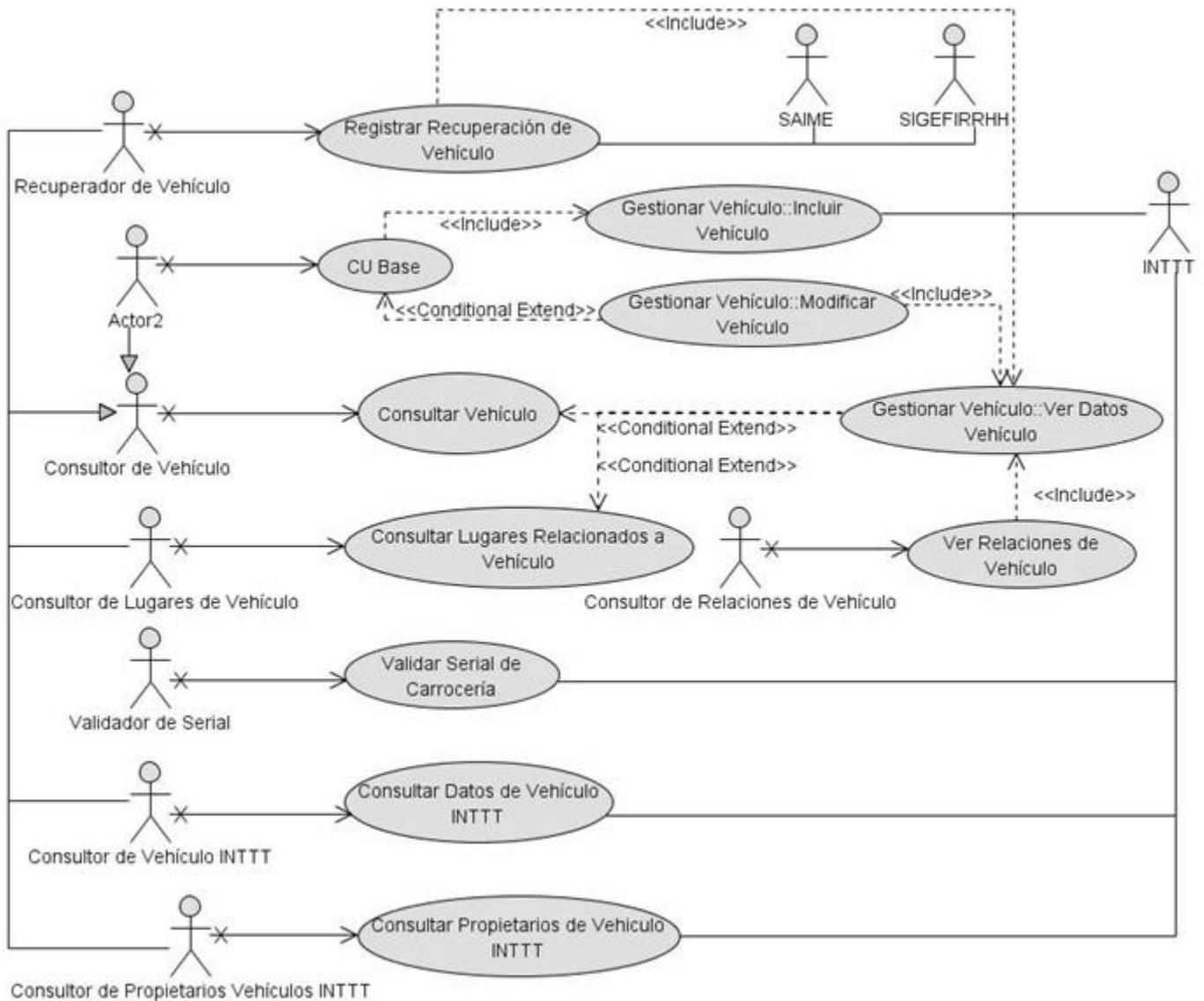


Fig. 2.2 Diagrama de CU significativos del Sistema pertenecientes al Sub-módulo Vehículo.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

A continuación se presentan las descripciones de los Casos de Uso más significativos del Sub-Módulo Vehículo.

Tabla 4 Descripción del CU Gestionar Vehículo.

Nombre del CU	Gestionar Vehículo.
Objetivo	Crear, ver, o modificar un vehículo.
Actor (es)	Caso de Uso Base.
Descripción	El Caso de Uso se inicia cuando el CU Base solicita realizar una operación sobre un Vehículo. En caso de crear un vehículo el sistema brinda la posibilidad de introducir los datos conocidos del vehículo, permitiendo validar con INTTT la información introducida. En caso de modificar los datos del vehículo, el sistema muestra todos los datos conocidos del mismo y brinda la posibilidad de añadir nuevos datos o modificar los ya existentes, actualizando el registro en el sistema y llevando un registro histórico de los cambios ocurridos. En caso de ver los datos del vehículo, el sistema muestra todos los datos conocidos del mismo.

Tabla 3 Descripción del CU Ver Relaciones de Vehículo.

Nombre del CU	Ver Relaciones de Vehículo.
Objetivo	Ver el carácter de las relaciones que tiene el vehículo con las Actas Procesales y Notificaciones con las que está relacionado.
Actor (es)	Consultor de Relaciones de Vehículo.
Descripción	El Caso de Uso inicia cuando el consultor indica ver las relaciones del vehículo con las Actas Procesales y Notificaciones con las que está relacionado. El sistema muestra las relaciones que tiene asociada el vehículo con las Actas Procesales y las Notificaciones registradas en el sistema.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

2.2.4 DISEÑO DE LAS INTERFACES DE USUARIO

La organización, diagramación y jerarquización de la información que se muestran en las interfaces de usuario del software, han sido definidas por los diseñadores de interfaz de usuario en correspondencia con lo pautado por el Arquitecto de Información. Para construir las interfaces de usuario (UI) es necesario leer previamente la documentación generada durante los procesos de diseño de las interfaces y las pautas de Arquitectura de la Información donde se encuentran detalladamente las medidas, proporciones y organización de los componentes visuales.



Fig. 2.3 Diseño de Interfaz de Usuario para el CU Gestionar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

La Figura 2.3 muestra un ejemplo de la organización de los componentes visuales en una de las ventanas perteneciente al CU Gestionar Persona.

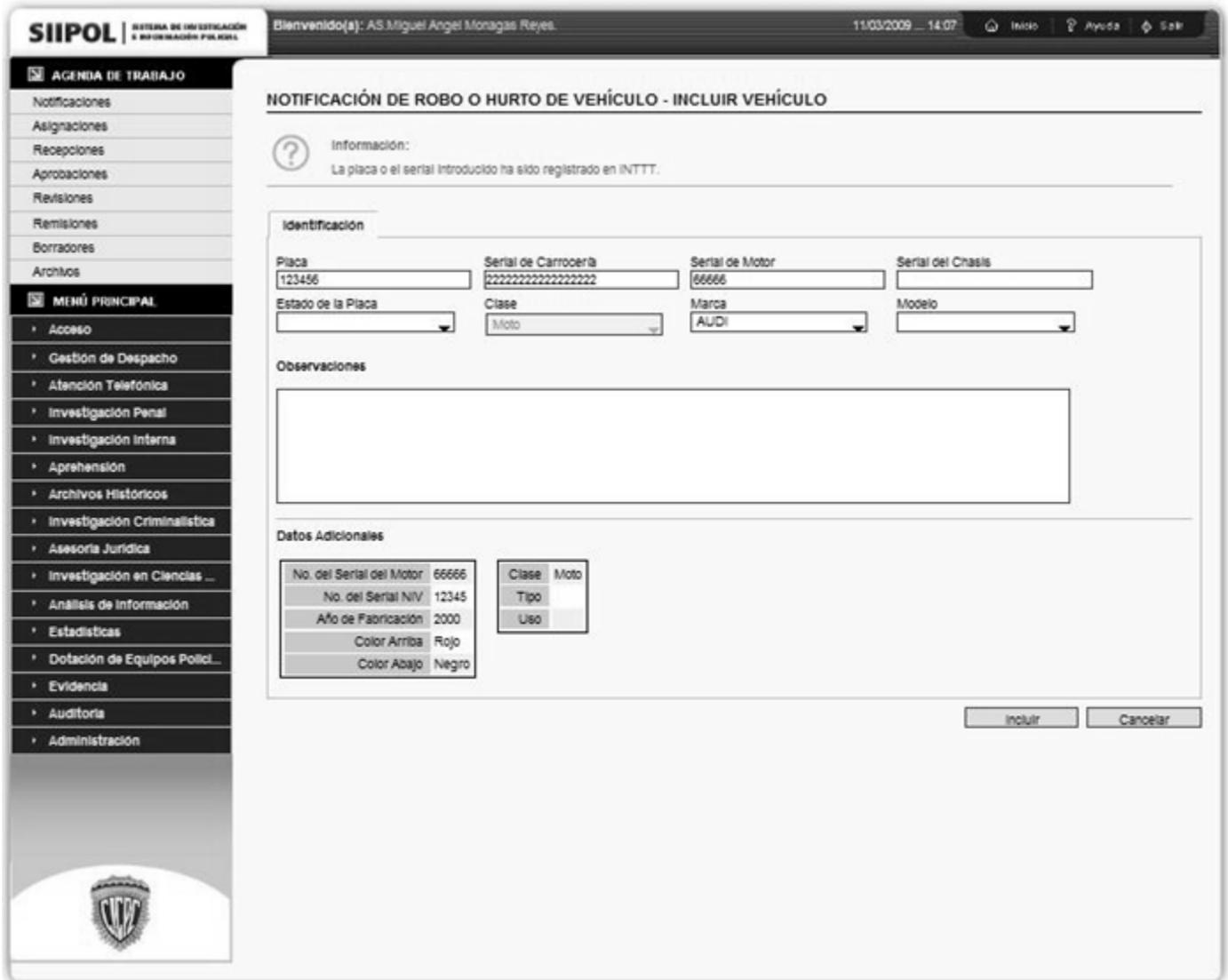


Fig. 2.4 Diseño de Interfaz de Usuario para el CU Gestionar Vehículo.

La Figura 2.4 muestra un ejemplo de la organización de los componentes visuales en una de las ventanas del CU Gestionar Vehículo.

2.3. MODELO DE ANÁLISIS

2.3.1 INTRODUCCIÓN AL FLUJO DE ANÁLISIS

Durante el Análisis, se estudian los requerimientos descritos en la captura de requerimientos. Aquí se refinan y estructuran con el objetivo de conseguir una comprensión más precisa de los requerimientos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero. A partir de este flujo se comienza a analizar el sistema con el lenguaje del desarrollador. El lenguaje que se utilizará en el análisis se basa en un modelo de objetos conceptual, llamado Modelo de Análisis.⁽⁴⁾

2.3.2 CLASES DEL ANÁLISIS

Las clases del Análisis se centran en los requerimientos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos formando los Diagramas de Clases del Análisis. RUP propone clasificar las clases en:⁽⁴⁾

- Clases Interfaz: Son aquellas que permiten al actor interactuar con el sistema, pueden ser pantallas, reportes o formularios.

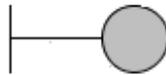


Fig. 2.5 Estereotipo de Clases Interfaz.

- Clases de Control: Las clases de control representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un Caso de Uso en concreto.



Fig. 2.6 Estereotipo de Clases de Control.

- Clases Entidad: Las clases entidad se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Estas clases modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.

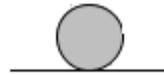


Fig. 2.7 Estereotipo de Clases Entidad.

2.3.3 DIAGRAMAS DE CLASES DEL ANÁLISIS

Para cada Caso de Uso se realizó un Diagrama de Clases del Análisis. A continuación se muestra el modelado de las relaciones entre las clases del Análisis pertenecientes a los CU más significativos dentro de los Sub-Módulos Persona y Vehículo.

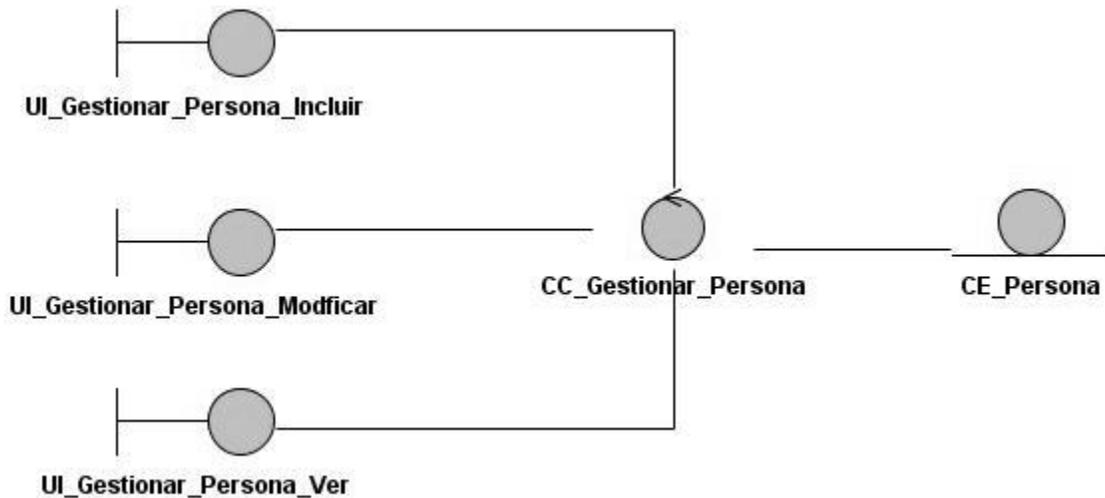


Fig. 2.8 Diagrama de Clases del Análisis del CU Gestionar Persona.

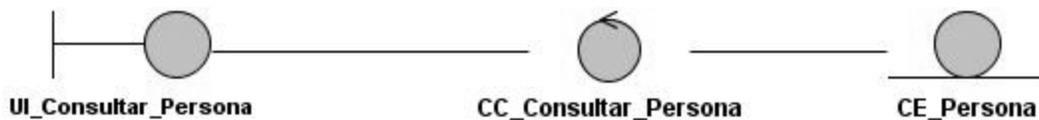


Fig. 2.9 Diagrama de Clases del Análisis del CU Consultar Persona.

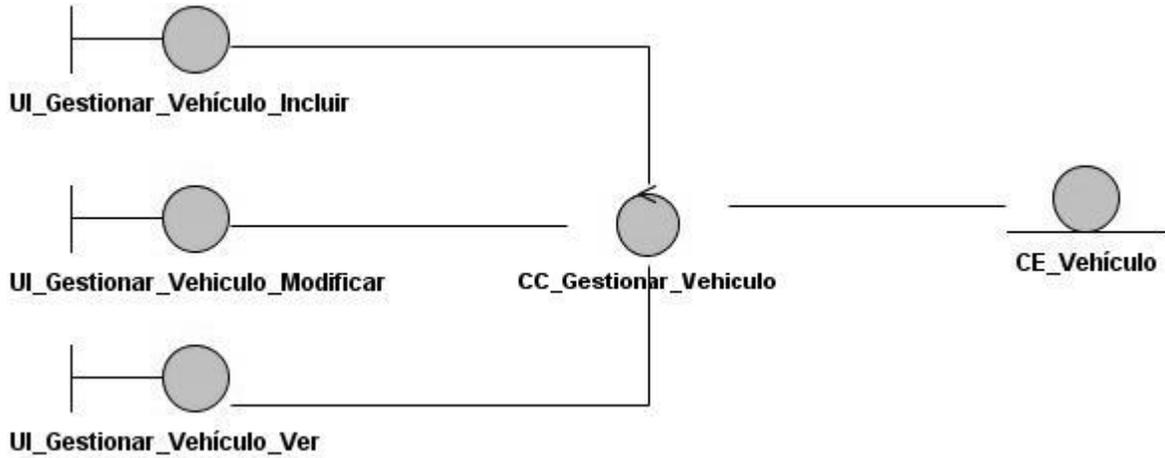


Fig. 2.10 Diagrama de Clases del Análisis del CU Gestionar Vehículo.



Fig. 2.11 Diagrama de Clases del Análisis del CU Ver Relaciones de Vehículo.

2.4. MODELO DE DISEÑO

2.4.1 INTRODUCCIÓN AL FLUJO DE DISEÑO

En el Diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requerimientos (funcionales y no funcionales). En este flujo se adquiere una comprensión en profundidad de los aspectos relacionados con los requerimientos no funcionales y restricciones relacionadas con los lenguajes de programación, componentes reutilizables y sistemas operativos. Se crea una entrada apropiada y un punto de partida para actividades de implementación. Se descomponen los trabajos de implementación en partes más manejables que puedan ser manejadas por diferentes equipos de desarrollo.⁽⁴⁾

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

2.4.2 DIAGRAMA DE PAQUETES

Los Sub-Módulos Persona y Vehículo interactúan con varios módulos del SIIPOL. Mantienen relaciones con las entidades y servicios de algunos de estos subsistemas y las funcionalidades que brindan el paquete *webservices* para la conexión con servicios externos y *core* para manejar trazas y seguridad. Los paquetes están organizados para aumentar su coherencia interna y minimizar el acoplamiento externo entre estos. En el Diagrama de Paquetes se muestran detalladamente todas estas relaciones. (Fig. 2.12).

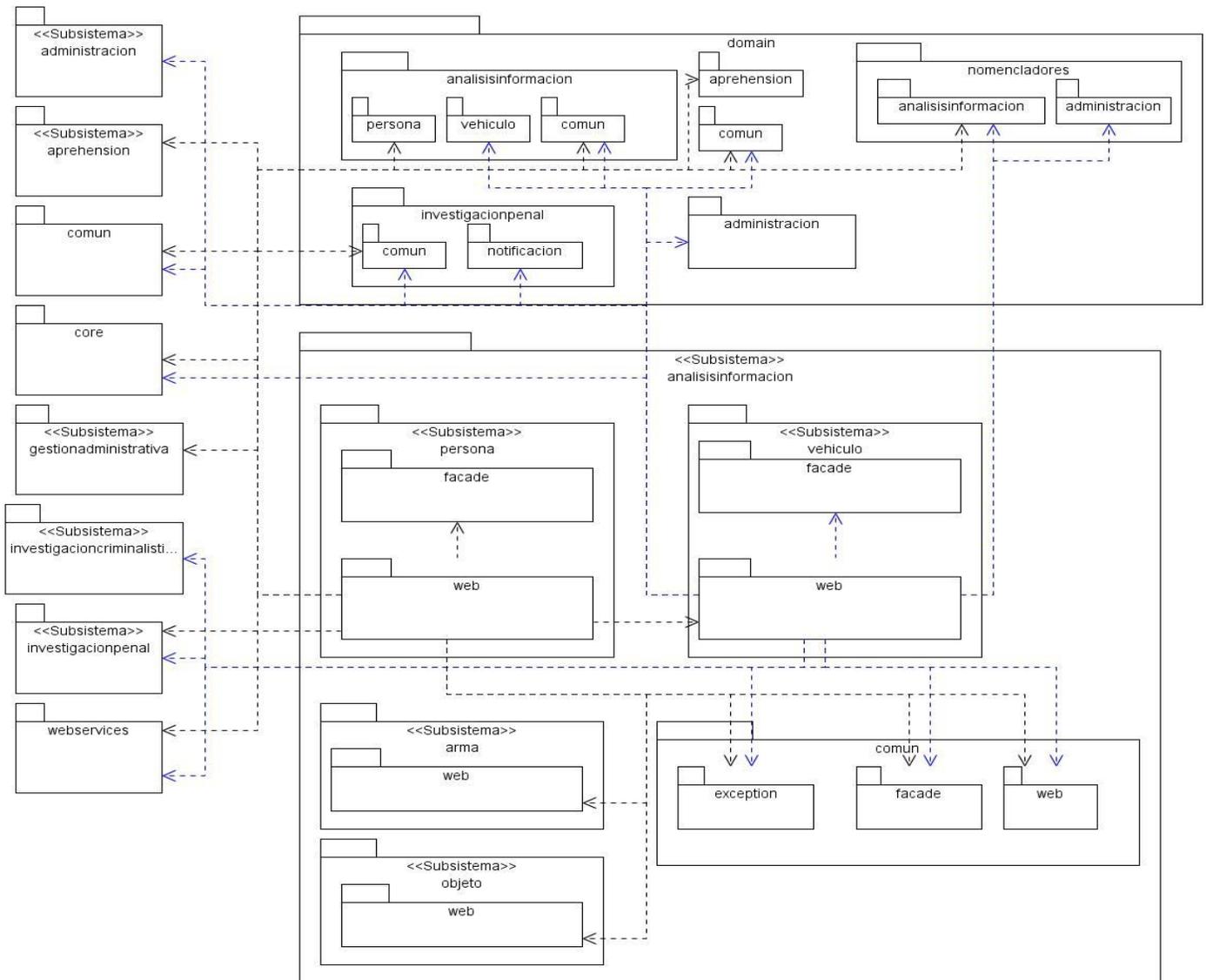


Fig. 2.12 Diagrama de Paquetes de los Sub-Módulos Persona y Vehículo.

2.4.3 DIAGRAMAS DE CLASES DEL DISEÑO

Una clase del Diseño es una abstracción de una clase o construcción similar en la implementación del sistema. El lenguaje utilizado para especificar estas clases es el mismo que el lenguaje de programación que se va a utilizar para el desarrollo del software, las relaciones de estas con otras clases a menudo tienen un significado directo cuando la clase es implementada. Los Diagramas de Clases del Diseño contienen las clases, sus objetos y también los subsistemas y paquetes que contienen las clases del Diseño.⁽⁴⁾

El uso de patrones es de vital importancia para el desarrollo de cualquier proyecto. Los patrones tienen dos características principales: una es que deben haber resuelto problemas similares en ocasiones anteriores y deben ser reusables, es decir, que pueden ser aplicables a diferentes problemas en distintas circunstancias. Durante el diseño se debe tomar en cuenta la arquitectura definida por el equipo de arquitectos del proyecto, usar los patrones seleccionados y las buenas prácticas que estos ofrecen.

Para el desarrollo de la aplicación fue definida una arquitectura **n-capas** (tres niveles para el SIIPOL) que brinda escalabilidad, modularidad, facilita el desarrollo y aumenta la mantenibilidad del sistema. Para algunas capas existen clases que influyen en el desarrollo de la aplicación, BaseBean es una de estas clases que forman parte de la capa de presentación, la cual agrupa funcionalidades afines con todas las clases que sirven de respaldo a cada una de las páginas de JSF, por lo que todos los beans de respaldo deberán heredar de dicha clase.

El **Modelo Vista Controlador (MVC)** es otro de los patrones a utilizar en el desarrollo del SIIPOL. Este patrón permite separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Es decir, se puede modificar cualquiera de estos elementos sin necesidad de realizar grandes cambios en los otros. JSF implementa el MVC, donde las páginas *.jsp* (*java server pages*) son la vista, los beans de respaldo el controlador y las entidades del dominio el modelo.

El patrón **facade** suministra una interfaz unificada y sencilla que funciona como intermediaria entre un cliente y una interfaz. Este patrón permite abstracción de la lógica interna de los subsistemas que se utilizarán y solo es necesario conocer las funcionalidades que este provee.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

En los diagramas de clase se evidencia el empleo de estos patrones, salvo el uso de n-capas ya que en el presente documento solo se describe el desarrollo de la capa de lógica de presentación. Debido a la dimensión de los diagramas estos han sido divididos en dos partes: la primera parte muestra la interacción de los beans manejados con las páginas .jsp, todo esto controlado por JavaServer Faces y la segunda la interacción de los beans manejados con el dominio y las fachadas.

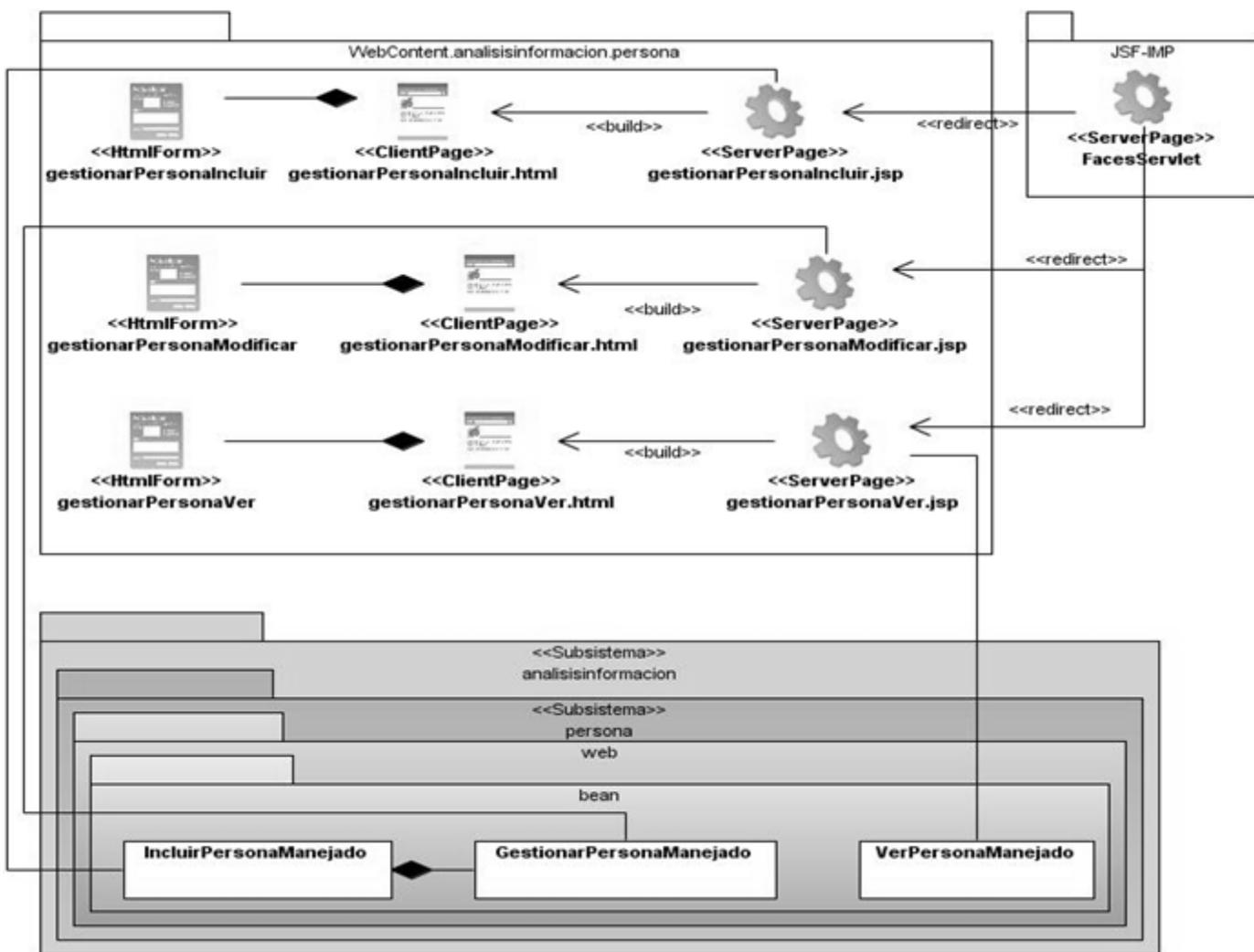


Fig. 2.13 Diagrama de Clases del Diseño Web perteneciente al CU Gestionar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

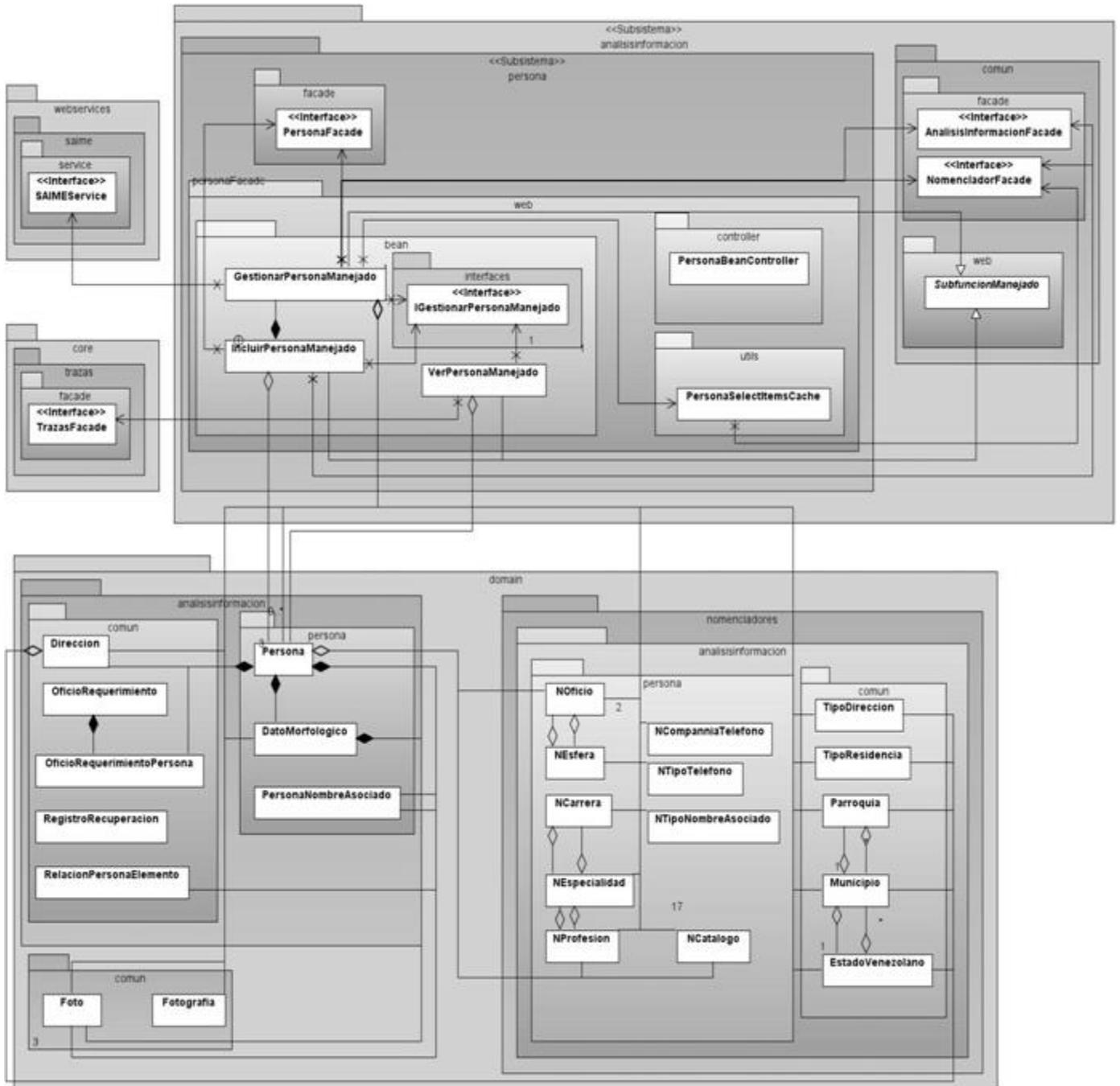


Fig. 2.14 Diagrama de Clases del Diseño perteneciente al CU Gestionar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

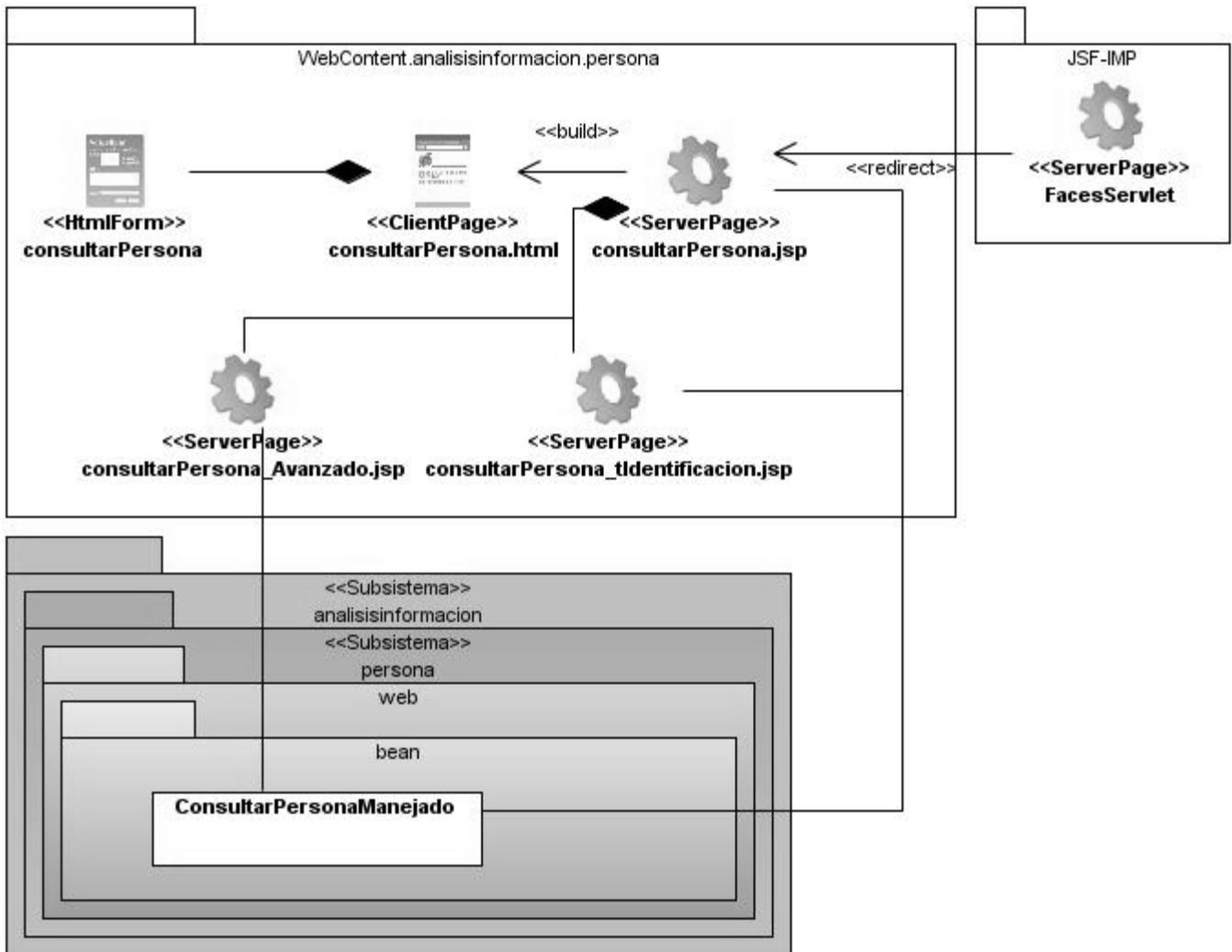


Fig. 2.15 Diagrama de Clases del Diseño Web perteneciente al CU Consultar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

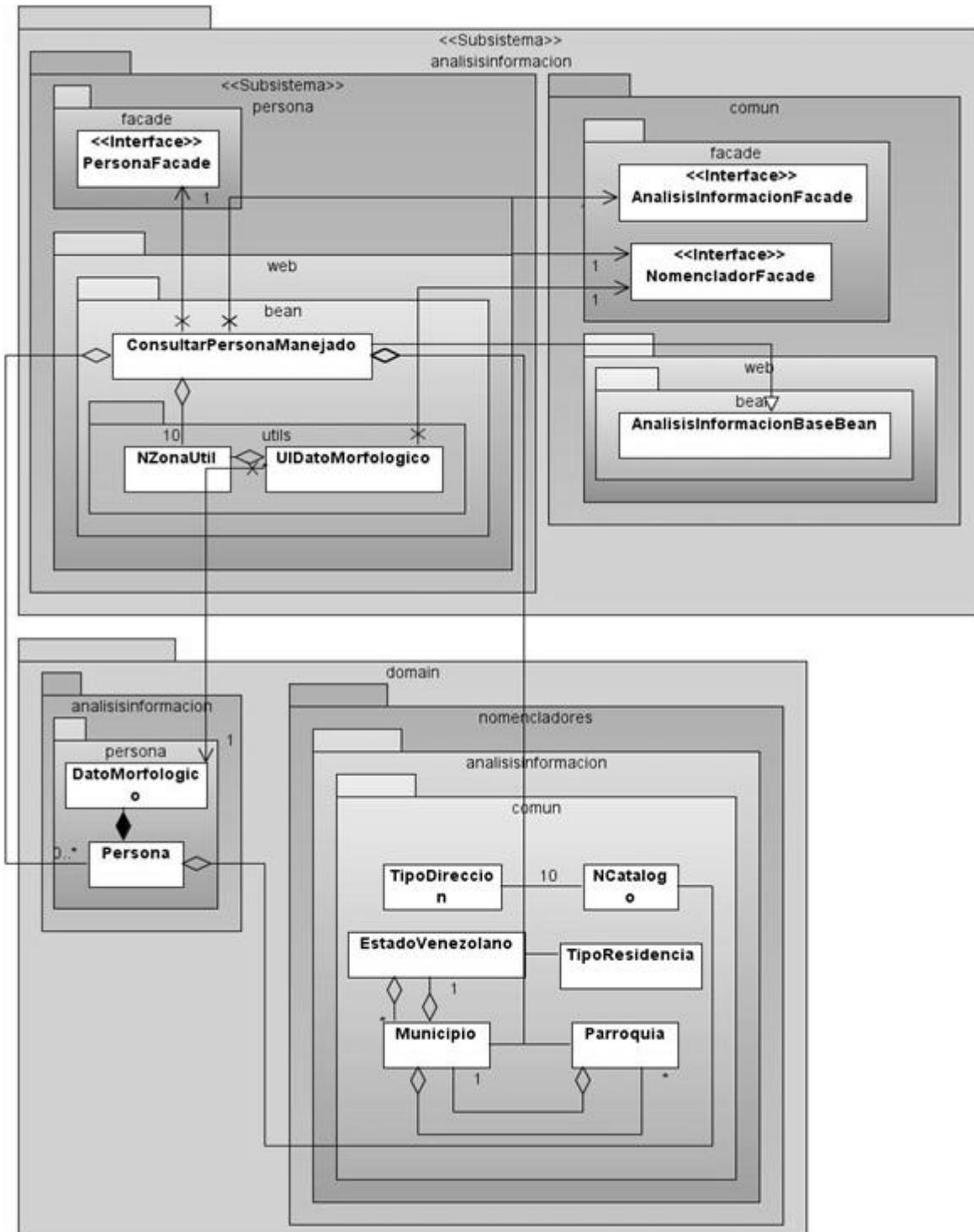


Fig. 2.16 Diagrama de Clases del Diseño perteneciente al CU Consultar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

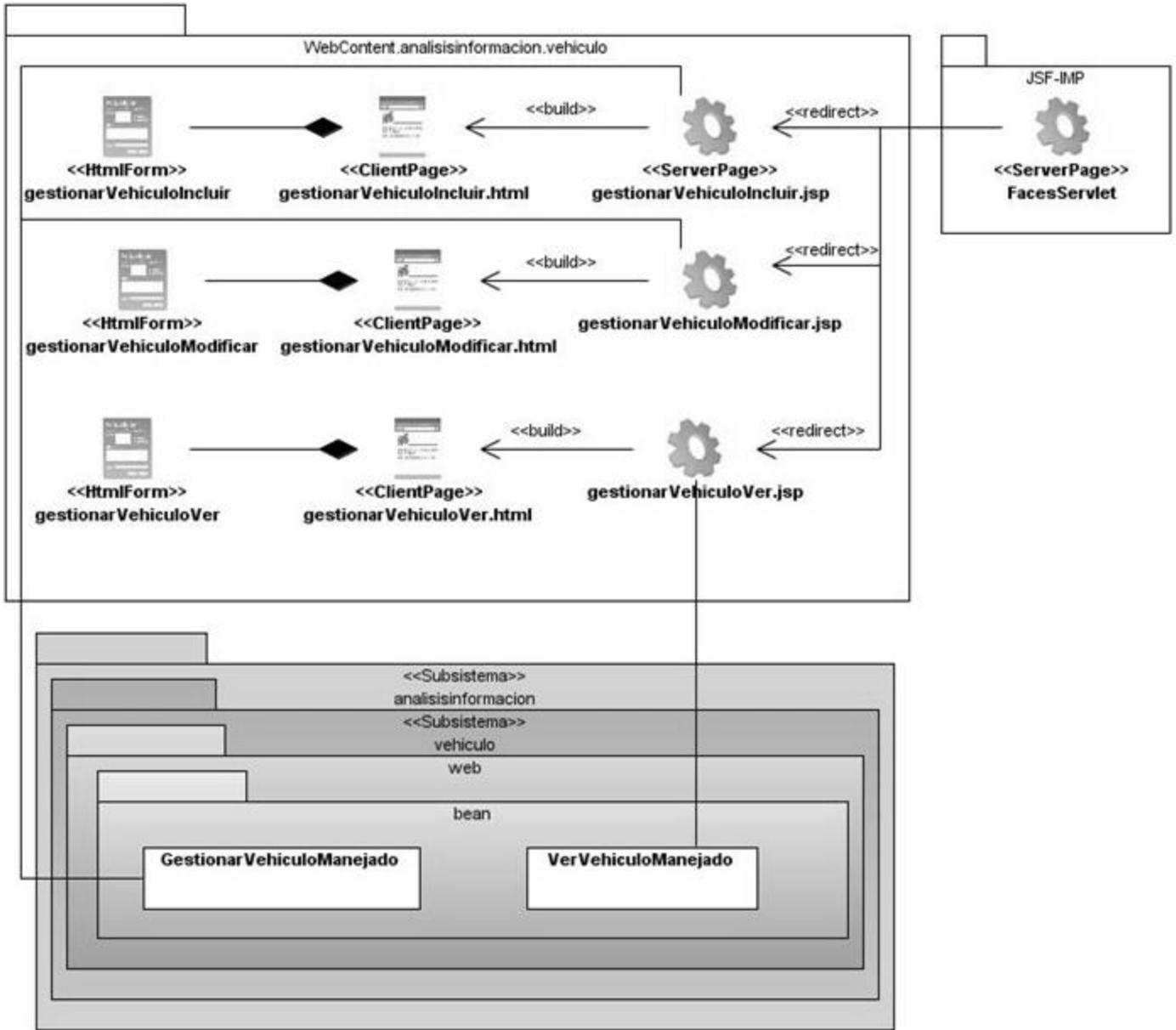


Fig. 2.17 Diagrama de Clases del Diseño Web perteneciente al CU Gestionar Vehículo.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

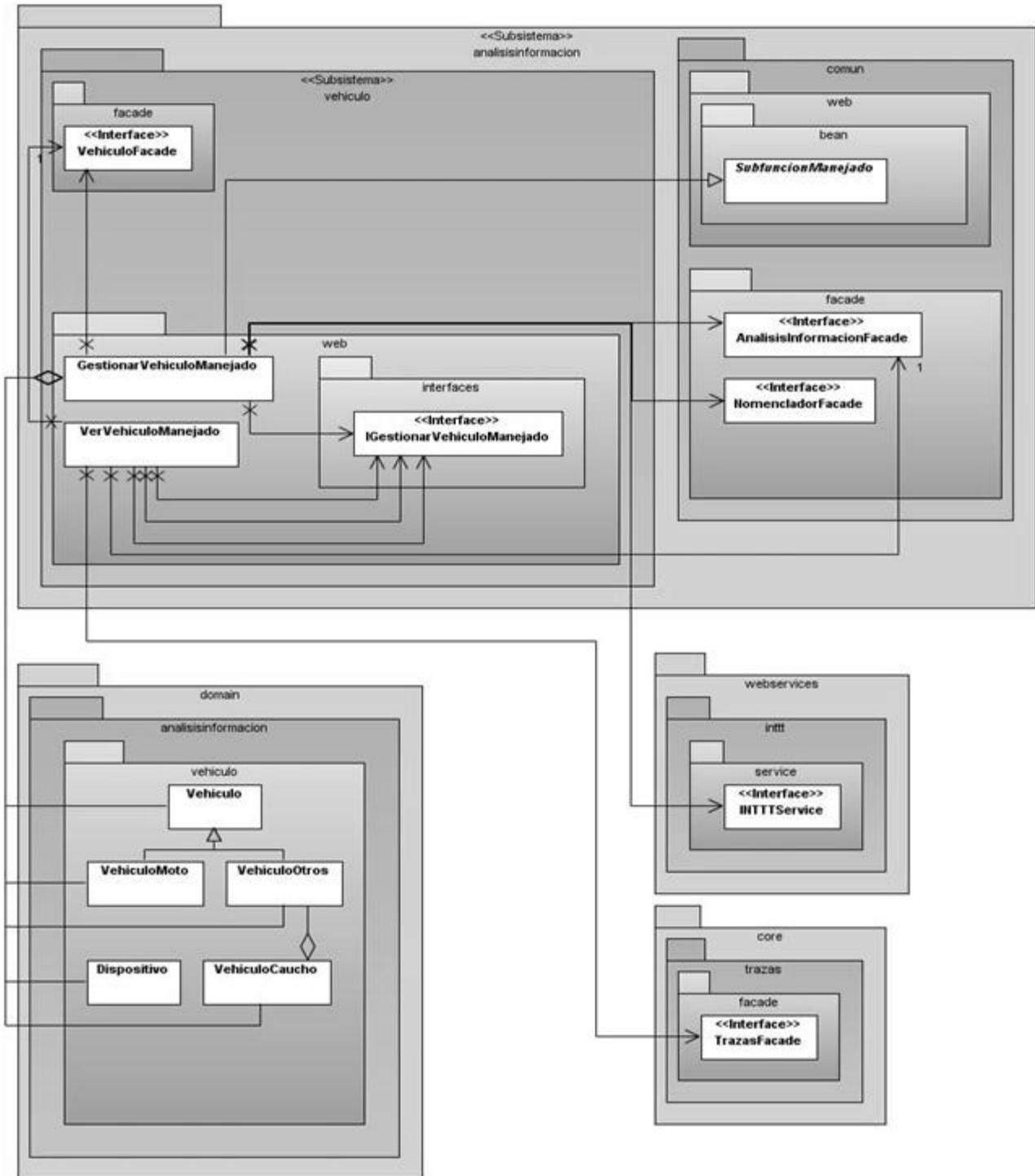


Fig. 2.18 Diagrama de Clases del Diseño perteneciente al CU Gestionar Vehículo.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

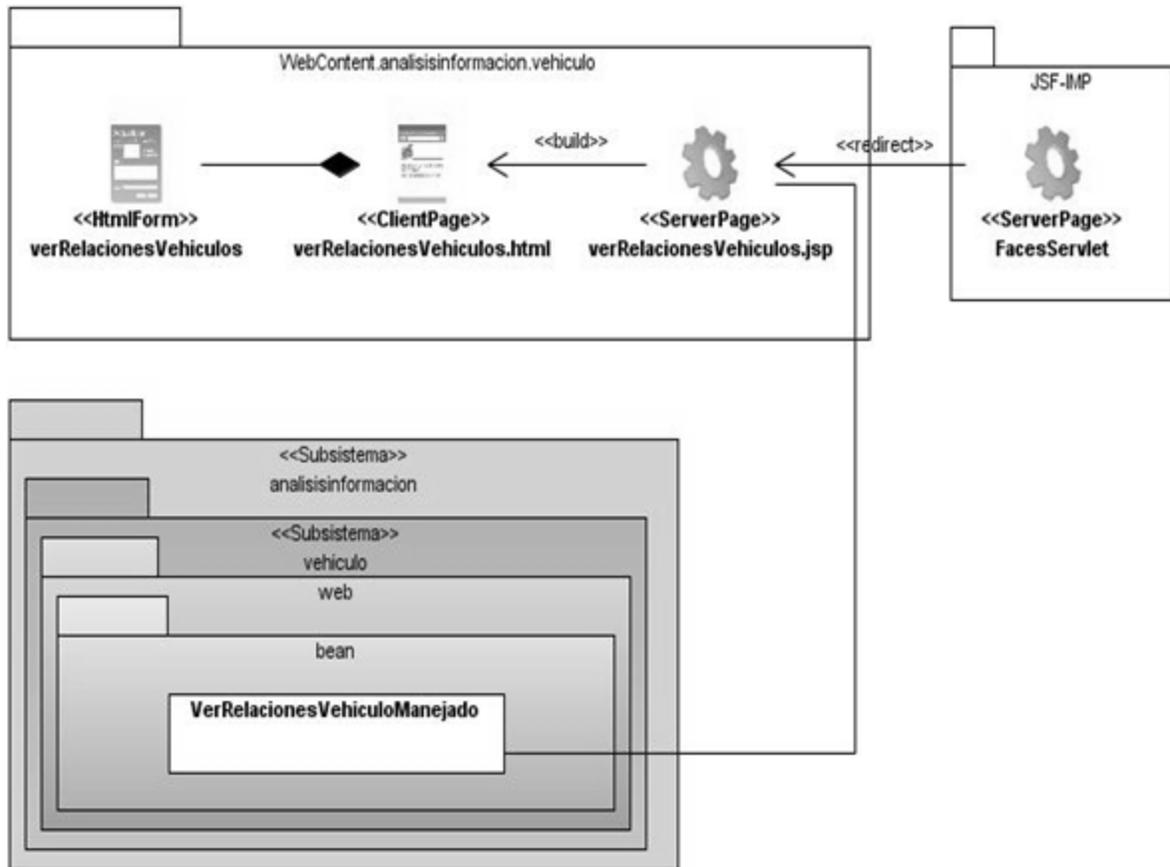


Fig. 2.19 Diagrama de Clases del Diseño Web perteneciente al CU Ver Relaciones de Vehículo.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

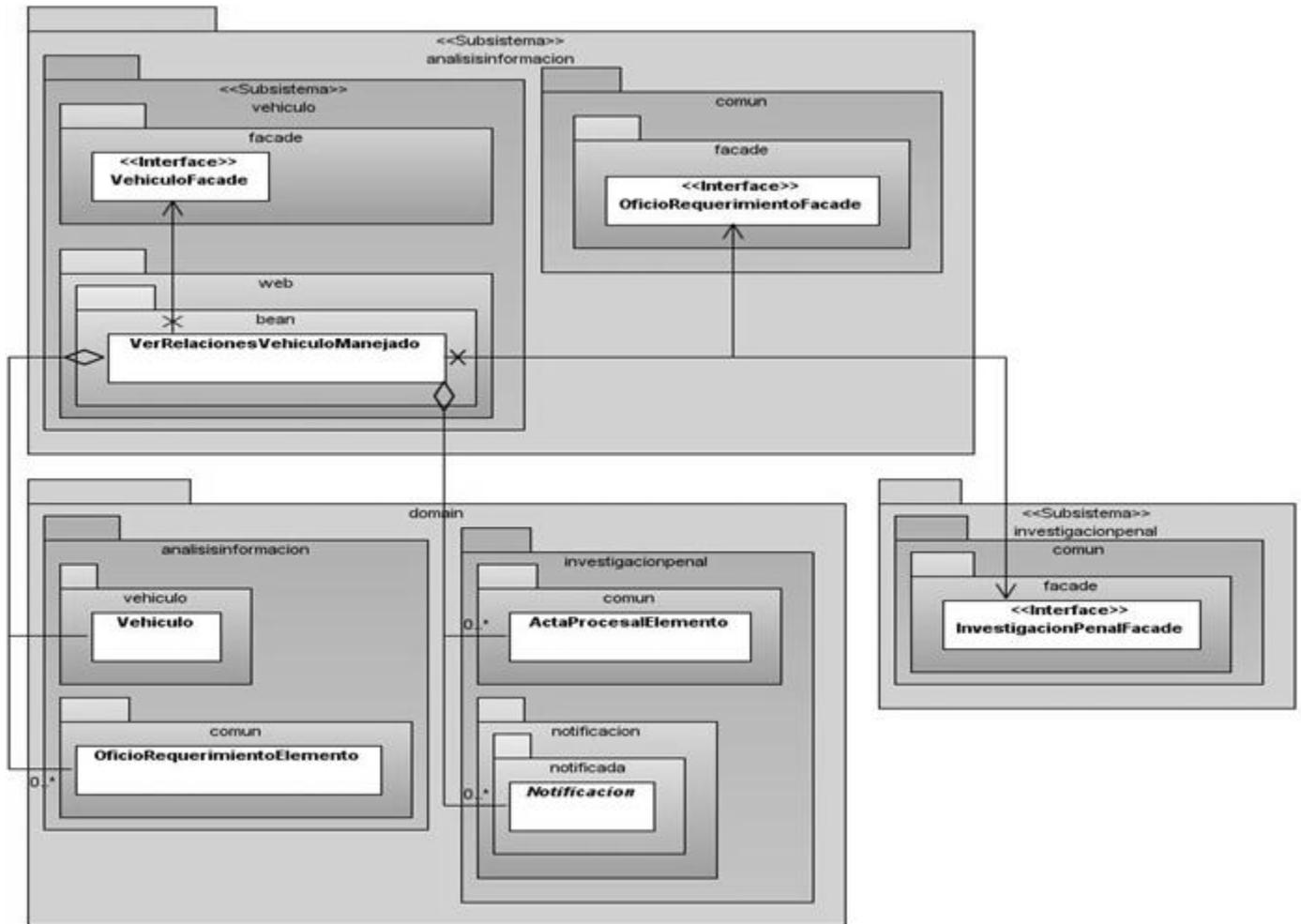


Fig. 2.20 Diagrama de Clases del Diseño perteneciente al CU Ver Relaciones de Vehículo.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

2.4.4 DESCRIPCIONES DE LAS CLASES

A continuación se presenta la descripción de clases más importantes de los CU analizados en el documento. Se describen en tablas que muestran el nombre, tipo de clase así como los atributos y las funcionalidades, de esta última se proporciona una breve descripción de su utilización y funcionamiento.

Tabla 6: Descripción del Bean Manejado GestionarPersonaManejado.

Nombre: GestionarPersonaManejado.	
Tipo de clase: Controladora	
Atributo	Tipo
persona	Persona
personaModificar	Persona
personaSaime	Persona
nomencladorFacade	NomencladorFacade
analisisInformacionFacade	AnalisisInformacionFacade
personaFacade	PersonaFacade
interfaceGestionarPersonaManejado	IGestionarPersonaManejado
imagenSubFuncionManejado	ImagenSubFuncionManejado
actualizarCambios	boolean
indocumentado	boolean
personaSaimeExistente	boolean
modificarNoValidadoSaime	boolean
sameObject	boolean
personaSelectItemCache	PersonaSelectItemCache
currentTab	String
fotografiaNombre	String
fotografiaFechaCaptura	Date

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

fotografiaTipo	NCatalogo
fotografía	Fotografia
fotografiaIndex	int
archivolmagen	UploadFile
selectItemFotoTipo	List<SelectItem>
nacionalidad	String
dtFotografias	UIData
listaNacionalidades	List<NCatalogo>
nombresAsocNombres	String
nombresAsocTipo	NTipoNombreAsociado
nombreAsociadoActual	PersonaNombreAsociado
ocupacionEsfera	NEsfera
ocupacionOficio	NOficio
ocupacionOficioActual	NOficio
profesionCarrera	NCarrera
profesionEspecialidad	NEspecialidad
profesión	NProfesion
profesionActual	NProfesion
telefonoNumero	String
telefonoCompania	NCompanniaTelefono
telefonoTipo	NTipoTelefono
telefonoActual	Telefono
direccionEstado	EstadoVenezolano
direccionMunicipio	Municipio
direccionParroquia	Parroquia
direccionTipo	TipoDireccion

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

direccionTipoResidencia	TipoResidencia
direccionCaserio	String
direccionCalle	String
direccionNoResidencia	String
direccionObservaciones	String
direccionActual	Direccion
selectItemNarizCaracteristicas	List<SelectItem>
selectItemMentonPosicion	List<SelectItem>
narizCaracteristica	NCatalogo
mentonCaracteristica	NCatalogo
cabelloTipo	NCatalogo
cabelloColor	NCatalogo
dentaduraPlanchas	Boolean
dentaduraDesdentado	Boolean
dentaduraPuentes	Boolean
dentaduraCaracteristica	NCatalogo
tipoPiel	NCatalogo
selectItemCabelloTipo	List<SelectItem>
selectItemCabelloColor	List<SelectItem>
selectItemDentaduraCaracteristicas	List<SelectItem>
selectItemPielTipo	List<SelectItem>
selectItemCaracteristicaLabioInferior	List<SelectItem>
selectItemCaracteristicaLabioSuperior	List<SelectItem>
selectItemOjolzquierdoCaracteristica	List<SelectItem>
selectItemOjoDerechoCaracteristica	List<SelectItem>
labioSuperiorCaracteristica	NCatalogo

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

labioInferiorCaracteristica	NCatalogo
ojoDerechoCaracteristica	NCatalogo
ojoIzquierdoCaracteristica	NCatalogo
sennaTipo	Integer
sennaZona	Integer
sennaLugar	Integer
sennaLocalizacion	Integer
sennaObservaciones	String
nombreImagenMorfologicos	String
fotoDatosMorfologicos	Foto
archivoImagenMorfologicos	UploadedFile
fotografiaDMIndex	int
datoMorfologicoActual	DatoMorfologico
dtDatosMorfologicos	UIData
selectItemCaracteristicaVoz	List<SelectItem>
selectItemTics	List<SelectItem>
selectItemHabitos	List<SelectItem>
selectItemCaracteristicasCaminar	List<SelectItem>
selectItemCondicionesFisicas	List<SelectItem>
caracteristicaVoz	NCatalogo
tic	NCatalogo
habito	NCatalogo
caracteristicaCaminar	NCatalogo
caracteristicaCondicionFisica	NCatalogo
Para cada responsabilidad:	
Nombre:	incluirPersona(ActionEvent event)

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Descripción:	Esta función se usa cuando se presiona el botón Incluir, realiza las acciones necesarias para incluir la persona y retorna a la vista anterior.
Nombre:	actualizarPersonaConSaime(ActionEvent event)
Descripción:	Esta función se usa para actualizar el objeto persona con los datos proporcionados por el servicio externo SAIME.
Nombre:	actualizarPersonaSinSaime(ActionEvent event)
Descripción:	Esta función se usa para terminar el flujo de eventos de Incluir Persona sin actualizar los datos de esta con el servicio externo SAIME.
Nombre:	cancelarIncluir(ActionEvent event)
Descripción:	Esta función se usa cuando el usuario presiona el botón Cancelar, saliendo de la ventana del incluir y retorna a la vista anterior.
Nombre:	finalizar(boolean verificarSaime)
Descripción:	Esta función es llamada cada vez que se va a terminar un flujo, ya sea modificar o incluir y realiza las acciones necesarias para llevar a cabo esta acción. Si el parámetro <i>verificarSaime</i> esta true se verifica con SAIME si existe esta persona.
Nombre:	modificarPersona(Persona persona, boolean actualizarCambios)
Descripción:	Esta función es llamada cuando se va a modificar la persona, donde <i>persona</i> es el objeto de la persona que va a ser modificada y <i>actualizarCambios</i> si esta true se hacen persistir esos cambios en la Base de Datos cuando se llame a la función finalizar().
Nombre:	incluirPorCedula(Character letraCedula, String numeroCedula, String numeroPasaporte, boolean actualizarCambios)
Descripción:	Esta función se utiliza para preparar el objeto <i>persona</i> para el flujo de eventos de Incluir Persona. Se le pasan por parámetros algunos de los atributos fundamentales de la persona.
Nombre:	preparePersona(Persona persona)
Descripción:	Esta función es utilizada para preparar el objeto persona para el flujo Incluir, donde son inicializados todos los atributos del objeto correctamente.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Tabla 7: Descripción del Bean Manejado ConsultarPersonaManejado.

Nombre: ConsultarPersonaManejado	
Tipo de clase: Control.	
Atributo	Tipo
nomencladorFacade	NomencladorFacade
personaFacade	PersonaFacade
<u>analisisInformacionFacade</u>	<u>AnalisisInformacionFacade</u>
busquedaBasica	boolean
persona	Persona
indiceVer	int
indocumentado	boolean
listaResultados	List<Persona>
selectItemEstado	List<SelectItem>
nacionalidad	NCatalogo
nombresAsocNombre	String
direccionEstado	EstadoVenezolano
direccionMunicipio	Municipio
direccionParroquia	Parroquia
direccionTipo	TipoDireccion
direccionTipoResidencia	TipoResidencia
direccionCaserio	String
direccionCalle	String
direccionNoResidencia	String
direccionNoResidencia	NCatalogo
cabelloColor	NCatalogo
tipoPiel	NCatalogo
listaDatosMorfologicos	List<UIDatoMorfologico>

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

caracteristicaVoz	NCatalogo
tic	NCatalogo
habito	NCatalogo
caracteristicaCaminar	NCatalogo
<u>dentaduraCaracteristicas</u>	NCatalogo
caracteristicaCondicionFisica	NCatalogo
datoMorfAmputaciones	DatoMorfologico
datoMorfManchas	DatoMorfologico
datoMorfCicatrices	DatoMorfologico
datoMorfTatuajesDibujos	DatoMorfologico
datoMorfDeformidades	DatoMorfologico
datoMorfTatuajesCaracteres	DatoMorfologico
datoMorfLunares	DatoMorfologico
datoMorfExtremidadesArtificiales	DatoMorfologico
datoMorfQuemaduras	DatoMorfologico
datoMorfPiercing	DatoMorfologico
<u>selectItemEstadoCivil</u>	List<SelectItem>
<u>selectItemSexo</u>	List<SelectItem>
selectItemRangoEdad	List<SelectItem>
selectItemRangoEstatura	List<SelectItem>
selectItemRangoPeso	List<SelectItem>
selectItemNacionalidad	List<SelectItem>
<u>selectItemGradoInstruccion</u>	List<SelectItem>
<u>selectItemPais</u>	List<SelectItem>
selectItemTipoDireccion	List<SelectItem>
selectItemTipoResidencia	List<SelectItem>
selectItemNarizTamanno	List<SelectItem>
<u>selectItemNarizForma</u>	List<SelectItem>
selectItemMentonForma	List<SelectItem>

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

selectItemMentonPosicion	List<SelectItem>
selectItemColorOjolzquierdo	List<SelectItem>
selectItemTipoOjolzquierdo	List<SelectItem>
selectItemDentaduraPlanchas	List<SelectItem>
selectItemDentaduraPuentes	List<SelectItem>
selectItemDentaduraDesdentado	List<SelectItem>
selectItemDentaduraCaracteristicas	List<SelectItem>
selectItemCabelloColor	List<SelectItem>
selectItemCabelloTipo	List<SelectItem>
selectItemPielColor	List<SelectItem>
selectItemPielTipo	List<SelectItem>
selectItemOrejalzqTamanno	List<SelectItem>
selectItemOrejalzqLobulo	List<SelectItem>
selectItemBarba	List<SelectItem>
selectItemBigote	List<SelectItem>
selectItemPatillas	List<SelectItem>
selectItemManeraHablar	List<SelectItem>
selectItemTipoVoz	List<SelectItem>
selectItemFormaCaminar	List<SelectItem>
selectItemCaracteristicasCaminar	List<SelectItem>
selectItemCondicionesFisicas	List<SelectItem>
selectItemTics	List<SelectItem>
selectItemHabitos	List<SelectItem>
selectItemContextura	List<SelectItem>
selectItemOrientacionSexual	List<SelectItem>
selectItemOtrasCondiciones	List<SelectItem>
listaZonaAmputaciones	List<NZonaUtil>
listaZonaCicatrices	List<NZonaUtil>
listaZonaDeformidades	List<NZonaUtil>

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

listaZonaLunares	List<NZonaUtil>
listaZonaQuemaduras	List<NZonaUtil>
listaZonaManchas	List<NZonaUtil>
listaZonaTatuajesDibujos	List<NZonaUtil>
listaZonaTatuajesCaracteres	List<NZonaUtil>
listaZonaExtremidadesArtificiales	List<NZonaUtil>
listaZonaPiercing	List<NZonaUtil>
parametros	Map<String, Object>
Para cada responsabilidad:	
Nombre:	cambiarBusqueda(ActionEvent event)
Descripción:	Esta función se utiliza para cambiar de búsqueda estándar a búsqueda avanzada.
Nombre:	mantenerDatos()
Descripción:	Esta función se utiliza para guardar en un objeto temporal los datos de una búsqueda.
Nombre:	nuevaBusqueda(ActionEvent actionEvent)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Nueva Búsqueda para limpiar todos los campos y la página quede lista para una nueva búsqueda.
Nombre:	cancelar(ActionEvent event)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Cancelar, se cancela la búsqueda y regresa a la vista anterior.
Nombre:	buscarCoincidencias(ActionEvent event)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Consultar, se realiza la consulta y se muestran los resultados.
Nombre:	prepareObject()
Descripción:	Esta función prepara el objeto persona a buscar dejándolo listo para realizar las búsquedas.
Nombre:	verPersona(ActionEvent event)
Descripción:	Esta función se encarga de re-direccionar hacia la vista de Ver Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Tabla 7: Descripción del Bean Manejado GestionarVehiculoManejado.

Nombre: GestionarVehiculoManejado.	
Tipo de clase: Control.	
Atributo	Tipo
fase	int
coincidencias	boolean
igestionarVehiculo	IGestionarVehiculoManejado
actualizarCambios	boolean
modificando	boolean
coincidenciasModificando	boolean
nomencladorFacade	NomencladorFacade
vehiculoFacade	VehiculoFacade
intttService	INTTTService
vehiculo	Vehiculo
vehiculoMoto	VehiculoMoto
vehiculoModificar	Vehiculo
vehiculoINTTT	Vehiculo
moto	Boolean
listaVehiculos	List<Vehiculo>
tablaCoincidencias	UIData
selectItemsEstadoPlaca	List<SelectedItem>
selectItemsClaseVehiculo	List<SelectedItem>
selectItemsUsoPlaca	List<SelectedItem>
selectItemsTipoVehiculo	List<SelectedItem>
selectItemsColor	List<SelectedItem>
selectItemsMoneda	List<SelectedItem>
selectItemsSiNoRoto	List<SelectedItem>
selectItemsSiNoV	List<SelectedItem>

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

selectItemsSiNoBoolean	List<SelectItem>
selectItemsSiNoString	List<SelectItem>
selectItemsMaterialRines	List<SelectItem>
selectItemsTipoDispositivo	List<SelectItem>
selectItemsLocalizacionCaucho	List<SelectItem>
selectItemEstadoCarroceria	List<SelectItem>
selectItemEstadoMotor	List<SelectItem>
selectItemEstadoChasis	List<SelectItem>
selectItemEstadoChapa	List<SelectItem>
dispositivo	Dispositivo
vehiculoCaucho	VehiculoCaucho
listaDispositivos	List<Dispositivo>
listaCauchos	List<VehiculoCaucho>
Para cada responsabilidad:	
Nombre:	cancelar(ActionEvent e)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Cancelar, se cancela todo y se regresa a la vista anterior.
Nombre:	AsignarClase()
Descripción:	Esta función se utiliza para asignar la clase al vehículo que se va a insertar teniendo en cuenta el número de placa del mismo.
Nombre:	buscarClase(String nombre)
Descripción:	Esta función retorna un objeto tipo ClaseVehiculo dado un nombre.
Nombre:	consultarVehiculo(ActionEvent e)
Descripción:	Esta función se utiliza para buscar coincidencias a partir de los datos suministrados inicialmente por el usuario.
Nombre:	seleccionarVehiculo(ActionEvent e)
Descripción:	Esta función se utiliza cuando el usuario selecciona un vehículo del listado de coincidencias.
Nombre:	nuevoVehiculo(ActionEvent e)

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Descripción:	Esta función se utiliza para crear un nuevo vehículo obviando los obtenidos en el listado de coincidencias.
Nombre:	incluirVehiculo(ActionEvent e)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Incluir e incluye el nuevo vehículo.
Nombre:	actualizarVehiculo()
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Actualizar y se actualizan los datos del vehículo.
Nombre:	reActualizar(ActionEvent actionEvent)
Descripción:	Esta función se utiliza cuando el usuario decide actualizar los datos del vehículo sin importar las alertas que muestre el sistema.
Nombre:	setVehiculoModificar(Vehiculo carr)
Descripción:	Esta función se utiliza para asignarle al bean manejado el vehículo que se va a modificar.
Nombre:	validarINTT()
Descripción:	Esta función se utiliza para validar los datos del vehículo con INTTT.
Nombre:	comparar()
Descripción:	Esta función se utiliza para comparar el vehículo que tiene el vean manejado con el devuelto por INTTT.

Tabla 7: Descripción del Bean Manejado VerRelacionesVehiculoManejado.

Nombre: VerRelacionesVehiculoManejado.	
Tipo de clase: Control.	
Atributo	Tipo
vehiculo	Vehiculo
dataTableNotificaciones	UIData
dataTable	UIData
dataTableOficio	UIData

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

showTableActaProcesal	boolean
showTableNotificaciones	boolean
showTableOficio	boolean
listaNotificaciones	List<Notificacion>
actaProcesal	List<ActaProcesalElemento>
oficioRequerimiento	List<OficioRequerimientoElemento>
oficioRequerimientoFacade	OficioRequerimientoFacade
Para cada responsabilidad:	
Nombre:	aceptar(ActionEvent e)
Descripción:	Esta función se utiliza cuando el usuario presiona el botón Aceptar, retorna a la vista anterior.
Nombre:	verOficioRequerimiento(ActionEvent e)
Descripción:	Esta función se utiliza cuando se van a mostrar los oficios de requerimientos, se re-direcciona para la página que los muestra.
Nombre:	setVehiculo(Vehiculo vehiculo)
Descripción:	Esta función se utiliza para asignarle al bean manejado el vehículo al que se le buscarán las relaciones a mostrar.

2.4.5 REALIZACIONES DE CASOS DE USO

La secuencia de acciones en un Caso de Uso comienza cuando un actor mediante el envío de algún tipo de mensaje al sistema invoca el Caso de Uso. Si consideramos el interior del sistema, tendremos algún objeto de Diseño que recibe el mensaje del actor. Después el objeto de Diseño llama a algún otro objeto, y de esta manera los objetos implicados interactúan para realizar y llevar a cabo el CU. En el Diseño, es preferible representar esto con diagramas de secuencia ya que el centro de atención principal es el de encontrar secuencias de interacciones detalladas y ordenadas en el tiempo. ⁽⁴⁾

Las realizaciones de CU se harán a través de Diagramas de Contrato entre Paquetes los cuales son diagramas de secuencia a un alto nivel que solo muestran el paso de mensajes entre los diferentes paquetes, abstrayéndose de lo que sucede dentro de estos. El objetivo de estos diagramas es brindar a los desarrolladores la información necesaria para la realización de los CU sin incurrir en un nivel de

detalles muy alto. Debido a que la realización de estos diagramas no requiere demasiadas especificaciones y simplifica de manera muy considerable los diagramas necesarios, no consume mucho tiempo, disminuyendo así el período de desarrollo.

Se ha elaborado un Diagrama de Contrato entre Paquetes para los CU más significativos de los Sub-Módulos Persona y Vehículo buscando una mayor comprensión y claridad de los mismos. En el presente documento se mostrarán los escenarios más significativos debido al tamaño que ocupan estos diagramas.

Dentro del CU Gestionar Persona uno de los escenarios más complicados es el de modificar. Una gran parte de los mensajes están destinados a cargar los nomencladores de persona, dado que son muchos, se resumen todas las llamadas en una sola: `listarNomencladoresPersona()`, los otros mensajes están destinados a realizar consultas al servicio externo que brinda SAIME y a garantizar que se actualicen los datos de la persona. (Fig. 2.21)

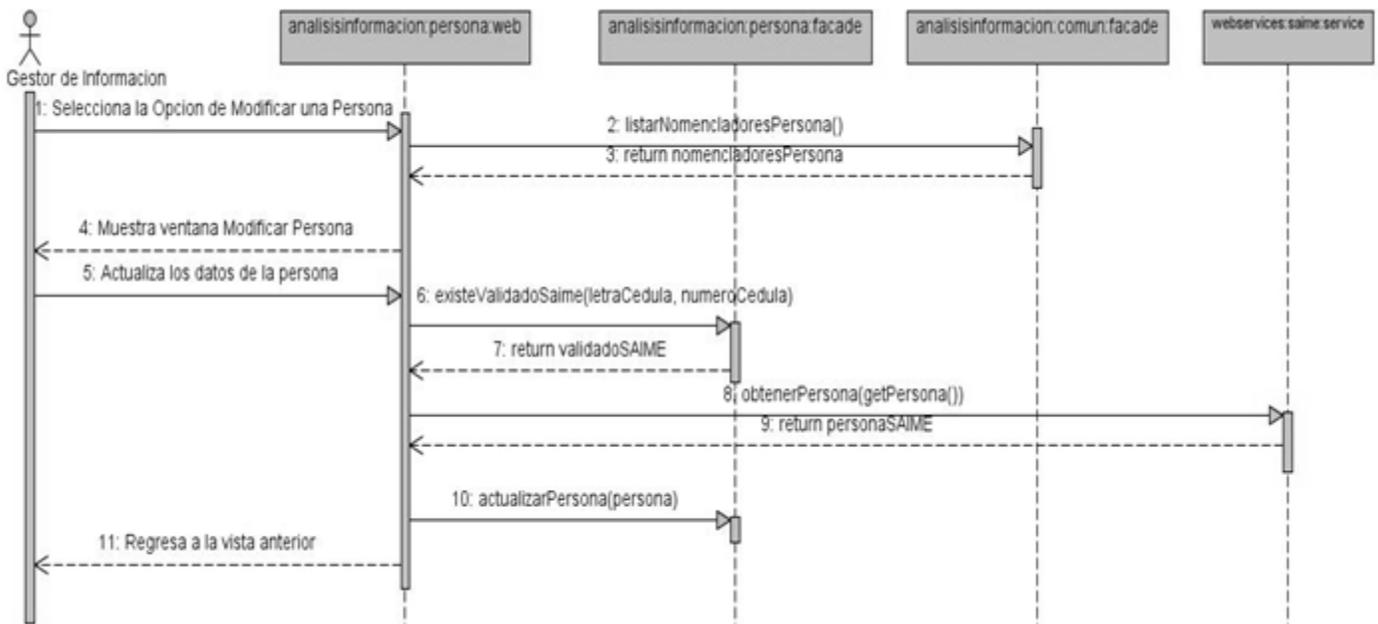


Fig. 2.21 Diagrama de Contrato entre Paquetes del CU Gestionar Persona: Modificar.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

En el CU Consultar Persona se procede a cargar los nomencladores necesarios para Persona cuando se decide a realizar una búsqueda avanzada y a realizar consultas a partir de los criterios de búsqueda proporcionados por el actor. (Fig. 2.22)

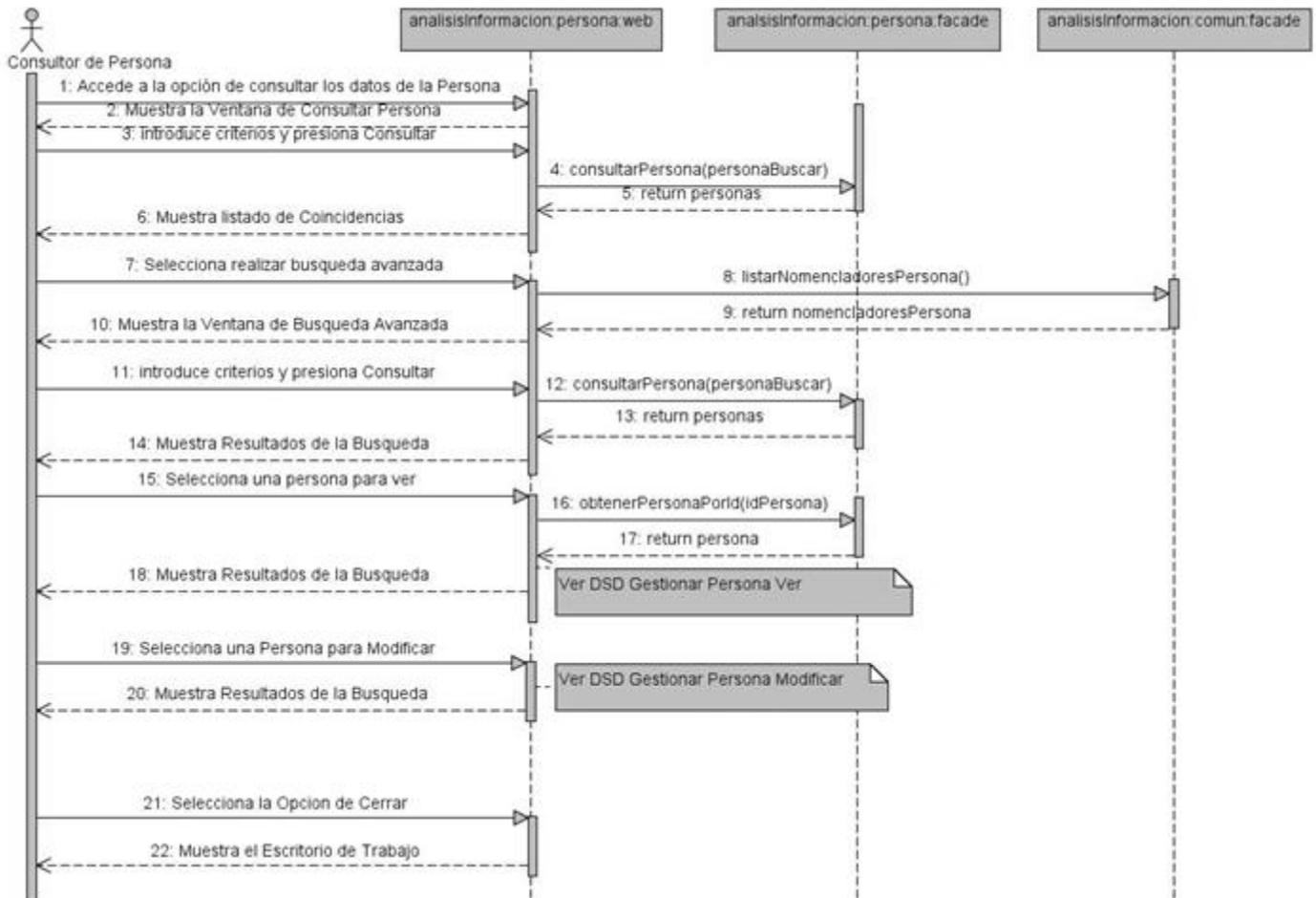


Fig. 2.22 Diagrama de Contrato entre Paquetes del CU Consultar Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E
IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO
PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Para ver las relaciones de un vehículo es necesario establecer comunicación con varios subsistemas ya que estos poseen las entidades que guardan relación con el mismo, de ahí que existan mensajes para tantos paquetes ajenos al subsistema en estudio. (Fig. 2.23)

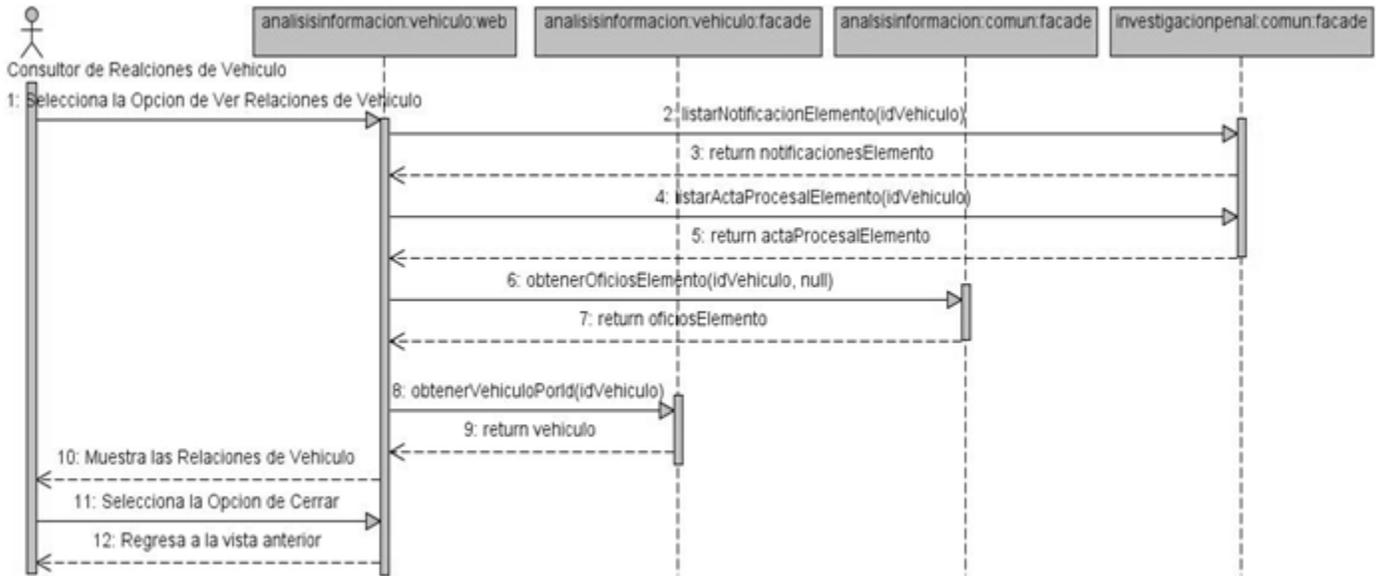


Fig. 2.23 Diagrama de Contrato entre Paquetes del CU Ver Relaciones de Vehículo.

2.4.6 DIAGRAMA DE CLASES PERSISTENTES

En la siguiente figura se muestra el Diagrama de las Clases Persistentes relacionadas a la lógica de presentación de los Sub-Módulos Persona y Vehículo, así como las relaciones entre estas. (Fig. 2.24)

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

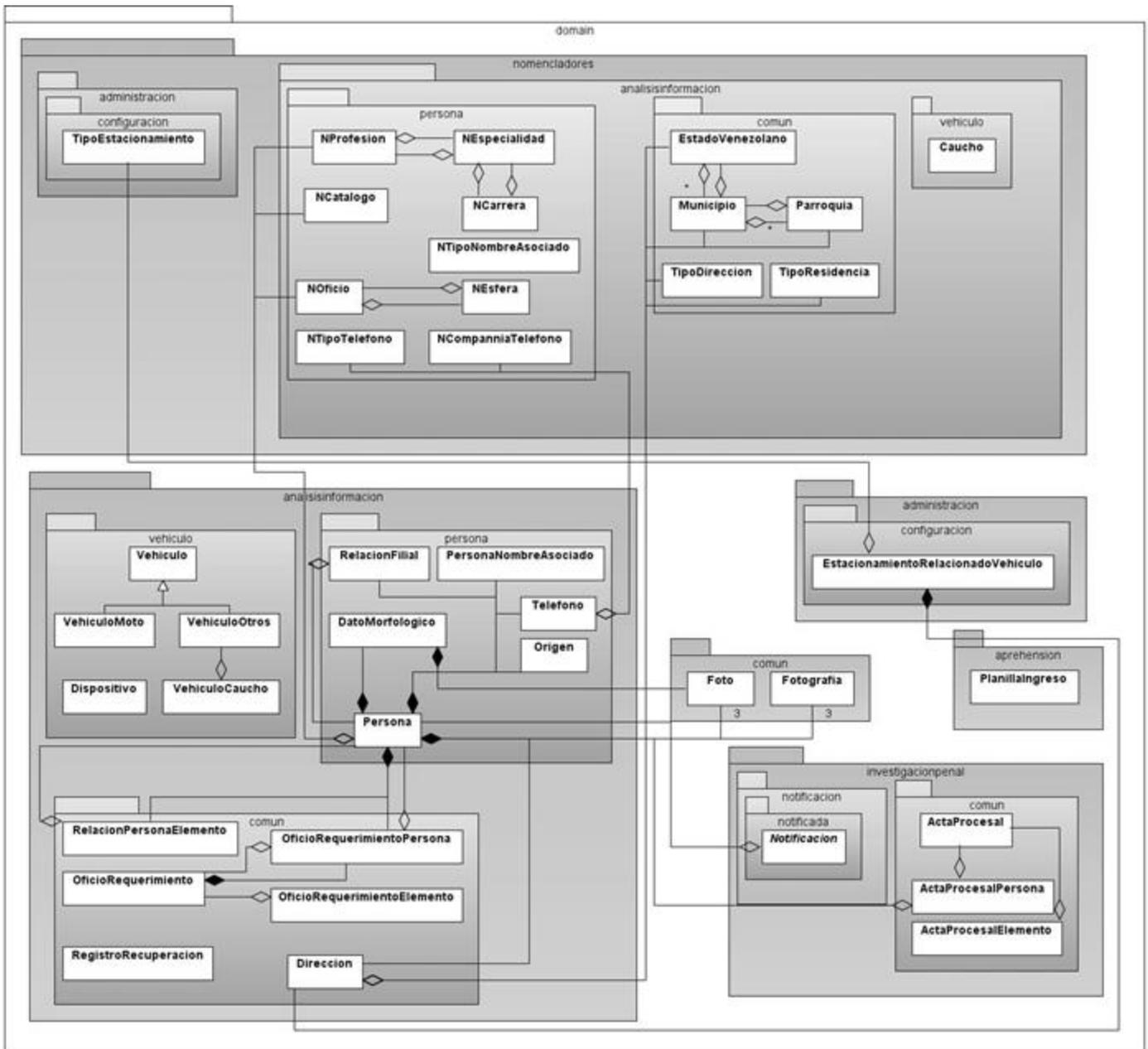


Fig. 2.24 Diagrama de Clases Persistentes.

2.5 MODELO DE IMPLEMENTACIÓN

2.5.1 INTRODUCCIÓN AL FLUJO DE IMPLEMENTACIÓN

La implementación es el centro durante las iteraciones de construcción, aunque también se lleva a cabo durante la fase de elaboración para crear una línea base ejecutable de la arquitectura. En este flujo se implementa el sistema en términos de componentes, es decir, ficheros de códigos fuentes, scripts, ficheros de código binario, ejecutables y similares. ⁽⁴⁾

2.5.2 DIAGRAMAS DE SUBSISTEMAS DE IMPLEMENTACIÓN

Los subsistemas de implementación proporcionan una forma de organizar los artefactos del Modelo de Implementación y están estrechamente relacionados con los subsistemas obtenidos durante el flujo de Diseño. El Diagrama de Subsistemas de Implementación para los Sub-módulos Persona y Vehículo se presenta en la Figura 2.25.

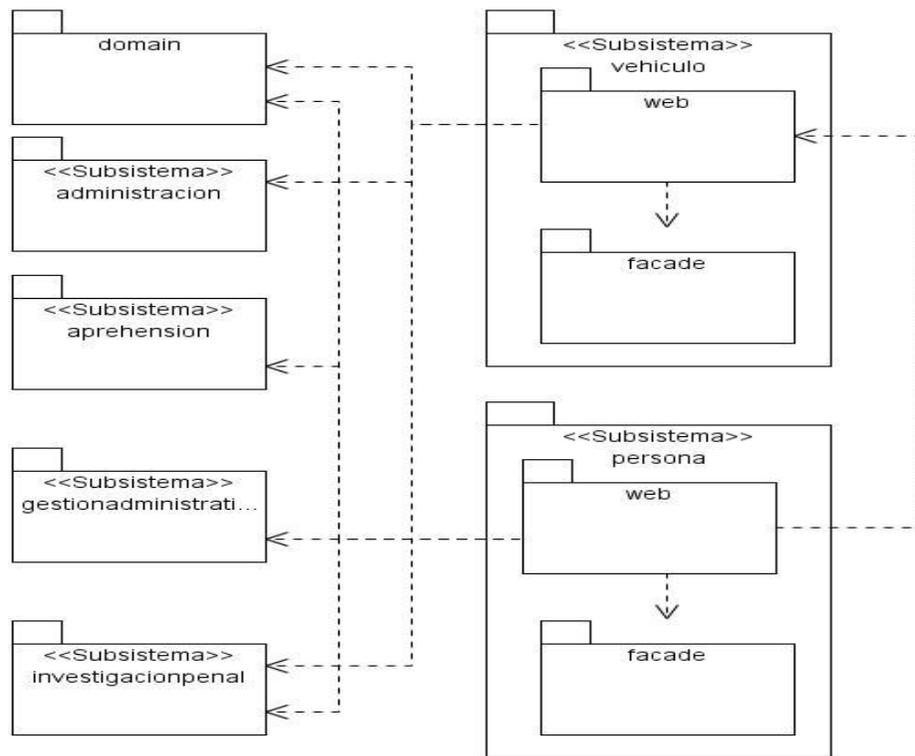


Fig. 2.25 Diagrama de Subsistemas de Implementación.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

2.5.3 DIAGRAMA DE COMPONENTES

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el Modelo de Diseño. Los Diagramas de Componentes muestran los elementos obtenidos durante la implementación así como sus relaciones.

En la Figura 2.26 se muestra uno de estos diagramas, los restantes se encuentran en el Anexo 5.

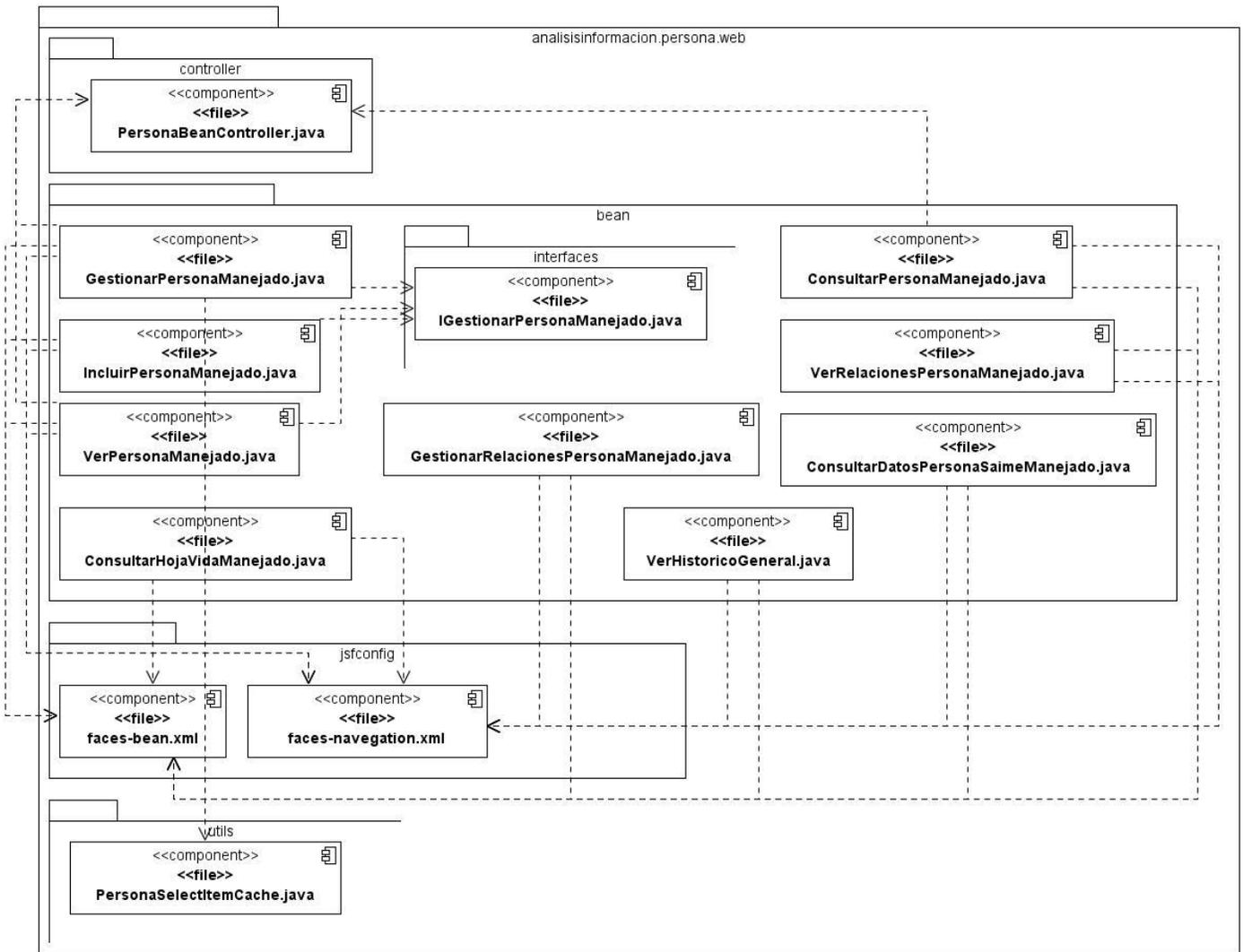


Fig. 2.26 Diagrama de Componentes de Implementación perteneciente al Sub-Módulo Persona.

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

2.5.4 ESTÁNDAR DE CODIFICACIÓN

Las convenciones de código son sugerencias que no se centran en la lógica de un programa, sino en la apariencia y la estructura física. Si se cumplen, se facilita la lectura, la comprensión y el mantenimiento del código. Las convenciones de código pueden incluir:

- Formatos estandarizados para etiquetar y comentar código.
- Instrucciones para agregar espacios, dar formato y aplicar sangría al código.
- Convenciones de nomenclatura para objetos, variables y procedimientos.

Los temas siguientes presentan un conjunto de directrices de programación que engloban estándares globales y especificaciones propias determinadas por el equipo de arquitectura del proyecto.

Convenciones de nombres

Todo lo referido a convención de nombre se regirá en última instancia por la “Convención de código de Java”, lo que quiere decir que lo que no esté especificado en este documento se hará tal como venga indicado en la “Convención de código de Java”.

Como principio todos los nombres serán en español exceptuando aquellos que correspondan al nombre de un patrón conocido, ejemplo: ServiceLocator, Facade y Builder.

Paquetes

Los nombres de todos los paquetes comenzarán por cicpc, este será nuestro prefijo de alto nivel, luego seguirán los nombres de acuerdo a la estructura de paquetes definida por la arquitectura:

```
<cicpc>.<nombre del subsistema>.<nombre del módulo>. <facade\web\config\dao\service\model>  
<cicpc>.<core><config\web\mensajeria\seguridad\util>
```

Clases

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Se utilizarán palabras completas, evitando acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL o HTML).

Interfaces

Los nombres de las interfaces siguen las mismas reglas que las clases.

Métodos

Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula y la primera letra de las siguientes palabras que lo forma en mayúscula.

Ejemplo: enviarSolicitud().

Variables

Excepto las constantes, todas las instancias y variables de clase o método comenzarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben comenzar con los caracteres subguión "_" o signo del dólar "\$", aunque ambos están permitidos por el lenguaje.

Los nombres de las variables deben ser cortos pero con significado. Los nombres de variables de un solo caracter estarán incorrectos, excepto para variables índices temporales, como son las de ciclos o para hacer intercambio de valores. Nombres comunes para variables temporales son *i, j, k, m, y n* para enteros; *c, d, y e* para caracteres.

Constantes

Los nombres de las variables declaradas como constantes deben escribirse en mayúsculas separando las palabras con un subguión ("_").

Ejemplo: ANCHURA_MAXIMA = 999;

Contextos de JSF

Estarán dentro del paquete web.jsfconfig de cada módulo y se nombrarán siempre con el prefijo faces- , el xml de navegación se nombrará faces-navigation.

2.6 CONCLUSIONES

En el presente capítulo se describió el Análisis, el Diseño y la Implementación de la capa de lógica de presentación de los Sub-Módulos Persona y Vehículo pertenecientes al subsistema de Análisis de Información del SIIPOL, utilizando como principal fuente de información la descripción de los Casos de

CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SOLUCIÓN | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

Uso de estos. Se obtuvieron varios artefactos como entregables, los principales son los Diagramas de Clases del Análisis y del Diseño, la realización de Casos de Uso de las clases del Diseño así como los Componentes de Implementación. Se tuvo en cuenta para su elaboración la utilización de los patrones definidos y la integración con el framework JSF aprovechando las facilidades del mismo.

3 VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 INTRODUCCIÓN

En el flujo de trabajo de la prueba se verifica el resultado de la implementación, probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros. ⁽⁴⁾

3.2 TIPOS Y NIVELES DE PRUEBA

Con la realización de las pruebas se pretende demostrar que hay errores, en ningún momento es posible asegurar lo contrario. El principal objetivo al diseñar las pruebas es tratar de ejecutar todos los flujos posibles, para descubrir la mayor cantidad de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Tipos de prueba: ⁽⁹⁾

- Pruebas de caja negra: Pruebas que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene (no se analiza el código).
- Pruebas de caja blanca: Se comprueban los caminos lógicos del software. Se examina la lógica interna del programa sin considerar los aspectos de rendimiento así como la estructura interna del mismo.

3.2.1 PRUEBAS DE CAJA NEGRA

También conocidas como Pruebas de Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas, para efectuarlas se selecciona un conjunto de posibles entradas y salidas sobre las que se realizarán las pruebas. Hay que tener en cuenta que en todo programa existe un

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

conjunto de entradas que causan un comportamiento erróneo en el sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos. ⁽⁹⁾

Para confeccionar los casos de prueba de Caja Negra existen distintos criterios. Algunos de ellos son:

- Particiones de Equivalencia.
- Análisis de Valores Límite.
- Métodos Basados en Grafos.
- Pruebas de Comparación.
- Análisis Causa-Efecto.

Las pruebas de Caja Negra fueron las utilizadas para validar la capa de presentación de los Sub-Módulos Persona y Vehículo, así como para comprobar el desempeño de los requisitos funcionales y no funcionales y el nivel de aceptación del cliente.

La confección de los casos de prueba aplicados a la solución de software se basan en el método empleado para el diseño de casos de prueba del proyecto, el cual, por las características y complejidad del sistema, es un híbrido obtenido a partir de los métodos de Particiones de Equivalencia y Análisis de Valores Límites, el cual se basa principalmente en la especificación de Casos de Uso. Para ver más detalles ver el documento Estrategias de Pruebas del proyecto.

3.2.2 NIVELES DE PRUEBAS

Para lograr una mayor efectividad las pruebas de software se realizan a diferentes niveles: ⁽⁹⁾

1. Pruebas Unitarias. Comienzan con la prueba de cada módulo.
2. Pruebas de Integración. A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces.
3. Prueba del Sistema. El software es ensamblado totalmente con cualquier componente de hardware que requiere y se prueba para comprobar que se cumplen los requisitos funcionales.
4. Pruebas de Aceptación. El cliente comprueba que el software funciona según sus expectativas.

Se le realizaron varias pruebas al sistema correspondientes a los niveles descritos anteriormente. Los resultados más notables se obtuvieron durante las pruebas de Calidad Interna, Aceptación y Pruebas Pilotos.

3.3 RESULTADOS OBTENIDOS

En los siguientes gráficos se muestran los resultados obtenidos en las principales pruebas realizadas al software.

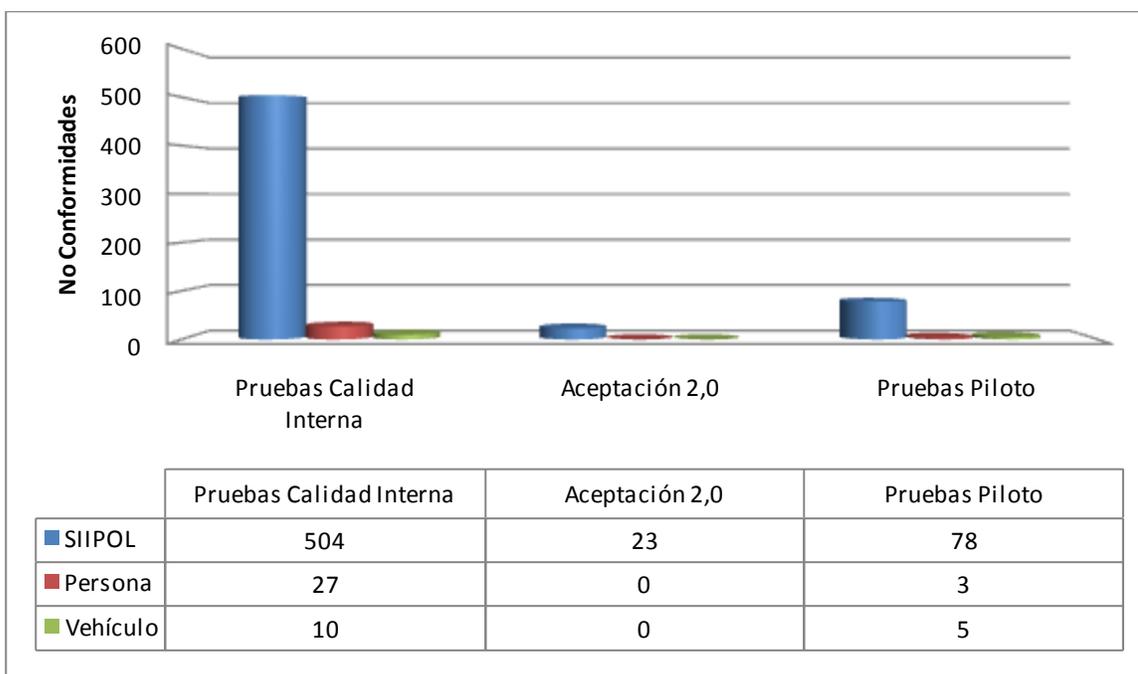


Fig. 3.1 Gráfico de Columnas de Pruebas Realizadas a la Aplicación.

Como se puede apreciar en el Gráfico 3.1, los Sub-Módulos Persona y Vehículo no presentan gran cantidad de problemas en las pruebas realizadas, han tenido muy buena aceptación por parte del cliente. En el Gráfico 3.2 se aprecia la estabilidad de estos sub-módulos al no tener muchas variaciones en la cantidad de errores, manteniéndose estos en cifras muy bajas.

CAPÍTULO 3. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA | ANÁLISIS DISEÑO E IMPLEMENTACIÓN DE LA CAPA DE PRESENTACIÓN DE LOS SUB-MÓDULOS PERSONA Y VEHÍCULO PERTENECIENTES AL SUBSISTEMA DE ANÁLISIS DE INFORMACIÓN DEL SIIPOL

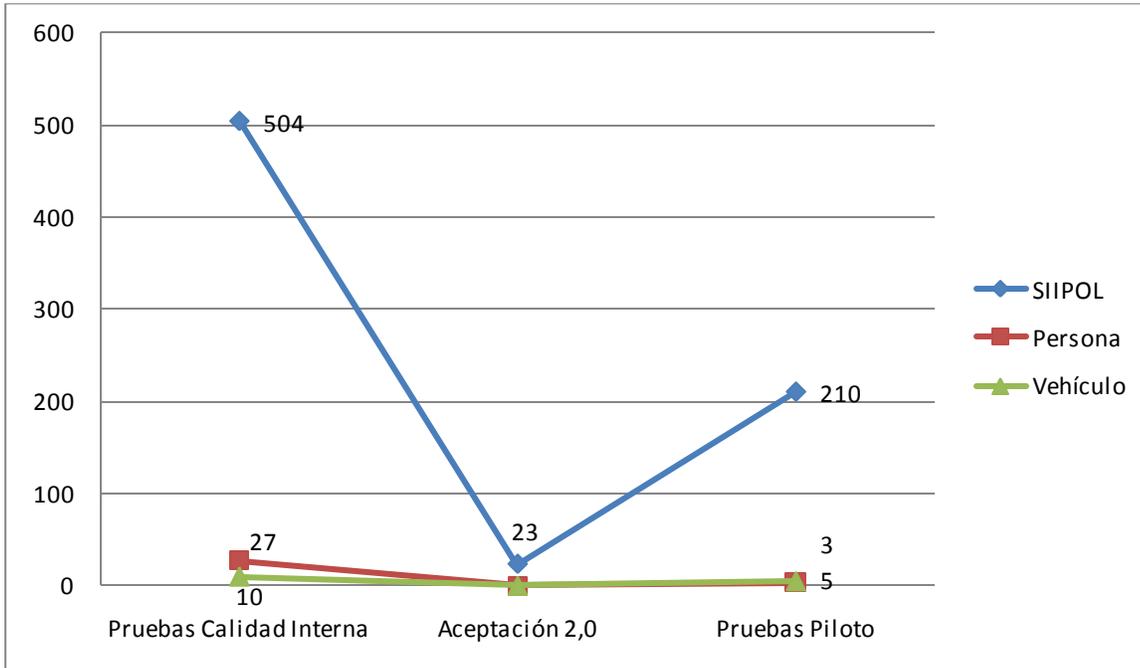


Fig. 3.2 Gráfico de Líneas de Pruebas Realizadas a la Aplicación.

3.4 CONCLUSIONES

A partir de los resultados obtenidos en las pruebas realizadas se demuestra la calidad del trabajo realizado en la implementación de los Sub-Módulos Persona y Vehículo, además de cumplir con los requisitos funcionales y no funcionales y tener una buena aceptación por parte del cliente. Los errores detectados a lo largo del proceso de pruebas fueron resueltos rápidamente a medida que estas se realizaban.

CONCLUSIONES

La presente investigación recoge todo el proceso de desarrollo de los Sub-Módulos Persona y Vehículo pertenecientes al Subsistema de Análisis de Información del SIIPOL. Para desarrollar el proyecto se elaboró un marco teórico conceptual que sirvió como base para todo el ciclo de desarrollo. Con el uso de tecnologías que presentan una buena integración, se analizó y diseñó una aplicación robusta obteniéndose además todos los artefactos necesarios dentro del ciclo de desarrollo del software, en todo momento utilizándose como guía rectora el Proceso Unificado de Desarrollo. Se realizó la implementación, respetando en todo momento el uso de los patrones seleccionados y los estándares de codificación. Se cumplió con el Cronograma de Tareas terminando en tiempo todas las metas planificadas. Posterior a la implementación, se realizaron varias iteraciones de pruebas para la validación de la propuesta de solución, obteniéndose buenos resultados en las mismas.

El resultado de este trabajo es una fracción del SIIPOL que cumple con los requerimientos funcionales y no funcionales relacionados a los Sub-Módulos Persona y Vehículo pertenecientes al Subsistema de Análisis de Información desde el punto de vista de Lógica de Presentación.

RECOMENDACIONES

- Continuar con la realización de las Pruebas de Calidad al software, para entregar una aplicación más robusta y estable.
- Realizar un proceso de refactorización de la capa de lógica de presentación de los Sub-Módulos Persona y Vehículo.
- Estudiar de forma más detallada la División de Información Policial en busca de otros procesos que puedan ser automatizados.

REFERENCIAS BIBLIOGRÁFICAS

1. **Gabaldón, Luis Gerardo.** Seguridad ciudadana, confianza pública y policía en Venezuela. *Revista Venezolana de Economía y Ciencias Sociales*. [En línea] Diciembre de 2008. http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1315-64112007000300006&lng=es&nrm=iso.
2. La Herramienta del Futuro Para la Policia. *La Herramienta del Futuro Para la Policia*. [En línea] Enero de 2009. <http://www.datapolice.org>.
3. Mapas Digitales S.A. *Dmapas*. [En línea] 21 de Enero de 2009. [Citado el: 21 de Enero de 2009.] http://dmapas.cl/productos_stegpol.htm.
4. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.
5. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : Addison-Wesley.
6. Why Visual Paradigm for UML? . *UML CASE Tools - Free for Learning UML, Cost-Effective for Business Solutions*. [En línea] Enero de 2009. <http://www.visual-paradigm.com/product/vpuml/index.jsp>.
7. Original Java Whitepaper. *Developer Resources for Java Technology*. [En línea] Enero de 2009. <http://java.sun.com/people/jag/OriginalJavaWhitepaper.pdf>.
8. **D. Mann, Kito.** *JavaServer Faces in Action*. s.l. : Manning Publications Co., 2005. ISBN 1-932394-11-7.
9. Evaluación de Software. <http://teleformacion.uci.cu>. [En línea] [Citado el: 10 de marzo de 2009.] <http://teleformacion.uci.cu/mod/resource/view.php?id=14103>.

BIBLIOGRAFÍA

1. **Rumbaugh, James, Jacobson, Ivar and Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de Referencia.* Madrid : Addison-Wesley.
2. **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos.* Mexico : Prentice Hall, 1999.
3. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000.
4. **Gabaldón, Luis Gerardo.** Seguridad ciudadana, confianza pública y policía en Venezuela. *Revista Venezolana de Economía y Ciencias Sociales.* [Online] Diciembre 2008.
http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1315-64112007000300006&lng=es&nrm=iso.
5. **D. Mann, Kito.** *JavaServer Faces in Action.* s.l. : Manning Publications Co., 2005. ISBN 1-932394-11-7.
6. Why Visual Paradigm for UML? . *UML CASE Tools - Free for Learning UML, Cost-Effective for Business Solutions.* [Online] Enero 2009. <http://www.visual-paradigm.com/product/vpuml/index.jsp>.
7. Tema 1: Introducción al proceso de desarrollo de software. *Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas.* [Online] Enero 2009.
http://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Basicos/Materiales_Basicos_Conf_1/Clase_1_Curso_Proceso_de_Desarrollo_de_Software.pdf.
8. Programación 3. Conferencia 10. DHTML y JavaScript. *Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas.* [Online] Enero 2009.
http://teleformacion.uci.cu/file.php/266/Actividades_Tema_3/Conferencia_10_Dhtml_y_JavaScriptl_.pdf.
9. Overview. SUN Microsystems. *Java 2 Platform, Enterprise Edition(J2EE).* [Online] Enero 2009.
<http://java.sun.com/j2ee/overview.html>.
10. Original Java Whitepaper. *Developer Resources for Java Technology.* [Online] Enero 2009.
<http://java.sun.com/people/jag/OriginalJavaWhitepaper.pdf>.
11. Mapas Digitales S.A. *Dmapas.* [En línea] 21 de Enero de 2009. [Citado el: 21 de Enero de 2009.]
http://dmapas.cl/productos_stegpol.htm.
12. La Herramienta del Futuro Para la Policía. *La Herramienta del Futuro Para la Policía.* [Online] Enero 2009. <http://www.datapolice.org>.

13. Evaluación de Software. *http://teleformacion.uci.cu*. [Online] [Cited: marzo 10, 2009.]
<http://teleformacion.uci.cu/mod/resource/view.php?id=14103>.
14. Apache MyFaces. *The Apache Software Foundation*. [Online] 2009.
<http://projects.apache.org/projects/myfaces.html>.
15. *Plataforma Eclipse*. [Online] 2009. <http://plataformaclipse.com/>.
16. **Larman, Craig**. *UML y Patrones*. Mexico : Prentice Hall, 1999. ISBN.
17. **Pressman, Roger S**. *Ingeniería de Software. Un enfoque práctico*. s.l. : McGraw Hill.

GLOSARIO DE TÉRMINOS

Adabas-Natural: Lenguaje de Programación en que está implementado el sistema anterior.

Código Fuente: Es un conjunto de líneas de texto que son las instrucciones que debe seguir el ordenador para ejecutar dicho programa.

Comunicación: Oficio o Memorando básico. Forma fundamental y oficial de comunicación dentro del CICPC (entre los diferentes despachos), en el caso del Memorando, y entre el CICPC y entes externos, en el caso del Oficio. Ambos tipos de Comunicación coinciden en sus características fundamentales.

JavaBeans: Componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción. Un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones en Java.

Nomenclador: Catálogo o lista de nombres.

Open Source (Código Abierto): Software distribuido y desarrollado libremente.

Requerimientos: Un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Se usa en un sentido formal en la ingeniería de sistemas o la ingeniería de software. Estos pueden ser clasificados en dos formas:

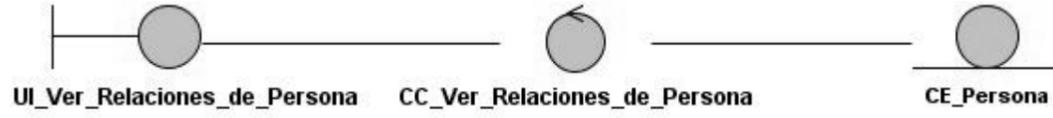
- **Requerimiento funcional:** Puede ser una descripción de lo que un sistema debe hacer. Este tipo de requerimiento especifica que es lo que debe de hacer el sistema.
- **Requerimiento no funcional:** Especifica algo sobre el propio sistema y cómo debe realizar sus funciones. Estos pueden ser clasificados como requerimientos no funcionales: de rendimiento, de calidad, de seguridad, entre otros.

Software: Es el equipamiento o soporte lógico de una computadora. Comprende un conjunto de componentes lógicos necesarios para hacer posible la realización de una tarea específica.

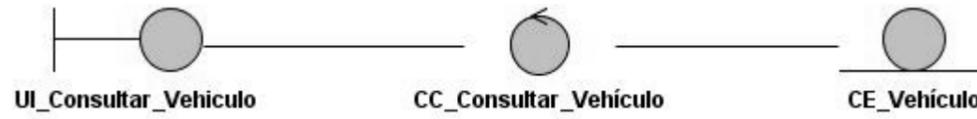
“write once, run everywhere”: Es la característica más sobresaliente de JAVA frente a otros lenguajes de programación por su operación independiente de la plataforma, con lo cual una aplicación se compila una sola vez, y el código de bytes de JAVA resultante (archivos .class) puede ser interpretado en cualquier sistema operativo, que tenga una máquina virtual instalada.

ANEXOS

ANEXO 1: DIAGRAMAS DE CLASES DEL ANÁLISIS DE LOS CU MÁS SIGNIFICATIVOS.

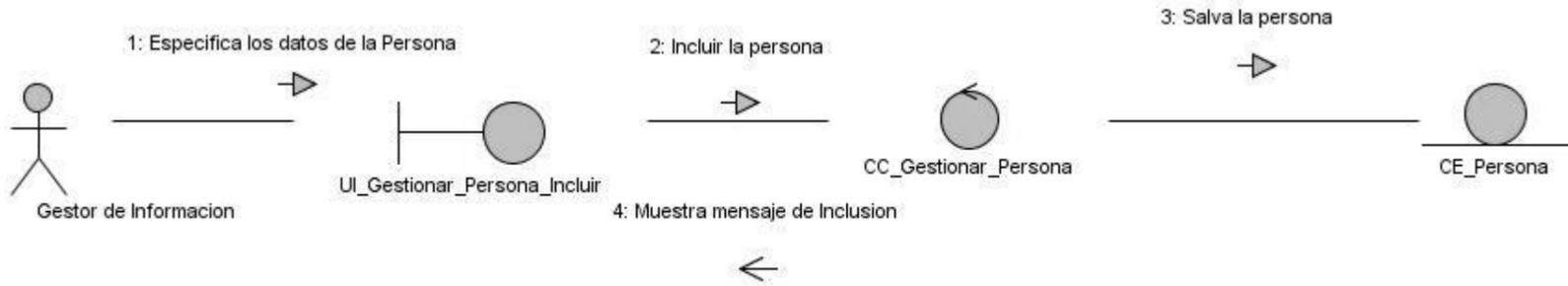


Anexo 1.1 Diagrama de Clases del Análisis perteneciente al CU Ver Relaciones de Persona.

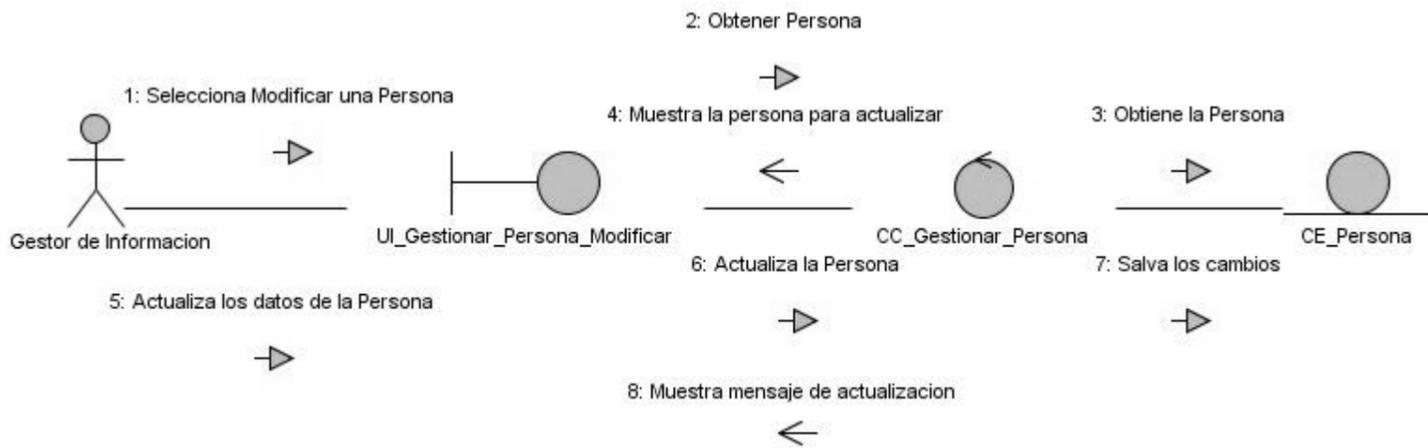


Anexo 1.2 Diagrama de Clases del Análisis perteneciente al CU Consultar Vehículo.

ANEXO 2: REALIZACIÓN DE CASOS DE USO MÁS SIGNIFICATIVOS DEL ANÁLISIS.



Anexo 2.1 Diagrama de Comunicación perteneciente al CU Gestionar Persona: Incluir.



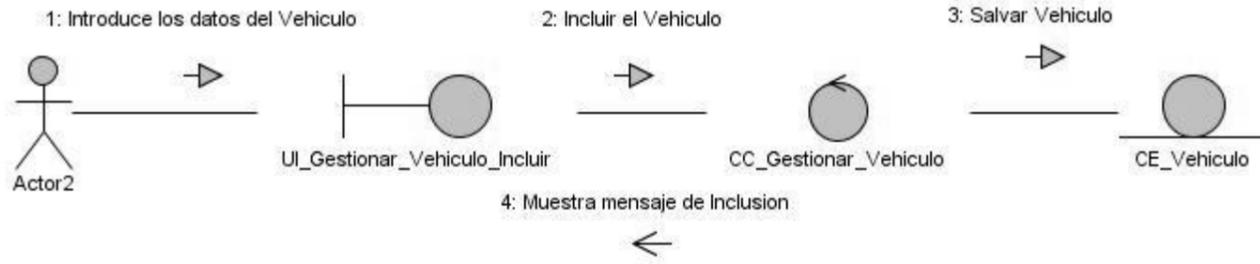
Anexo 2.2 Diagrama de Comunicación perteneciente al CU Gestionar Persona: Modificar.



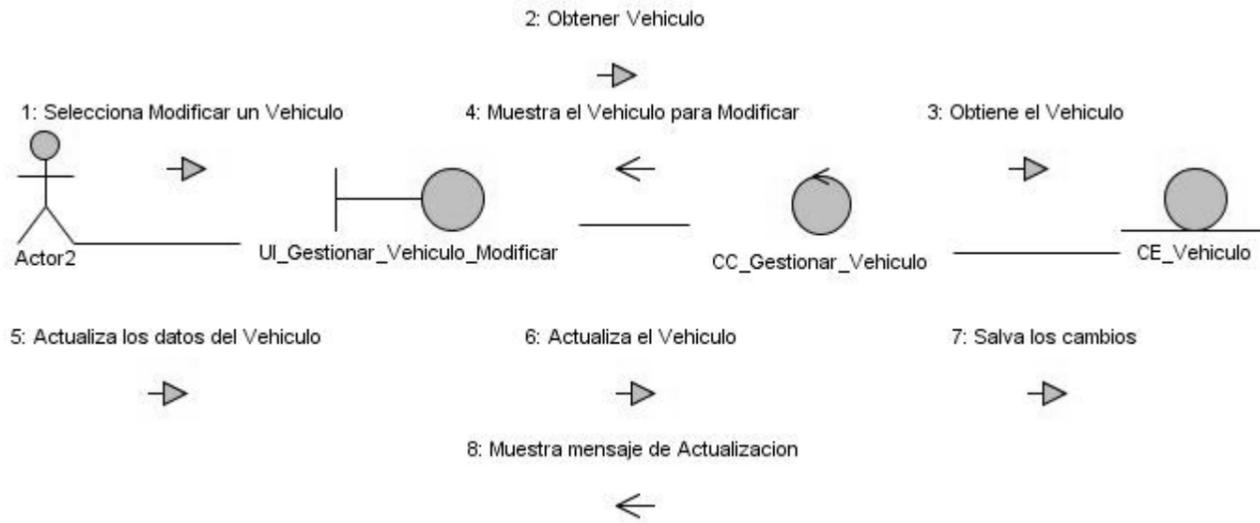
Anexo 2.3 Diagrama de Comunicación perteneciente al CU Gestionar Persona: Ver.



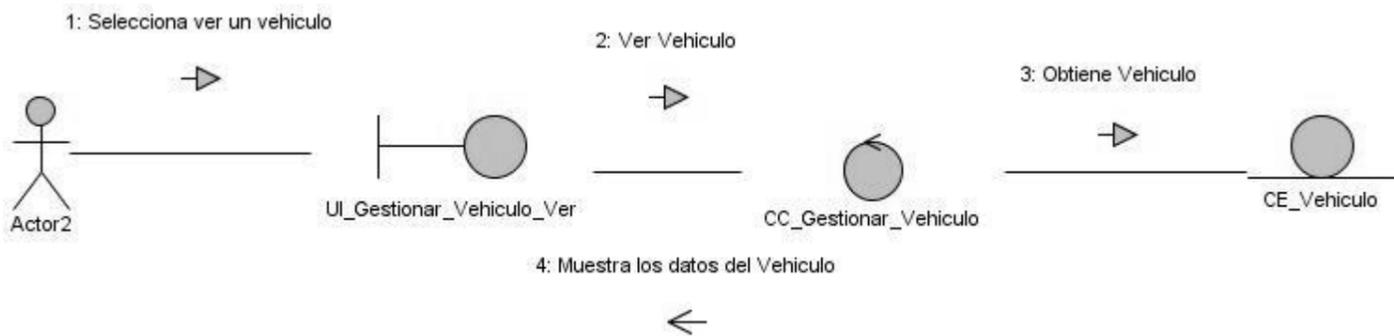
Anexo 2.4 Diagrama de Comunicación perteneciente al CU Consultar Persona.



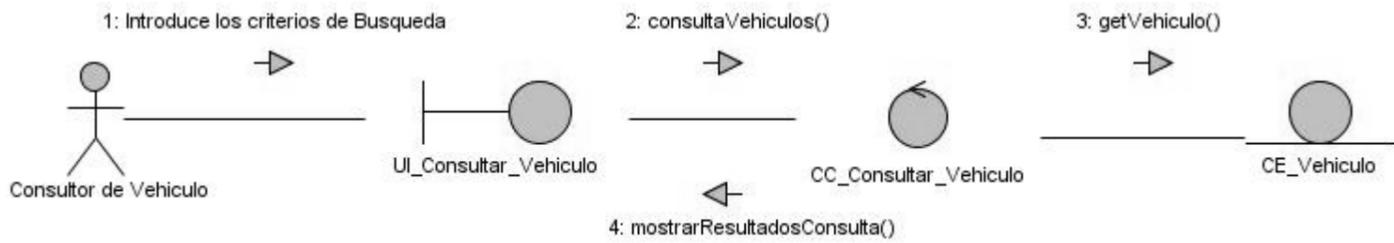
Anexo 2.5 Diagrama de Comunicación perteneciente al CU Gestionar Vehículo: Incluir.



Anexo 2.6 Diagrama de Comunicación perteneciente al CU Gestionar Vehículo: Modificar.

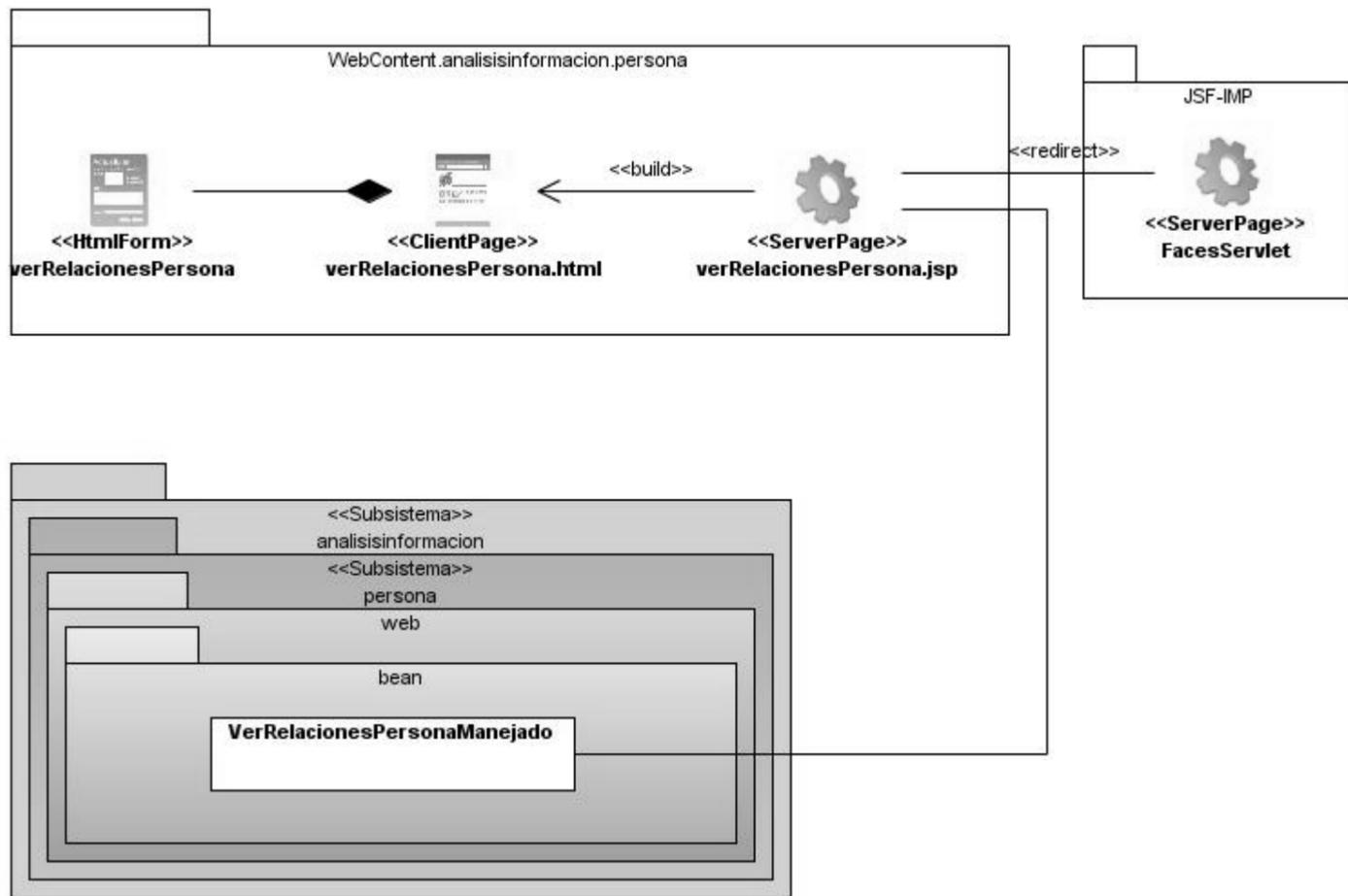


Anexo 2.7 Diagrama de Comunicación perteneciente al CU Gestionar Vehículo: Ver.

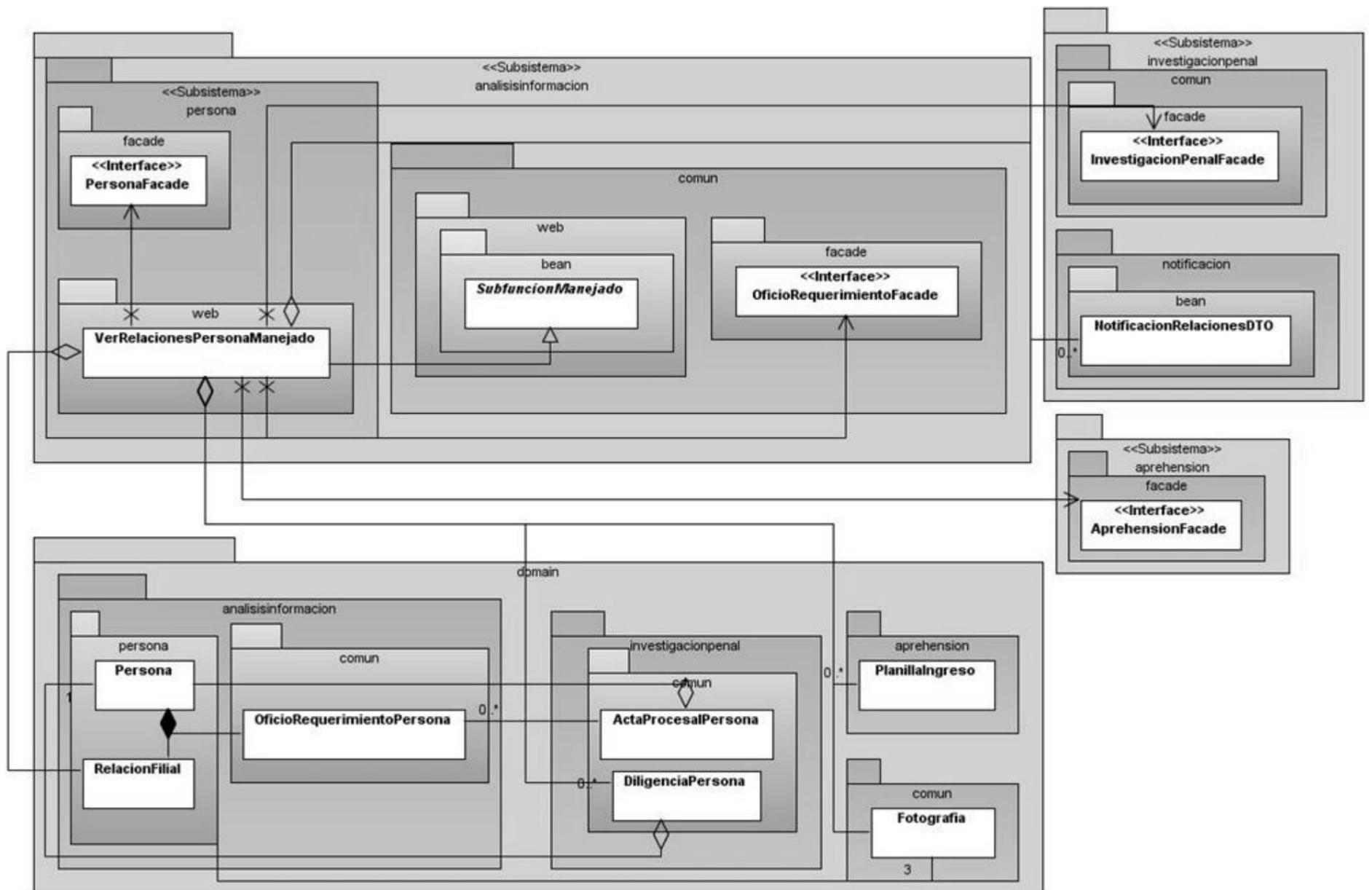


Anexo 2.8 Diagrama de Comunicación perteneciente al CU Consultar Vehículo.

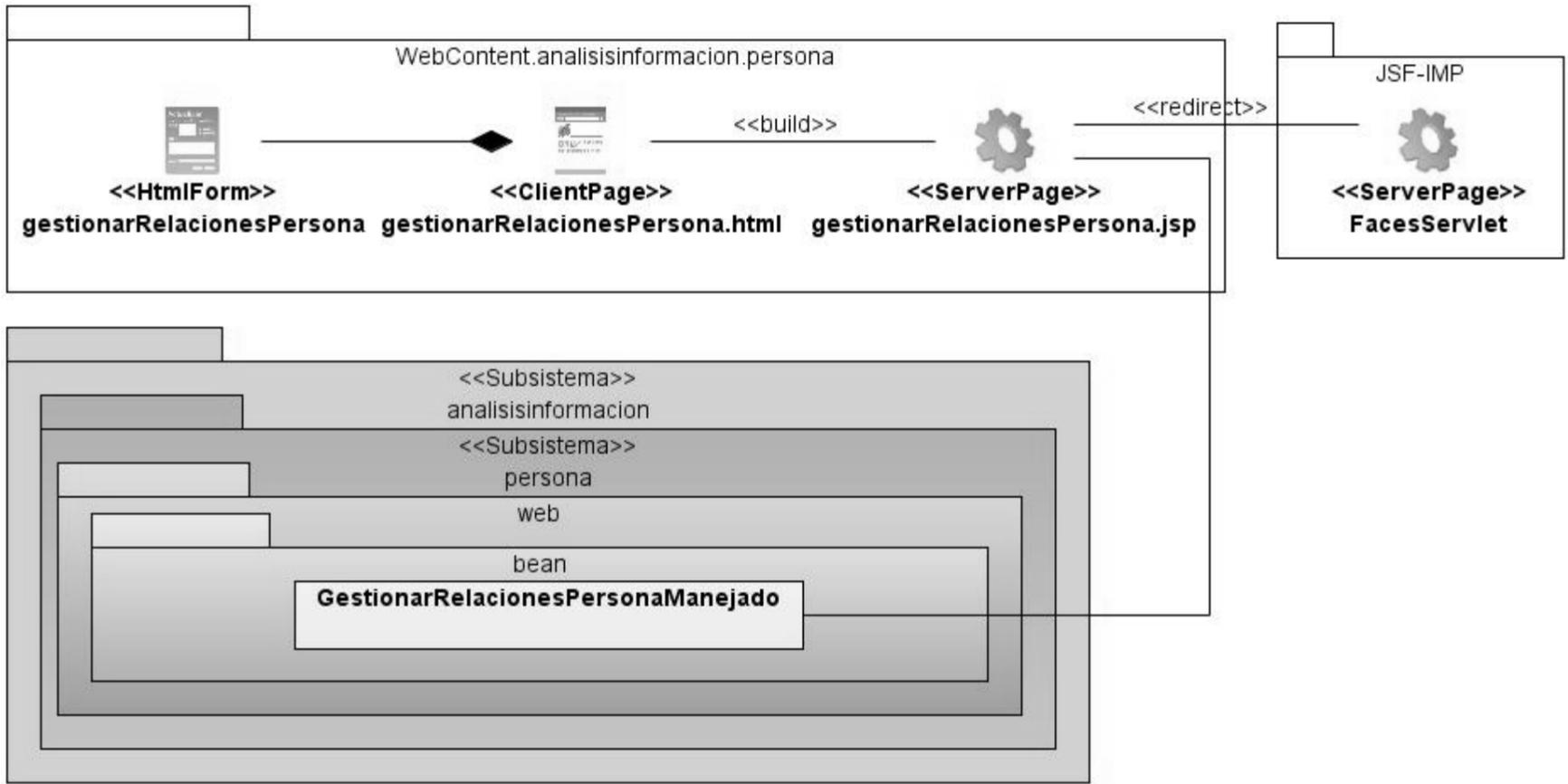
ANEXO 3: DIAGRAMAS DE CLASES DEL DISEÑO.



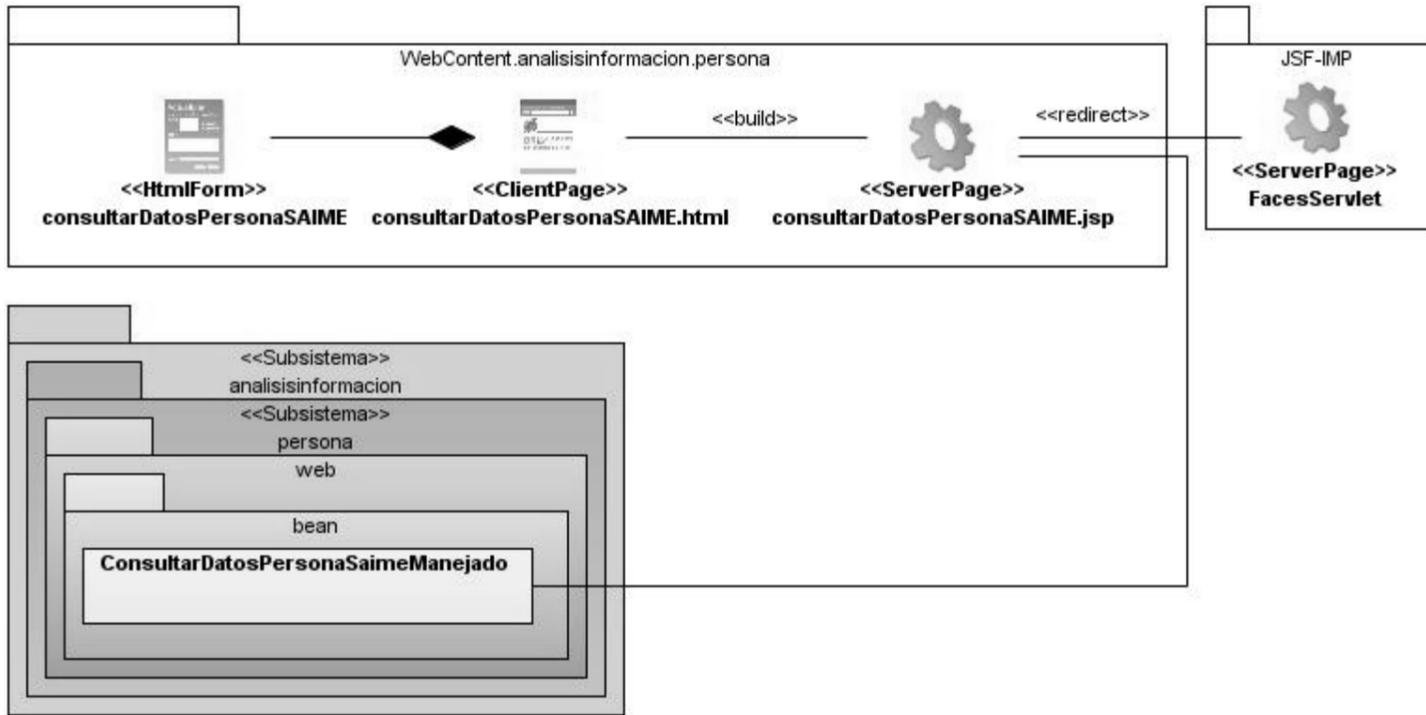
Anexo 3.1 Diagrama de Clases del Diseño Web pertenecientes al CU Ver Relaciones de Persona.



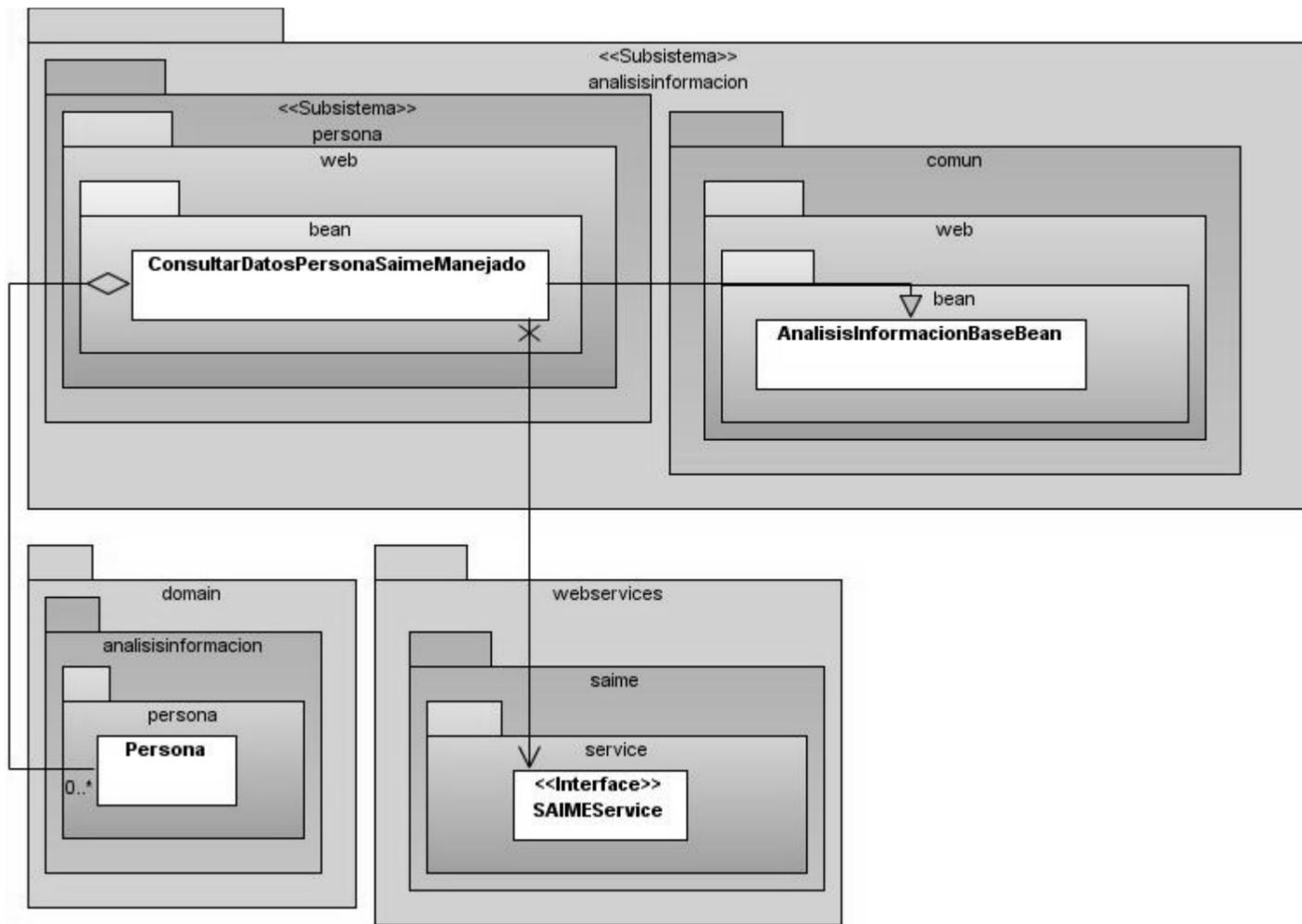
Anexo 3. 2 Diagrama de Clases del Diseño pertenecientes al CU Ver Relaciones de Persona.



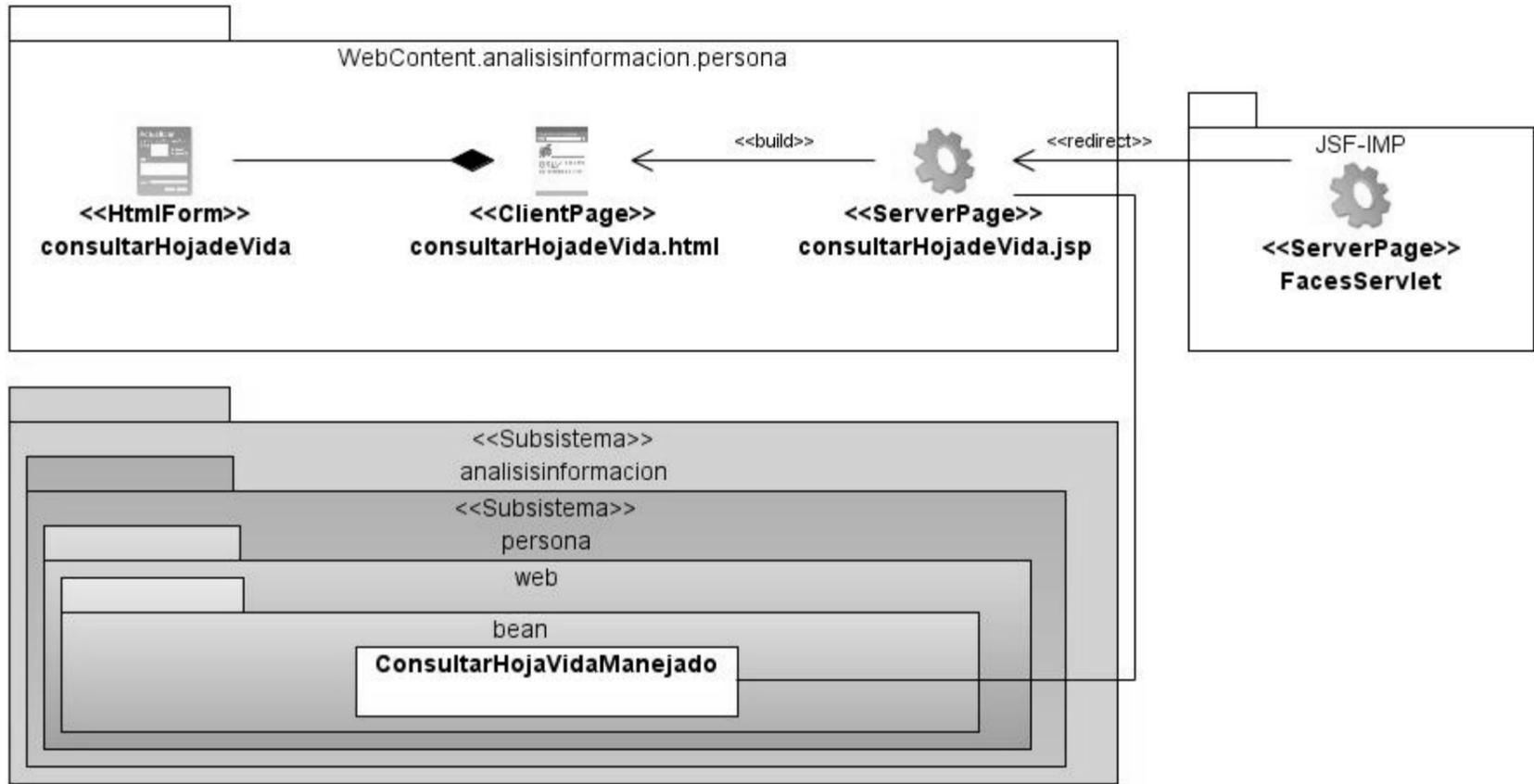
Anexo 3.3 Diagrama de Clase del Diseño Web pertenecientes al CU Gestionar Relaciones de Persona.



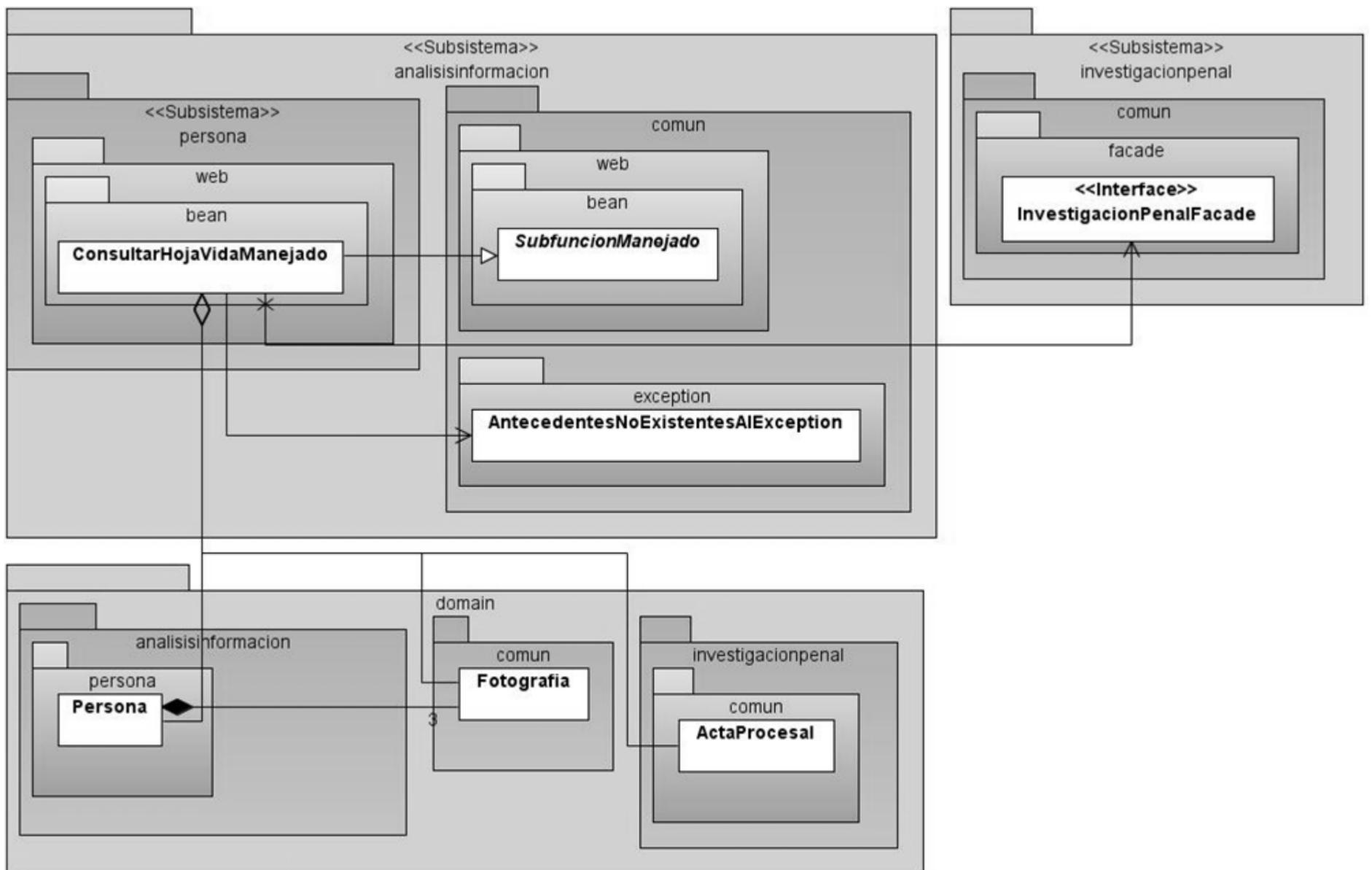
Anexo 3.5 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Datos de Persona SAIME.



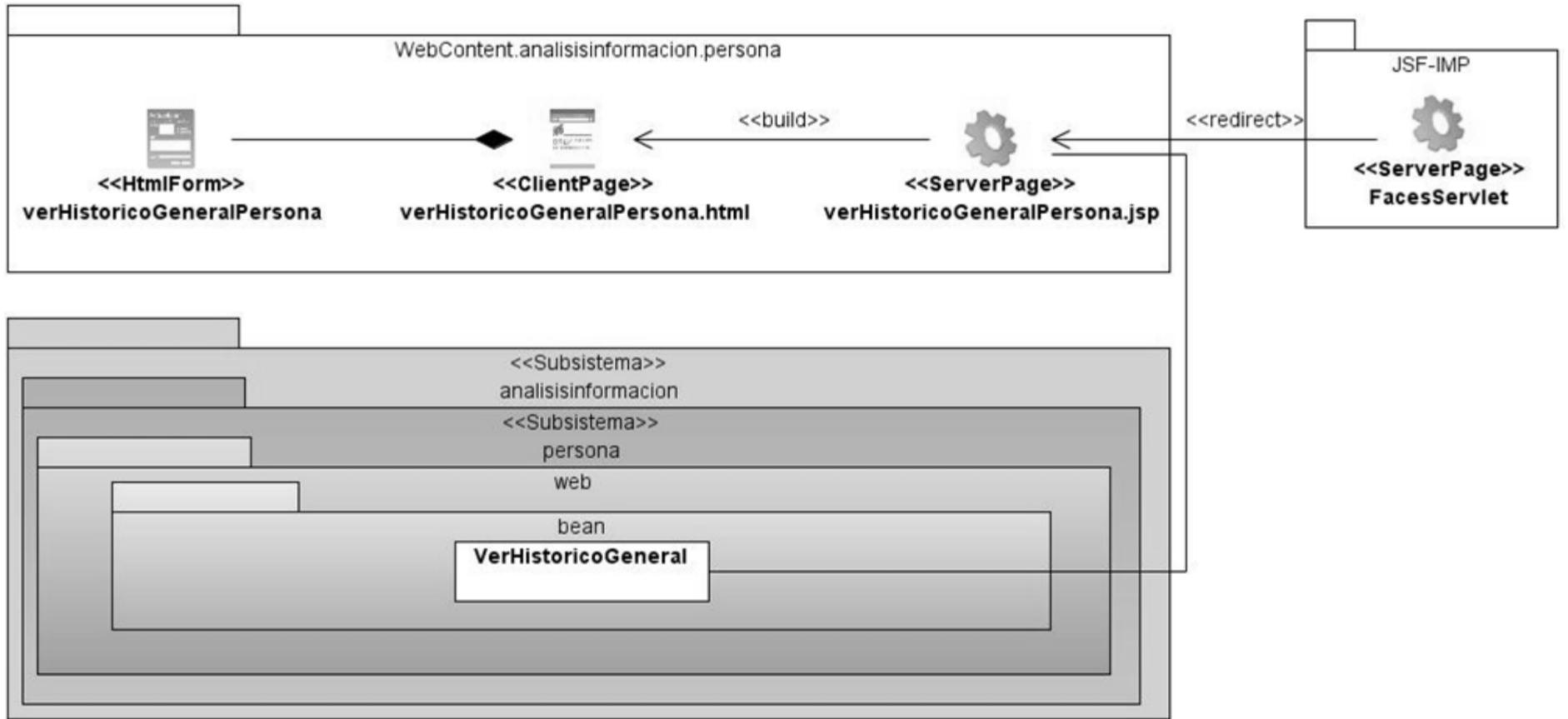
Anexo 3.6 Diagrama de Clases del Diseño pertenecientes al CU Consultar Datos de Persona SAIME.



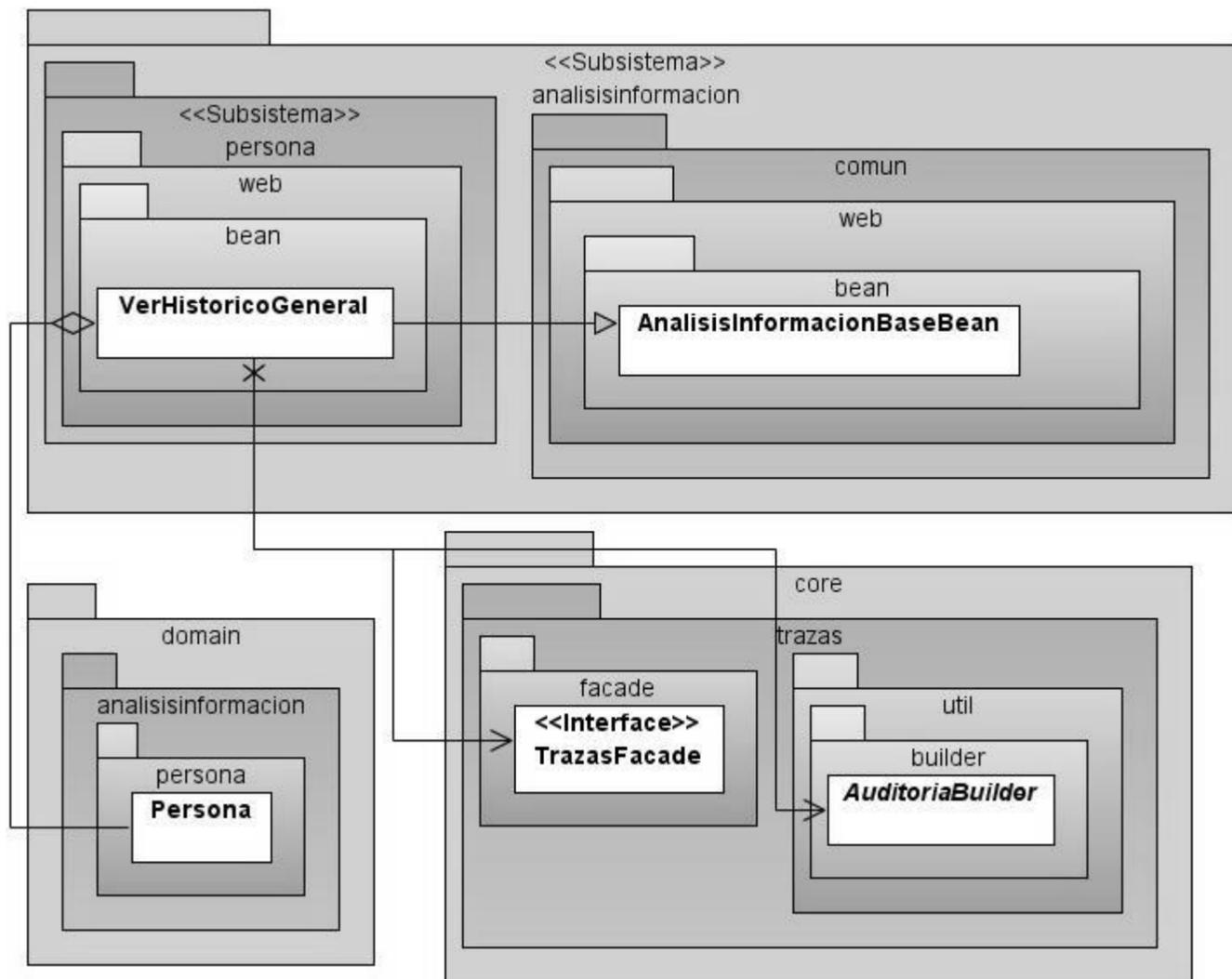
Anexo 3.7 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Hoja de Vida de Persona.



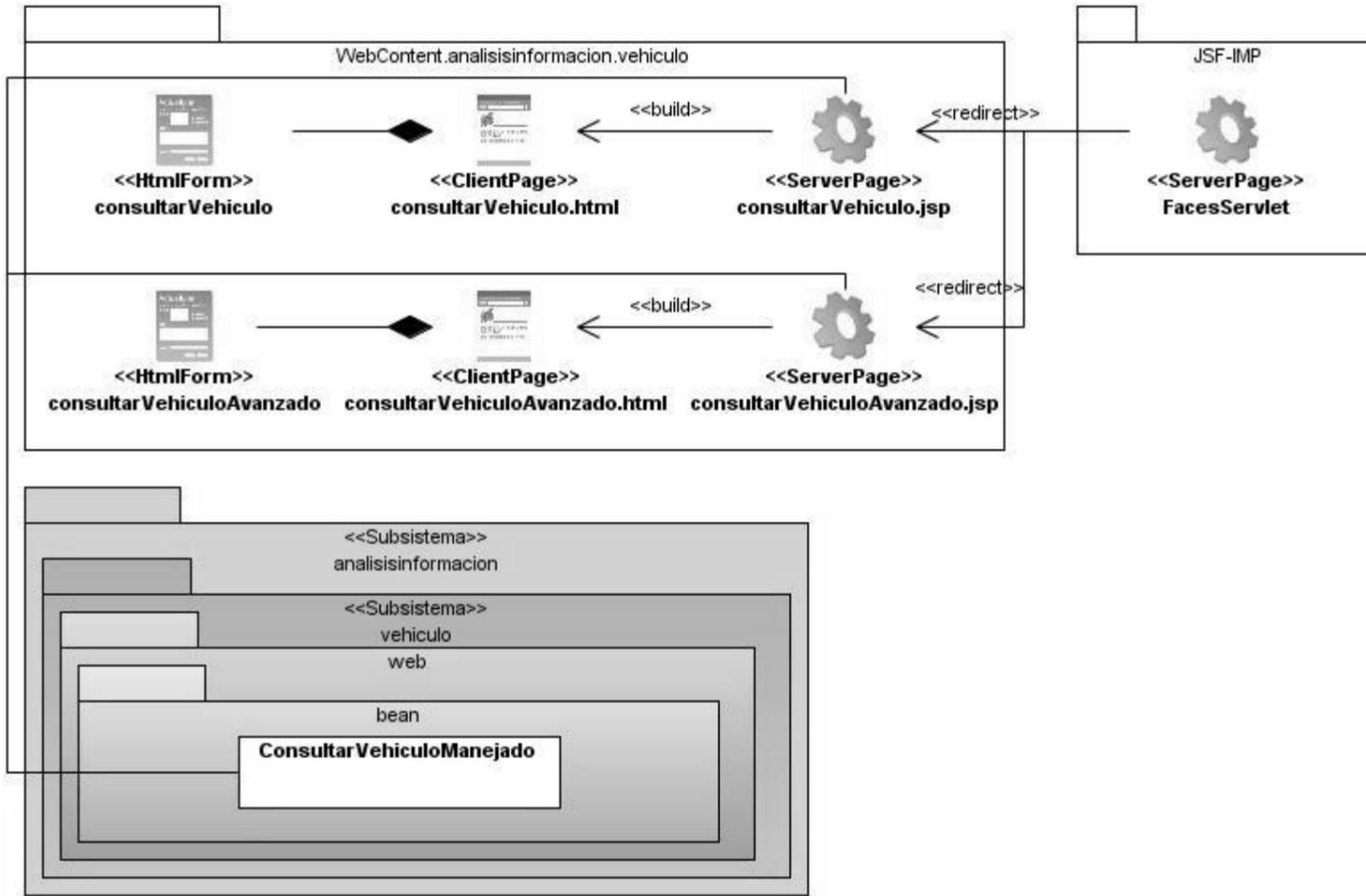
Anexo 3.8 Diagrama de Clases del Diseño pertenecientes al CU Consultar Datos de Persona SAIME.



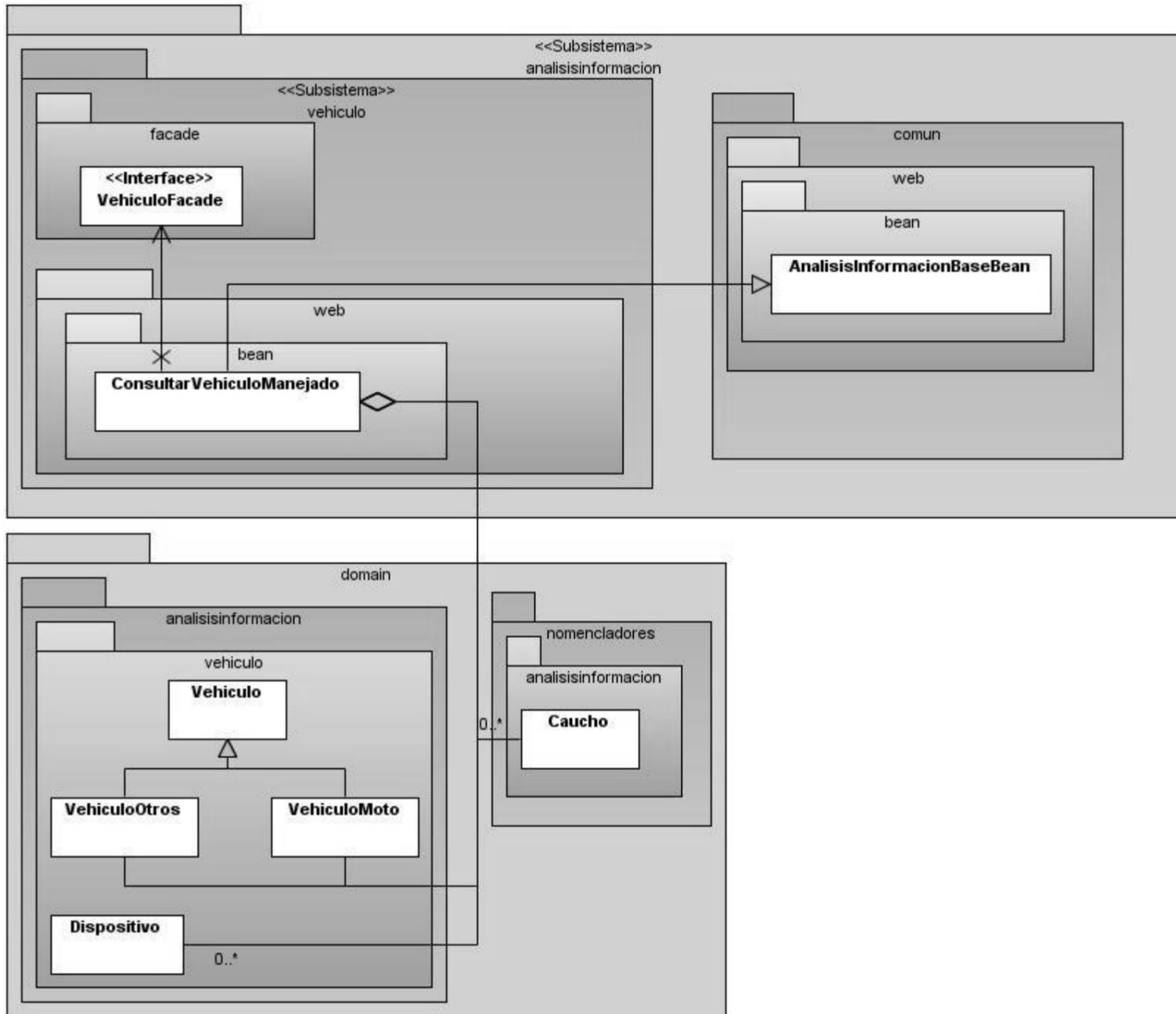
Anexo 3.9 Diagrama de Clases del Diseño Web pertenecientes al CU Ver Histórico General de Persona.



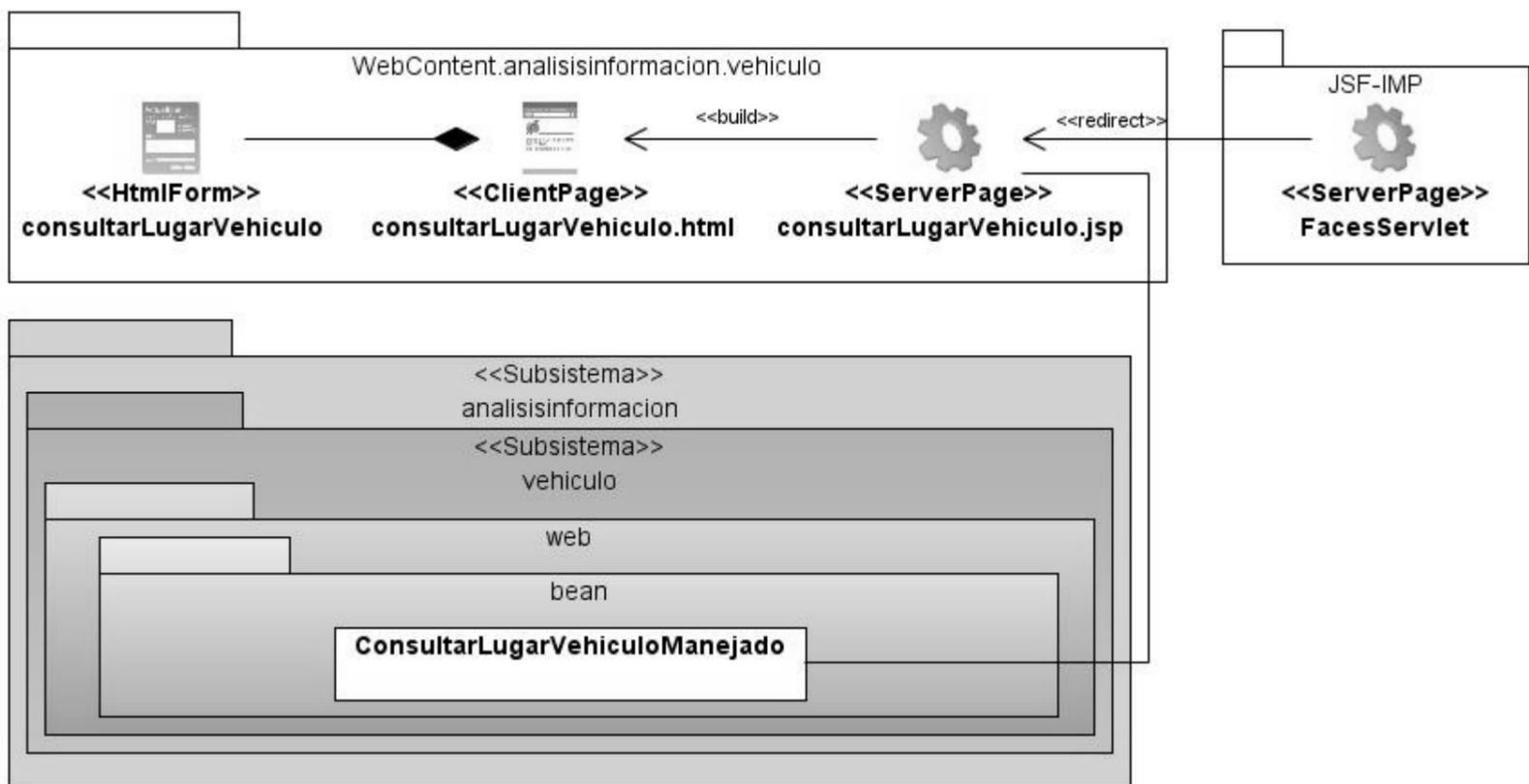
Anexo 3.10 Diagrama de Clases del Diseño pertenecientes al CU Ver Histórico General de Persona.



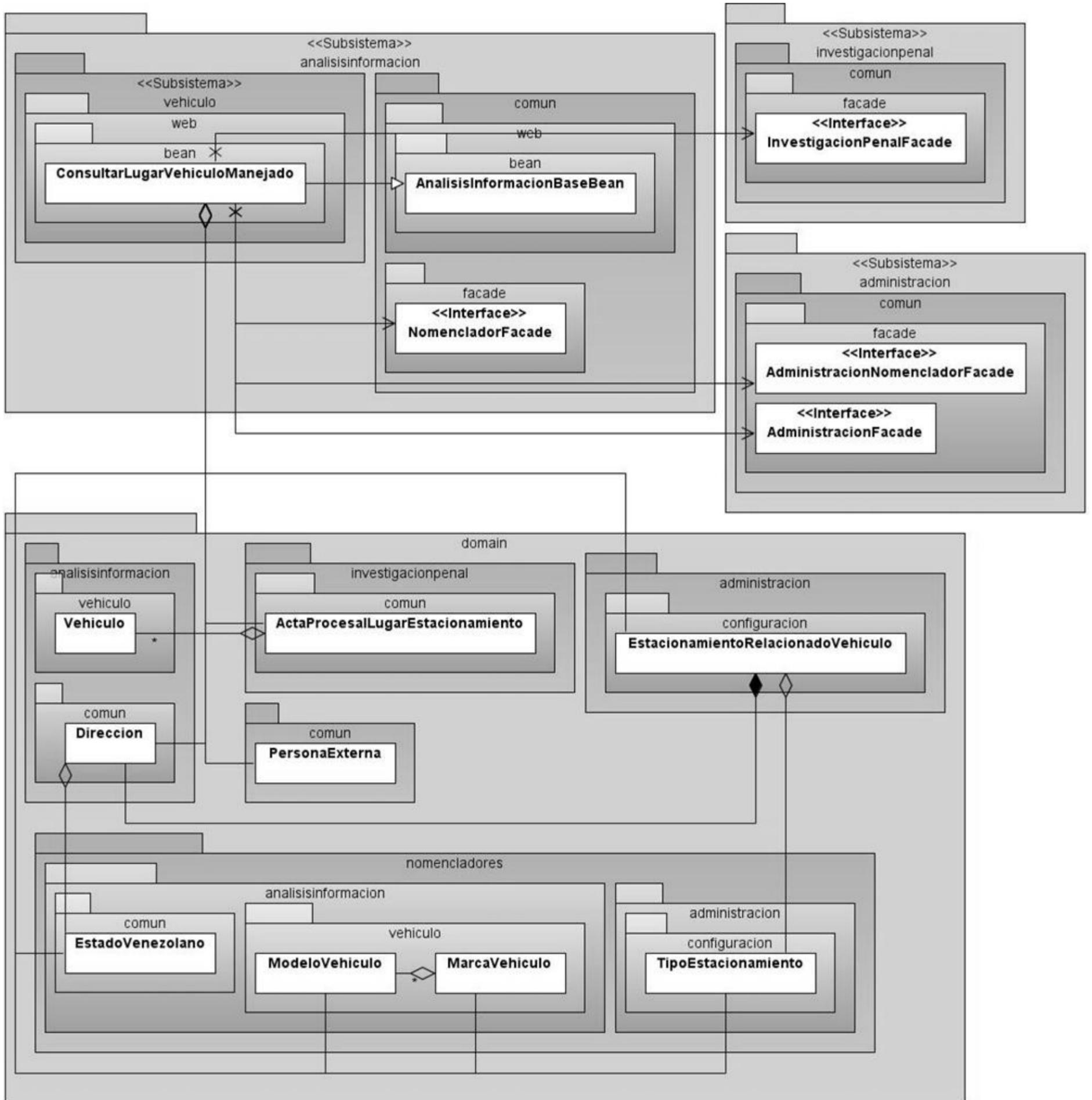
Anexo 3.11 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Vehículo Manejado.



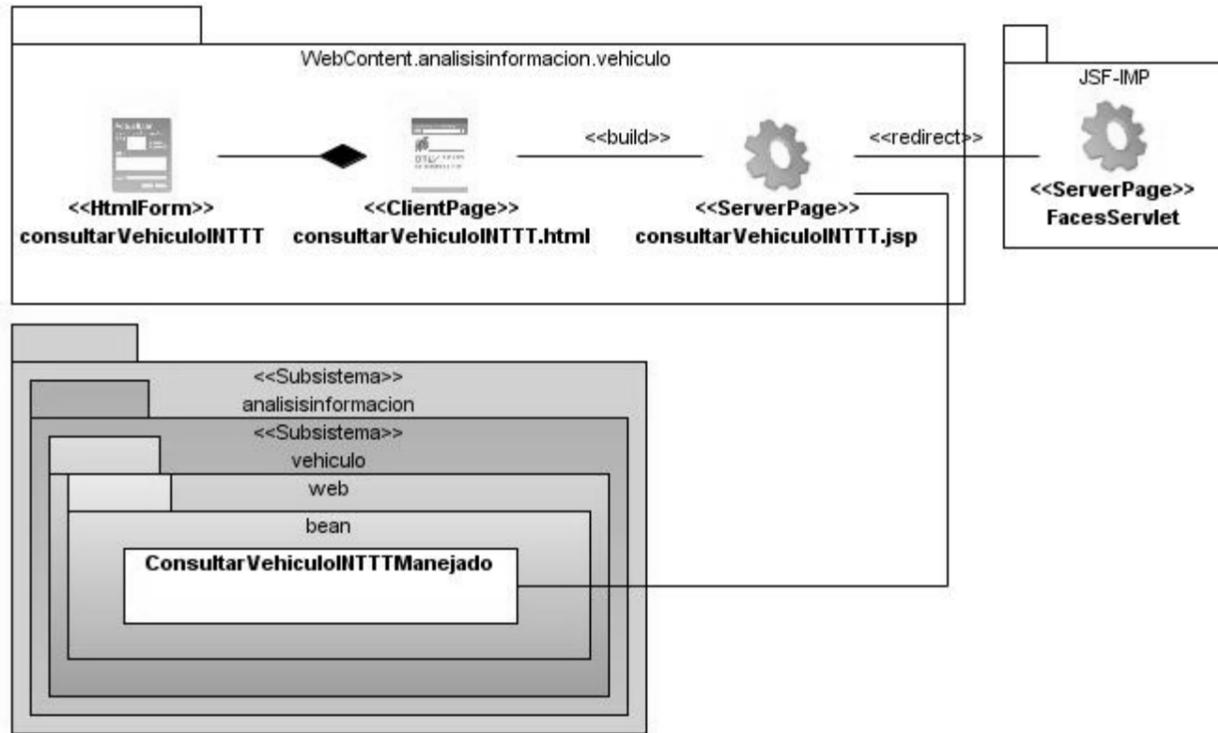
Anexo 3.12 Diagrama de Clases del Diseño pertenecientes al CU Consultar Vehículo.



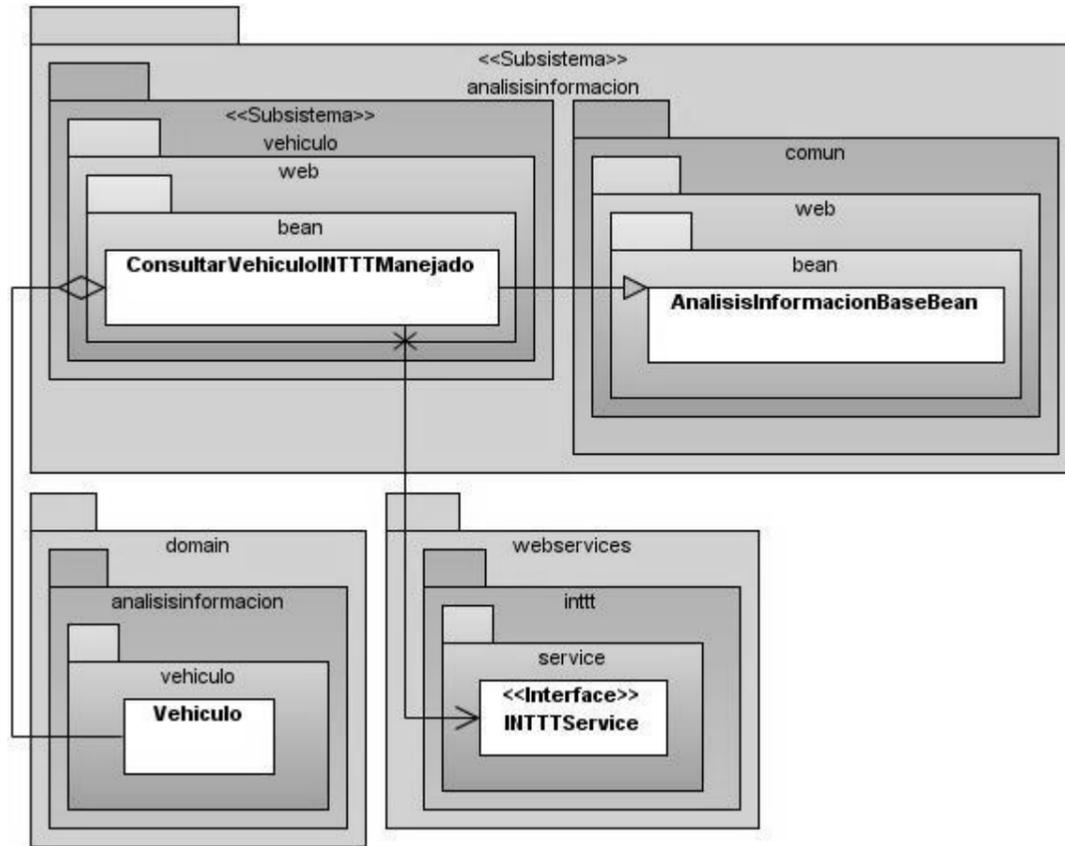
Anexo 3.13 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Lugares Relacionados a Vehículo.



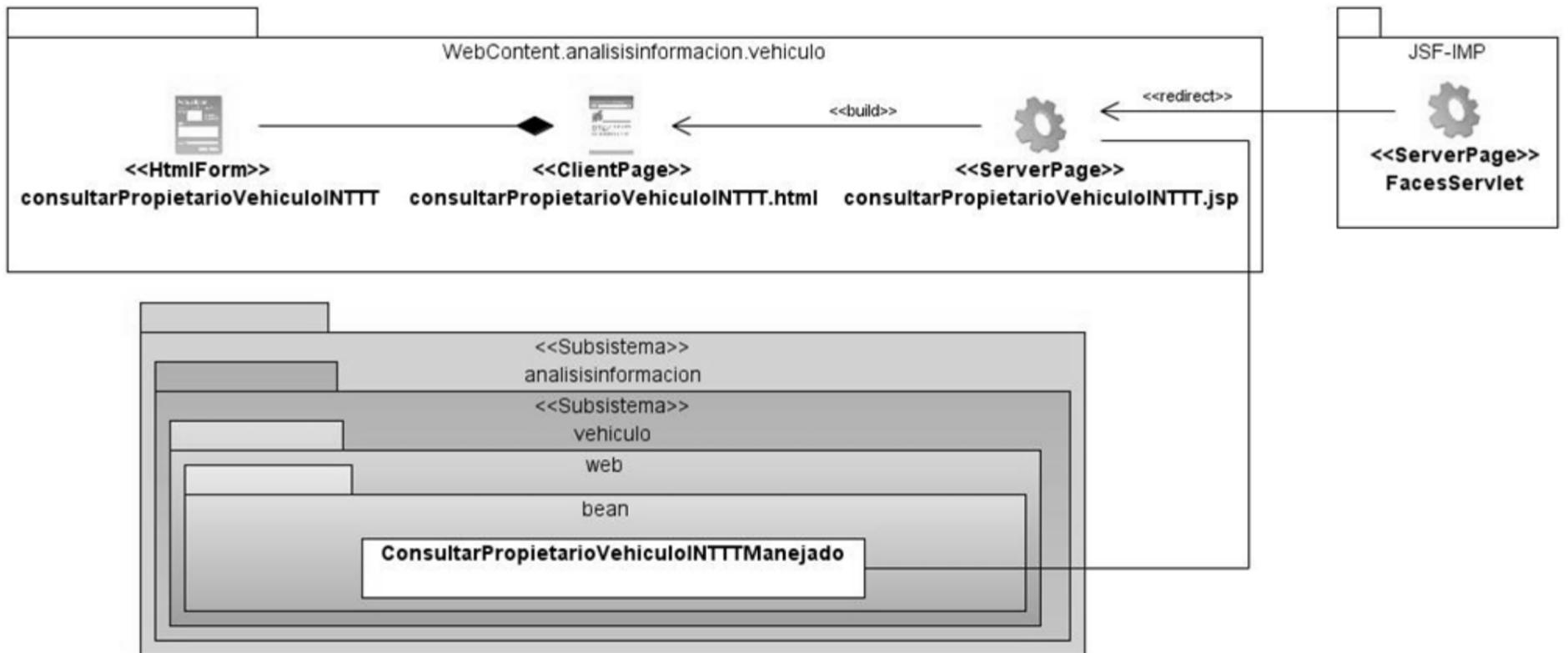
Anexo 3.14 Diagrama de Clases del Diseño pertenecientes al CU Consultar Lugares Relacionados a Vehículo.



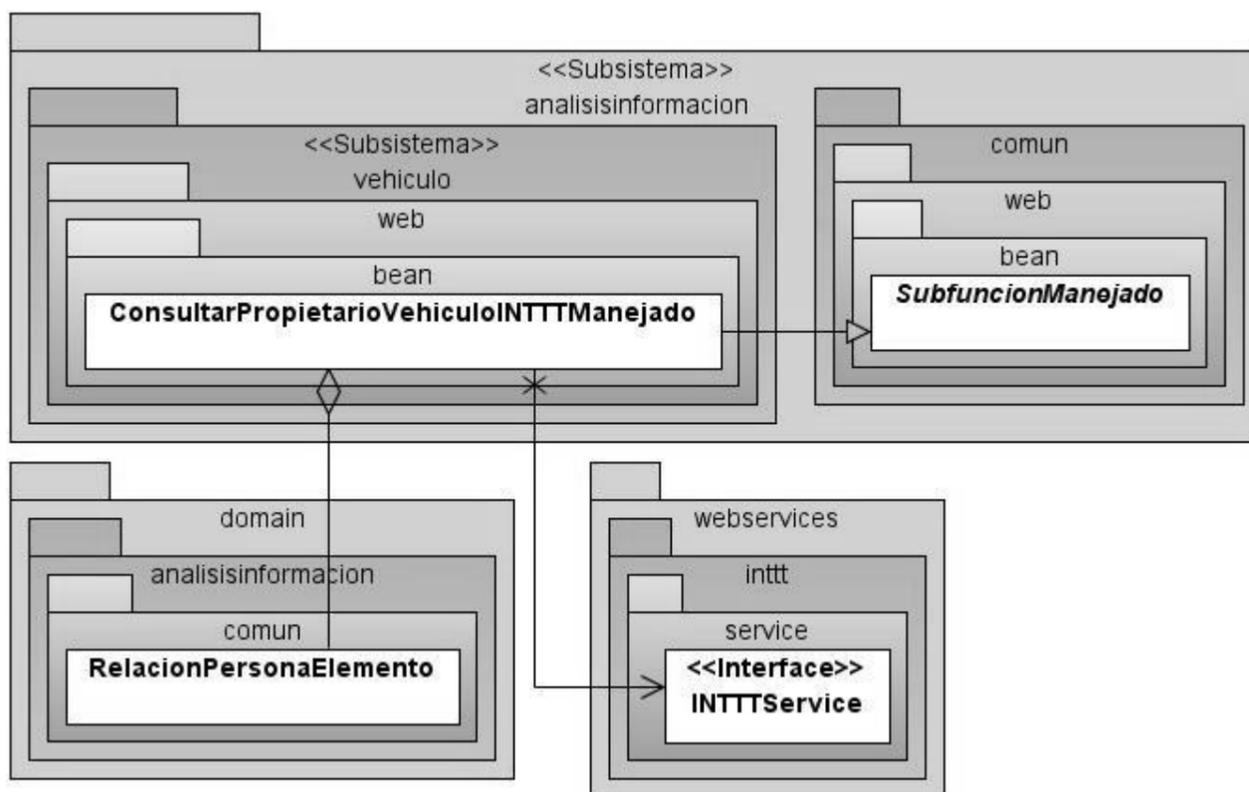
Anexo 3.15 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Datos de Vehículo INTTT.



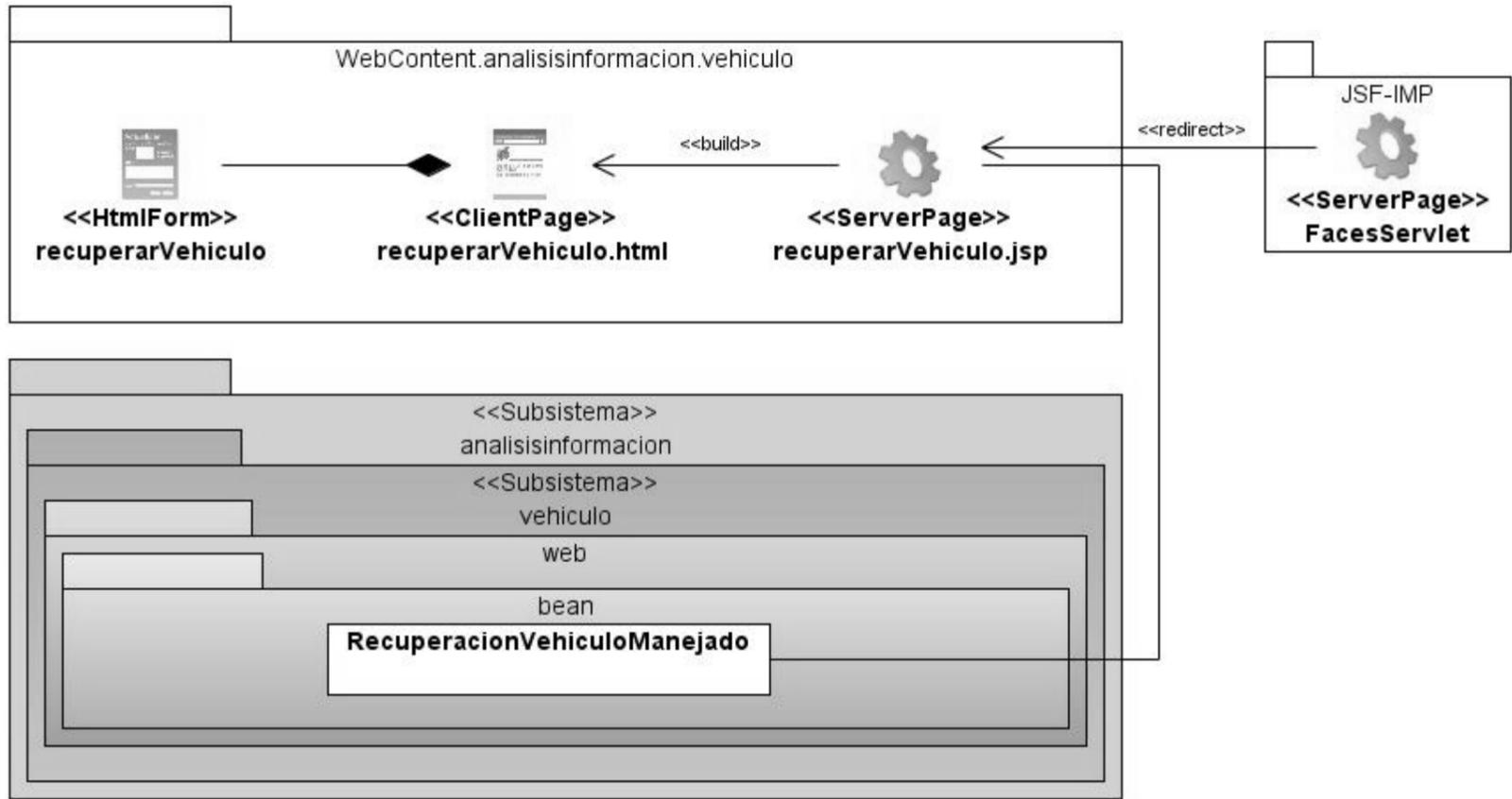
Anexo 3.16 Diagrama de Clases del Diseño pertenecientes al CU Consultar Datos de Vehículo INTTT.



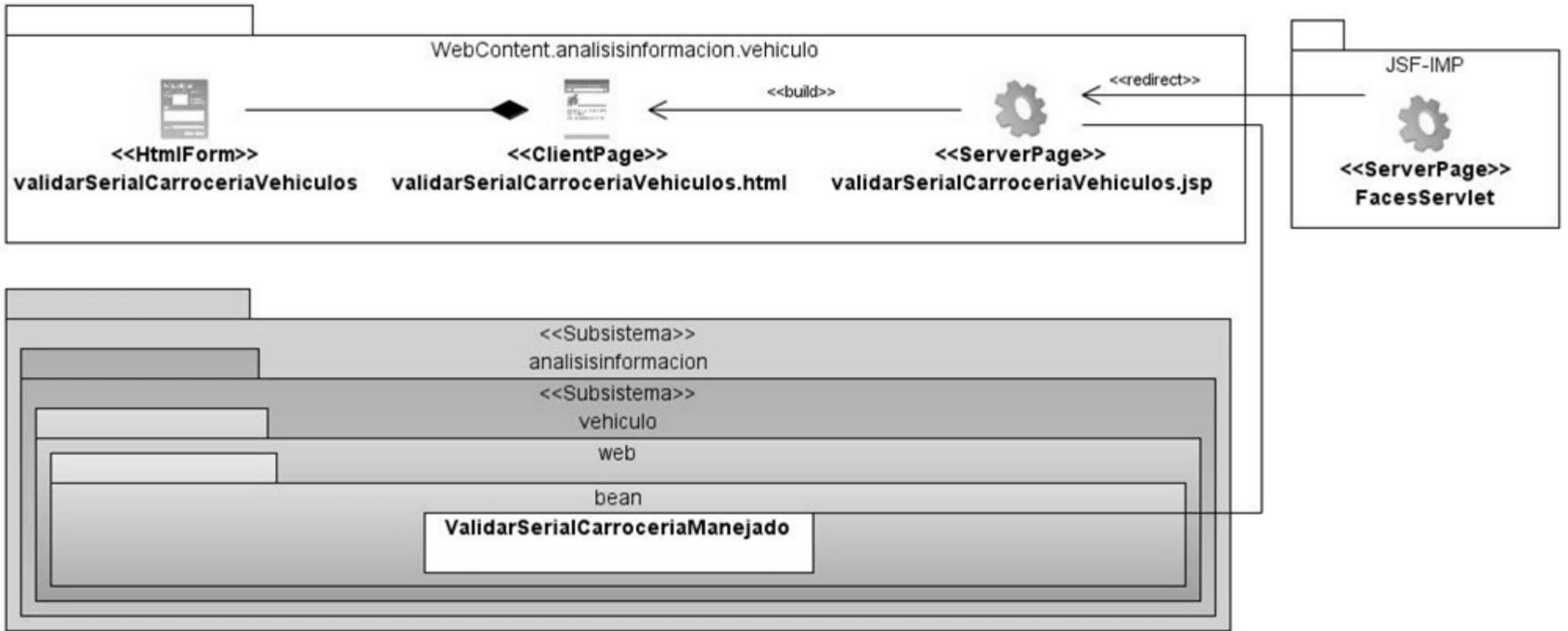
Anexo 3.17 Diagrama de Clases del Diseño Web pertenecientes al CU Consultar Propietarios de Vehículo INTTT.



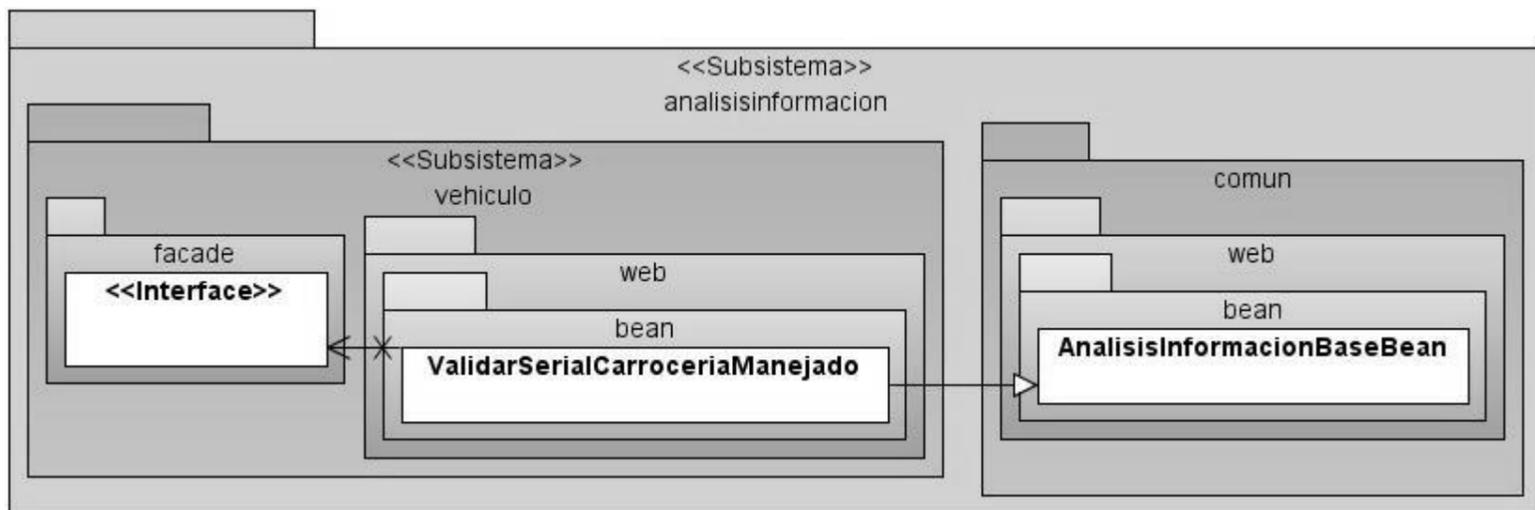
Anexo 3.18 Diagrama de Clases del Diseño pertenecientes al CU Consultar Propietarios de Vehículo INTTT.



Anexo 3. 19 Diagrama de Clases del Diseño Web pertenecientes al CU Registrar Recuperación de Vehículo.

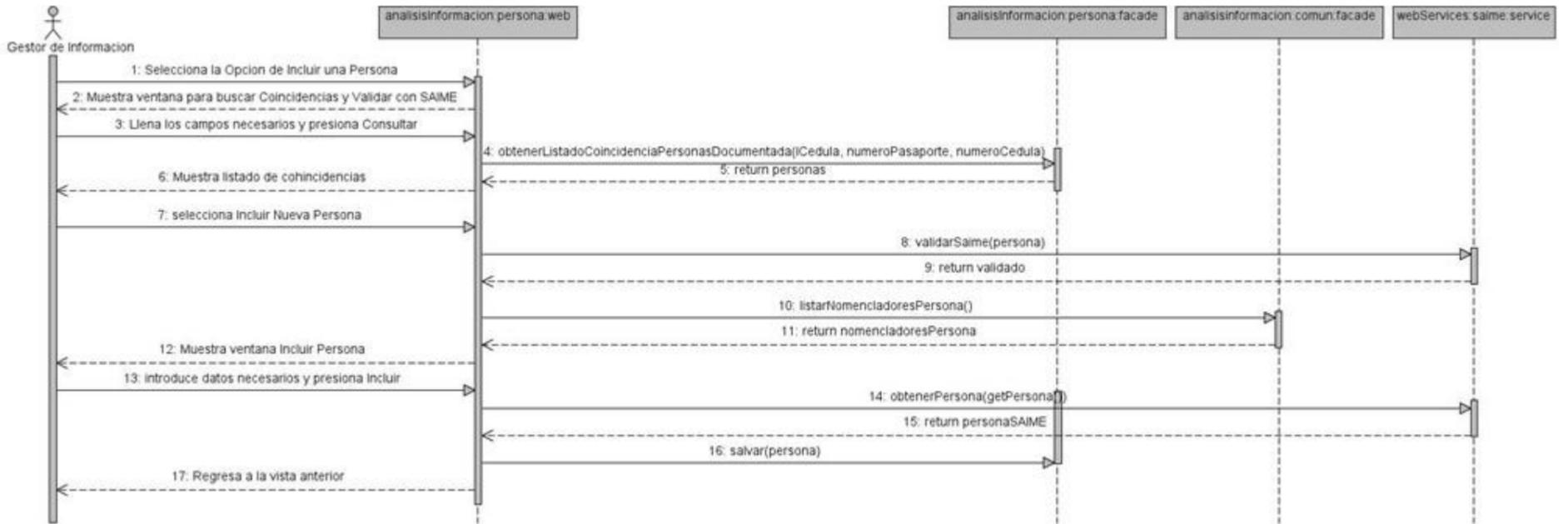


Anexo 3.21 Diagrama de Clases del Diseño Web pertenecientes al CU Validar Serial de Carrocería.

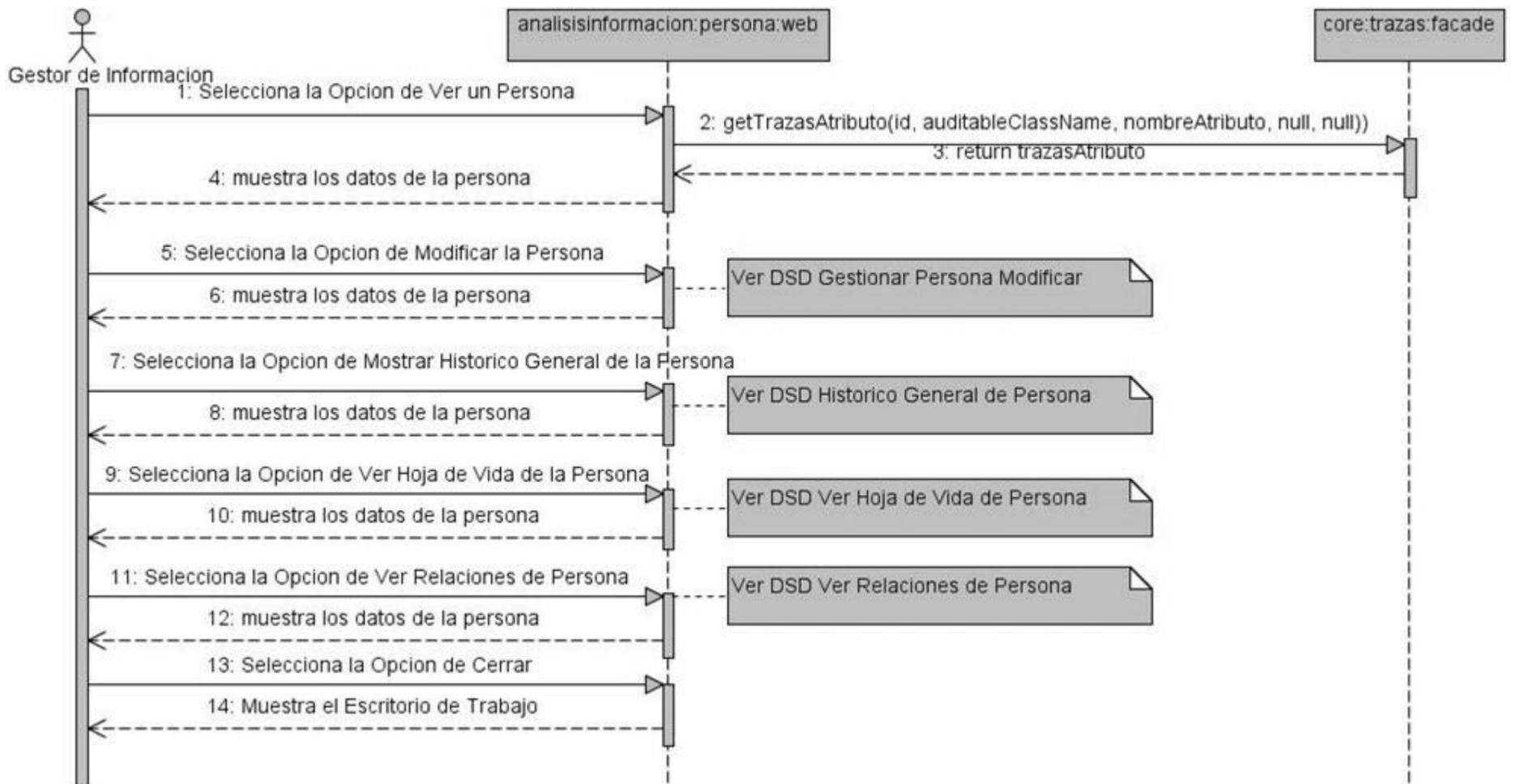


Anexo 3.22 Diagrama de Clases del Diseño pertenecientes al CU Validar Serial de Carrocería.

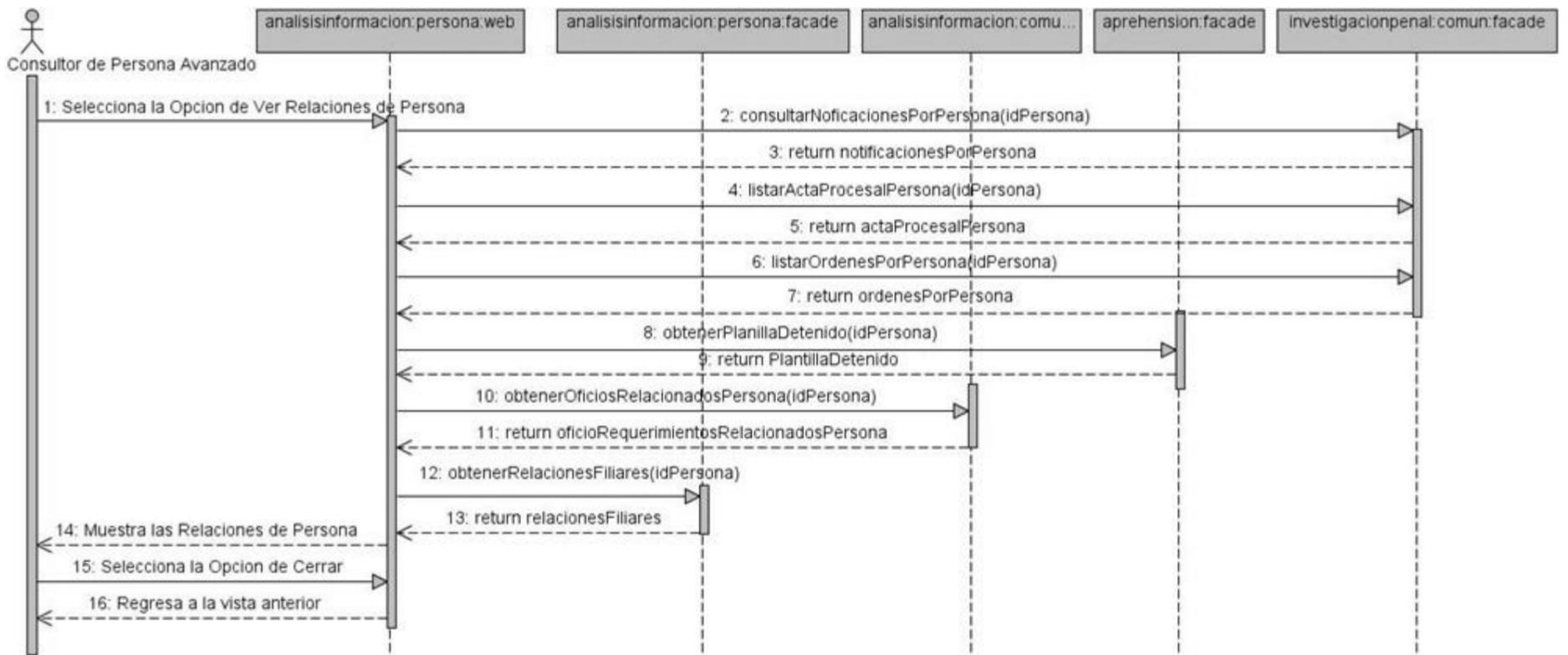
ANEXO 4: REALIZACIONES DE LOS CASOS DE USO MÁS SIGNIFICATIVOS DEL DISEÑO.



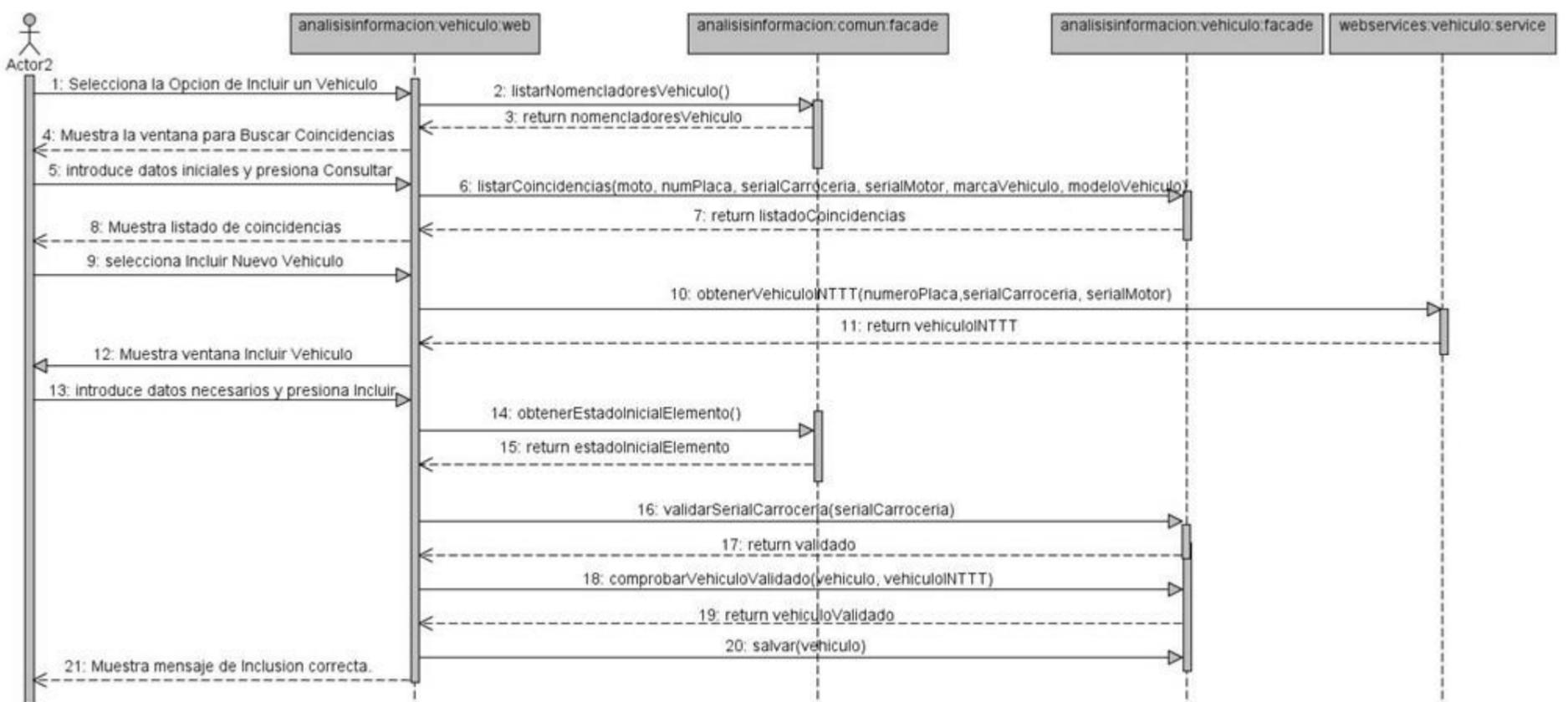
Anexo 4.1 Diagrama de Secuencia perteneciente al CU Gestionar Persona: Incluir.



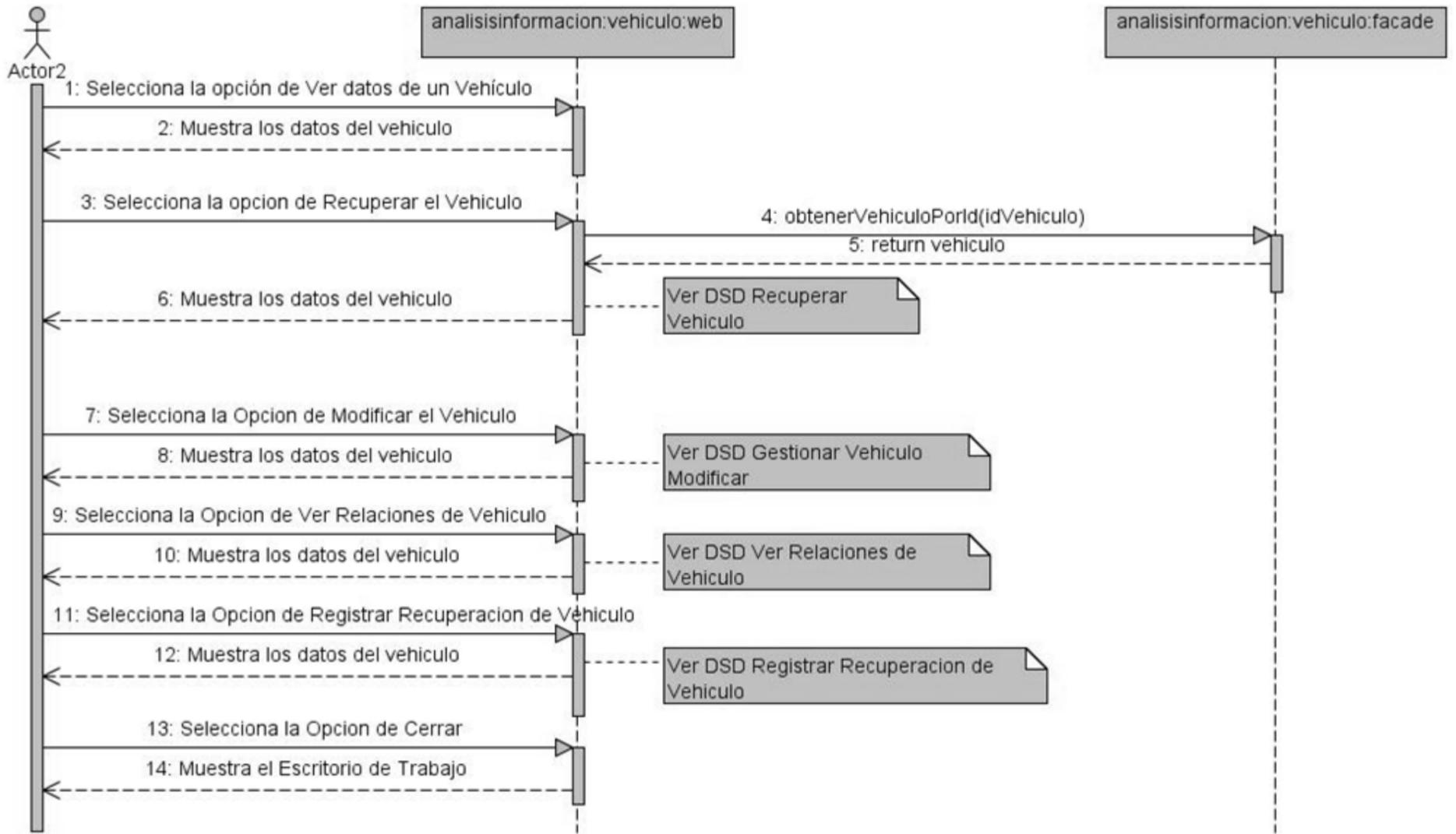
Anexo 4.2 Diagrama de Secuencia perteneciente al CU Gestionar Persona: Ver.



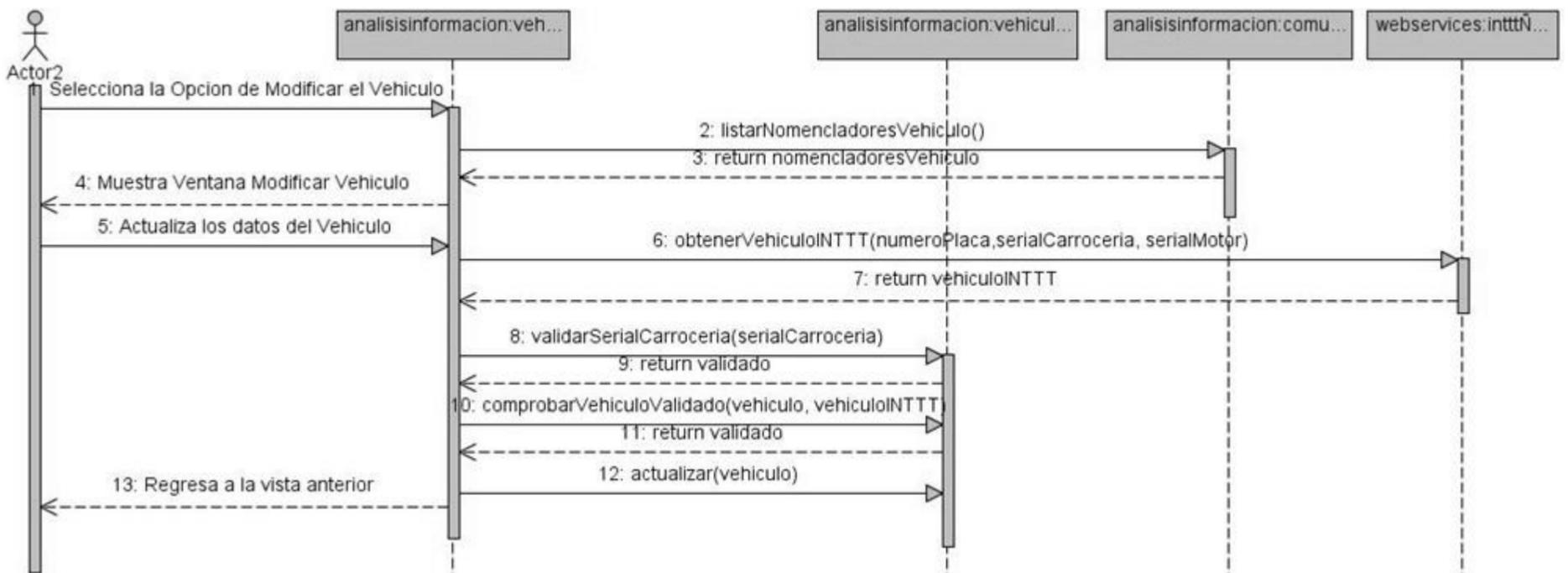
Anexo 4.3 Diagrama de Secuencia perteneciente al CU Consultar Persona.



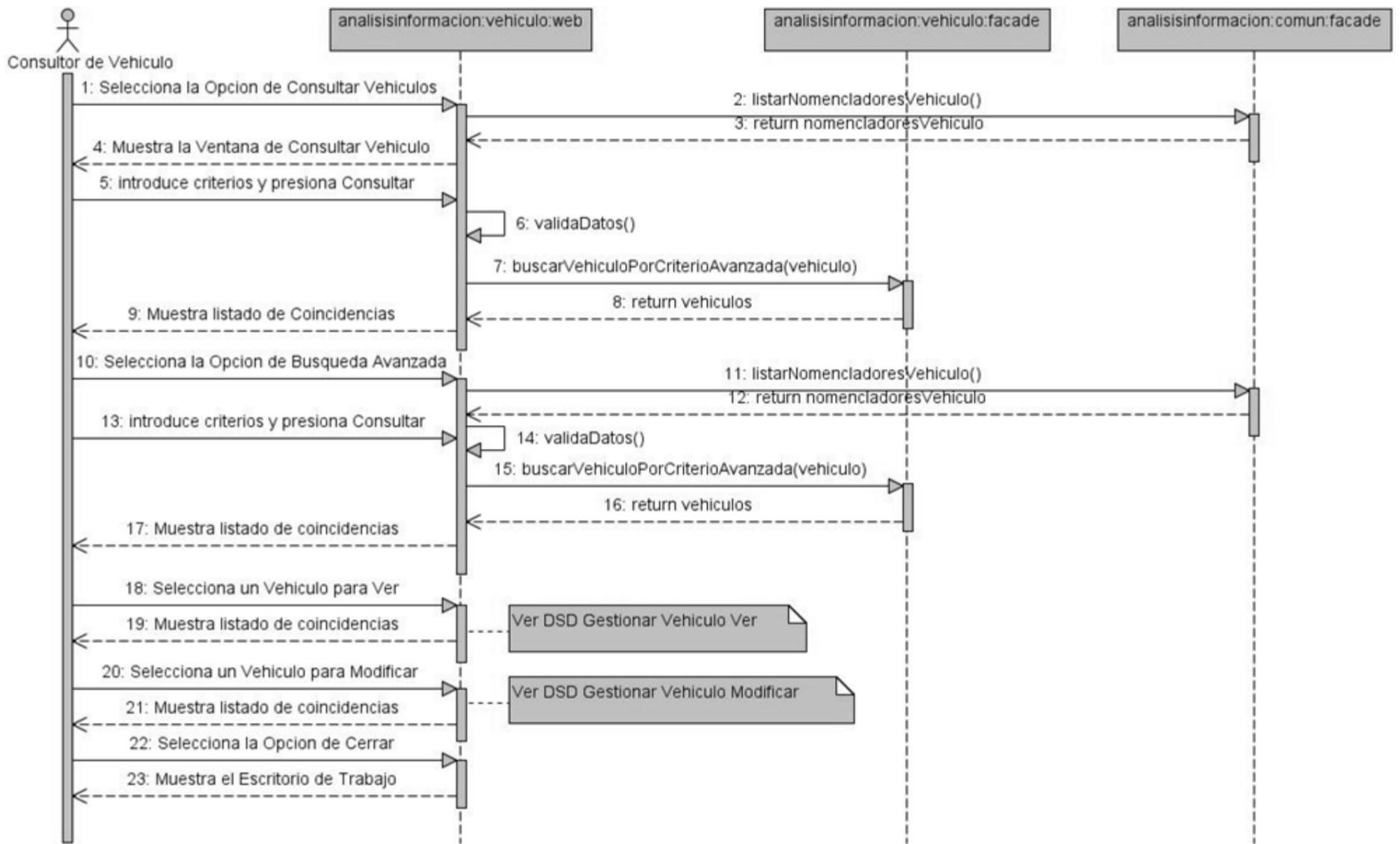
Anexo 4.4 Diagrama de Secuencia perteneciente al CU Gestionar Vehículo: Incluir.



Anexo 4.5 Diagrama de Secuencia perteneciente al CU Gestionar Vehículo: Ver.

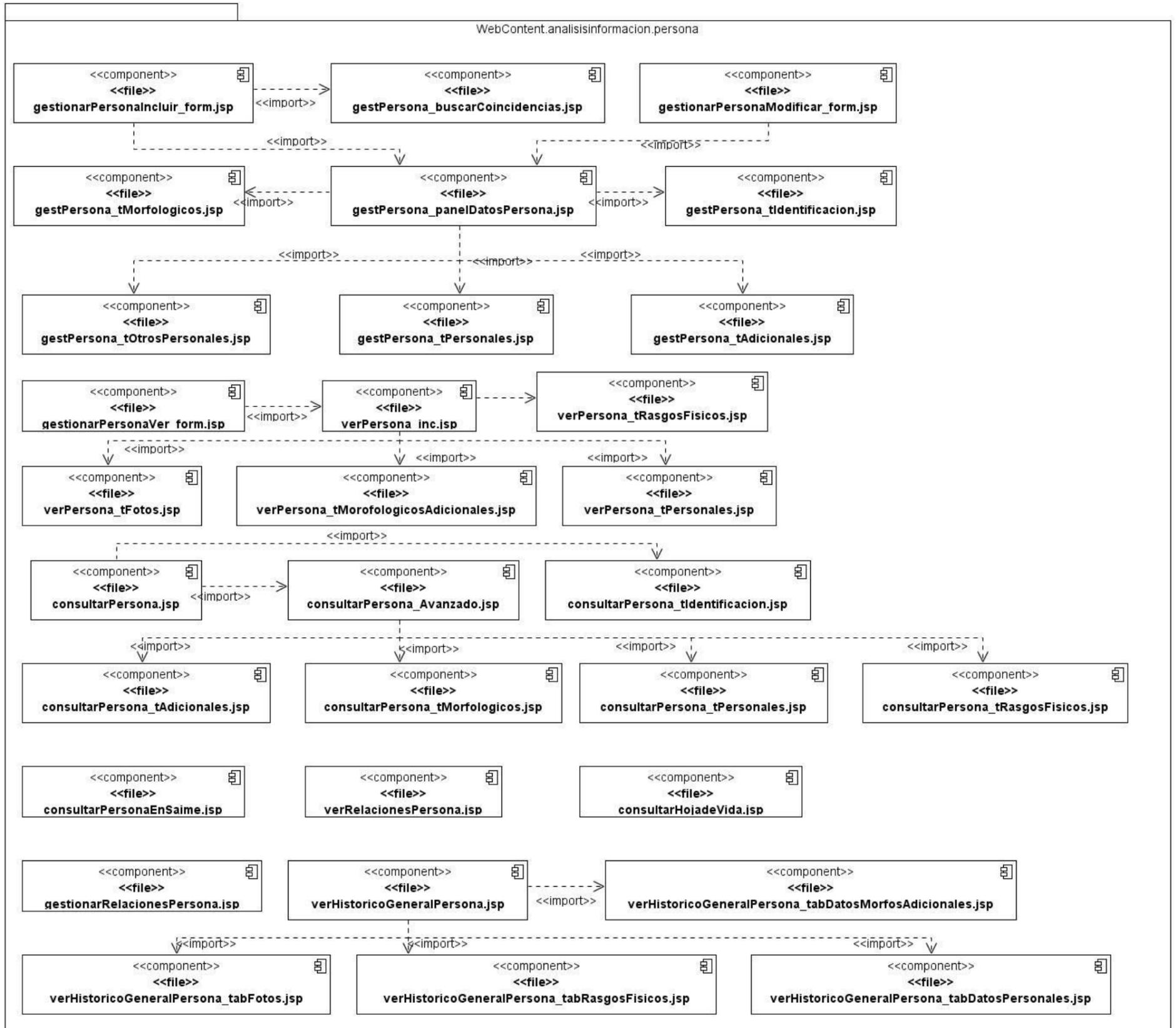


Anexo 4.6 Diagrama de Secuencia perteneciente al CU Gestionar Vehículo: Modificar.

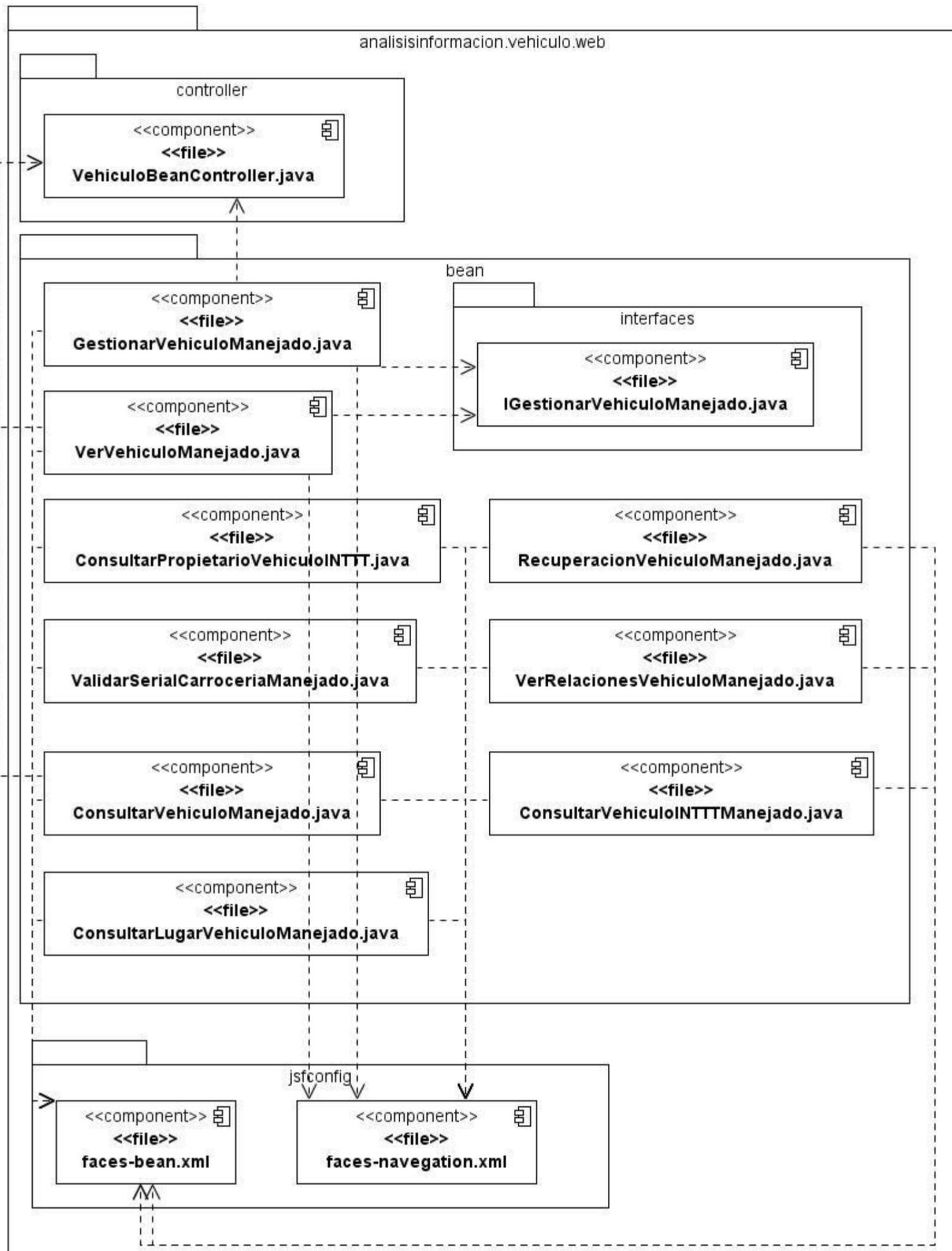


Anexo 4.7 Diagrama de Secuencia perteneciente al CU Consultar Vehículo.

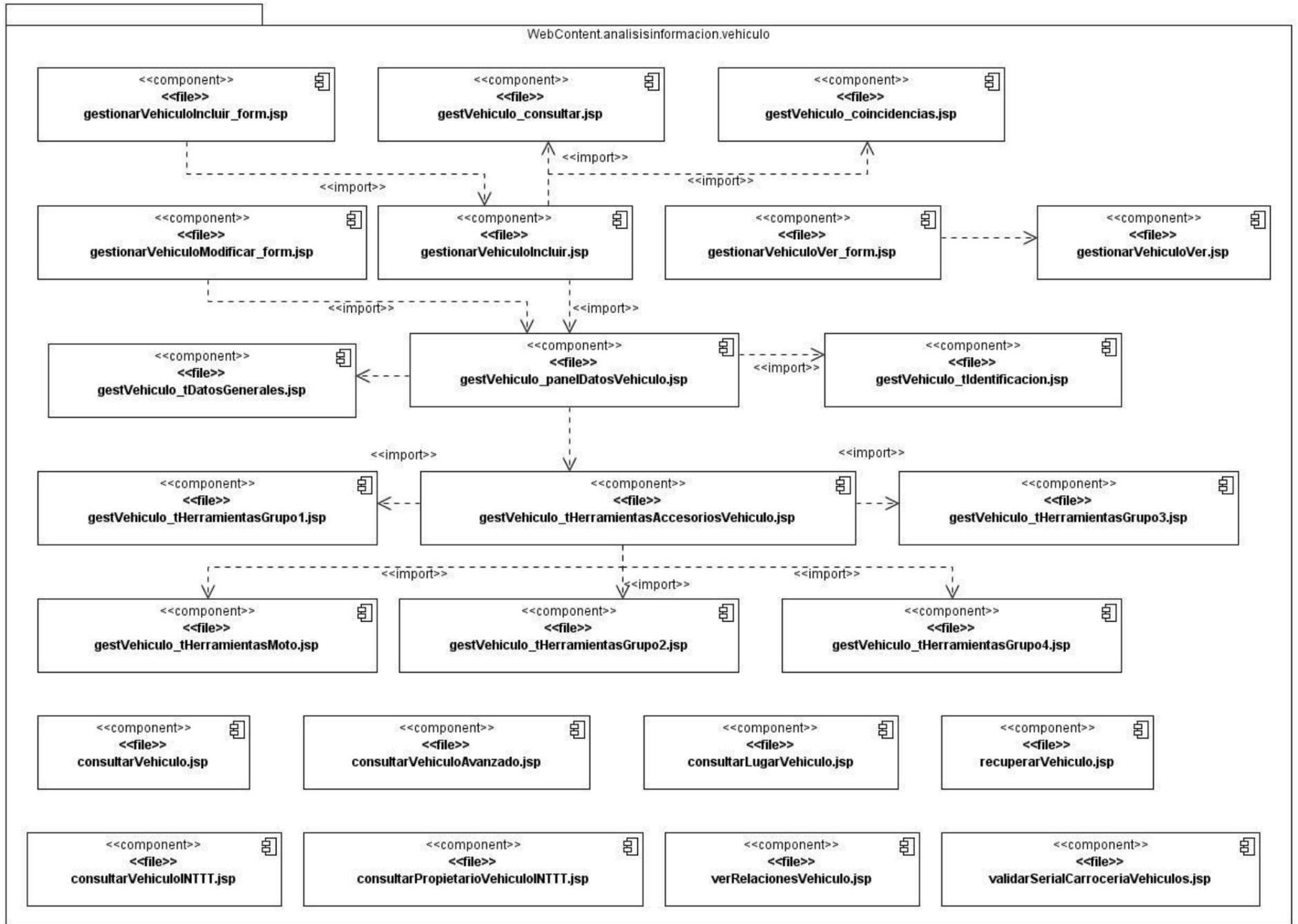
ANEXO 5: DIAGRAMAS DE COMPONENTES DE IMPLEMENTACIÓN.



Anexo 5.1 Diagrama de Componentes de Implementación Web Perteneciente al Sub-Módulo Persona.



Anexo 5.2 Diagrama de Componentes de Implementación Perteneciente al Sub-Módulo Vehículo.



Anexo 5.3 Diagrama de Componentes de Implementación Web Pertenciente al Sub-Módulo Vehículo.