

Universidad de las Ciencias Informáticas

*Análisis, Diseño e Implementación de los
Sub Módulos*

***Denuncia y Control de
Investigación
pertencientes al Módulo
de Investigación Penal
del Sistema de Investigación e Información
Policial (SIIPOL).***

*Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
informáticas.*

Autores:

Adriel Alejandro Aliaga Benavides.

Eduardo Alfonso Sánchez.

Tutor:

Ing. Humberto Ribero Guevara.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los autores de este trabajo y autorizamos 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 firmamos la presente a los ____ días del mes de _____ del año 2009.

AUTOR
<NOMBRE AUTOR>

AUTOR
<NOMBRE AUTOR>

TUTOR
<NOMBRE TUTOR>

AGRADECIMIENTOS

A la Revolución, a Fidel y a la UCI por darnos la oportunidad de formarnos como profesionales.

A nuestros compañeros de aula, que día a día han hecho de la universidad una etapa para recordar, en especial a las muchachas y muchachos de nuestro antiguo grupo.

A nuestro equipo de desarrollo, Arian Morejon, Shael Garcia, Jorge Luis, Yander Caceres y en especial a Adonys Alea.

A nuestros familiares, padres, abuelas (os), hermanas (nos), tías (os) que de una forma u otra han contribuido a este resultado.

A Enerys Mesa Morales por haber formado parte de mi vida en la universidad dando en todo momento su apoyo.

DEDICATORIA

Dedicado a la persona que ha hecho realidad este sueño por encima de todas las dificultades, la más especial que he conocido y la que siempre estará en mi corazón: Yaquelin Sanchez (Mi Madre).

Eduardo Alfonso Sanchez.

Dedicado sobre todo a mis padres que han dado de ellos lo mejor para que mi formación fuera la mejor posible, los que siempre estarán apoyándome en todas las dificultades que me muestre la vida, por supuesto a mi hermana también: María Teresa Benavides Cutiño, Segundo Gilfredo Aliaga Céspedes (Mis Padres) y Lázara Grisel Aliaga Benavides (Mi Hermana).

Adriel A. Aliaga Benavides.

RESUMEN.

El Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC), es uno de los organismos más importantes para la investigación criminal en la República Bolivariana de Venezuela. Para su gestión diaria utiliza el Sistema Integrado de Información Policial (SIIPOL). Aunque dicho sistema agiliza las tareas de la institución, presenta algunos problemas que hicieron posible que se firmara un contrato entre ambos países con el objetivo de modernizarlo.

Como parte de este convenio bilateral se construye el Sistema de Investigación e Información Policial con tecnología de punta. Se destacan el uso de la metodología RUP, los framework hibernate, Java Server Faces, Spring, Xfire entre otros, destacar el uso de la tecnología Ajax y la autenticación por huellas dactilares e imágenes.

En el presente trabajo se presenta la documentación generada por los flujos análisis, diseño e implementación correspondiente a los Sub Módulos Denuncia y Control de Investigación del Módulo Investigación Penal. Se muestran además aspectos de los test realizados con el fin de validar el ajuste a los requisitos funcionales y no funcionales definidos con el cliente.

ABSTRACT

The CICPC is one of the most important institutions of criminal investigation in the Bolivarian Republic of Venezuela. For its daily management it uses the Integrated System of Police Information (SIIPOL). Although said system speeds the task of the institution, it also presents some problems that allowed the signing of a contract between both countries aiming to its updating.

As part of this bilateral agreement the Police Investigation and Information System is built, using state of the art technology: RUP software methodology, application frameworks such as Hibernate, Java Server Faces, Spring and XFire, AJAX technology and authentication using fingerprints and images.

In this paper we present the documentation generated by the flows of analysis, design and implementation corresponding to the sub modules of Denouncement and Investigation Control of the Module of Penal Investigation Also, aspects of the test applied in order to validate functional and non functional requisites defined by the client are shown.

ÍNDICE DE CONTENIDOS.

INTRODUCCIÓN.....	14
CAPÍTULO 1.....	19
FUNDAMENTACIÓN TEÓRICA.....	19
1.1. Sistemas de gestión de información.....	19
1.1.1. Sistema Integrado de Información Policial.....	19
1.1.2. Sistema Territorial de Emergencias y Gestión Policial (STEGPOL)....	20
1.1.3. Sistema de Gestión Penitenciaria.	20
1.2. Análisis de la Propuesta de Solución.	20
1.2.1. Metodología Utilizada.....	22
1.2.2. Lenguaje de Modelado.....	23
1.2.3. Herramientas Case.	24
1.2.4. Plataforma de Desarrollo.....	25
1.2.5. Entorno de Desarrollo.	26
1.2.6. Frameworks Utilizados.....	27
Conclusiones.	34
CAPITULO 2.....	35
ANÁLISIS Y DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SISTEMA. .	35
2.1. Análisis.....	35
2.1.1. Entrada al Análisis.	36
2.1.2. Análisis de casos de usos.....	37
2.2. Diseño.....	50
2.2.1. Aspectos significativos del Diseño.	50
2.2.2. Clases significativas de la Arquitectura.	51
2.2.3. Patrones.	52
2.2.4. Diagrama de clases del diseño.	53
2.2.5. Clases significativas propias de la solución.....	64
2.2.6. Realizaciones de casos de uso.....	66
2.3. Modelo de Datos.	69
2.3.1. Diagramas de Tablas del Modelo Relacional.	69
2.4. Modelo de Implementación.....	74
2.4.1. Diagramas de subsistemas de implementación.	74
2.4.2. Diagrama de componentes.....	75

Conclusiones.....	80
CAPITULO 3.....	81
VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	81
3.1. Métodos de Pruebas.....	81
3.1.1. Prueba de Caja Blanca.....	81
3.1.2. Prueba de Caja Negra.....	81
3.2. Tipos de Pruebas.....	81
3.2.1. Prueba de Funcionalidad.....	81
3.3. Niveles de Pruebas.....	82
3.3.1. Pruebas Unitarias.....	82
3.3.2. Pruebas de Integración.....	82
3.3.3. Pruebas de Aceptación.....	85
Conclusiones.....	88
CONCLUSIONES.....	89
RECOMENDACIONES.....	90
GLOSARIO.....	91
ANEXOS.....	94
Anexo 1: Diagrama de Procesos del CICPC.....	94
Anexo 2: Diagrama de casos de usos perteneciente al Sub Módulo Denuncia.....	94
Anexo 3: Diagrama de casos de usos perteneciente al Sub Módulo Control de Investigación. Primera parte.....	95
Anexo 4: Diagrama de casos de usos perteneciente al Sub Módulo Control de Investigación. Segunda parte.....	96
Anexo 5: CU Asignar / Reasignar Actas Procesales. Descripción reducida.....	97
Anexo 6: CU Aprobar Diligencias Penales. Descripción reducida.....	97
Anexo 7: CU Autorizar Diligencias. Descripción reducida.....	98
Anexo 8: CU Comprobar Punibilidad del Hecho. Descripción reducida.....	98
Anexo 9: CU Comprobar Repetición de Denuncia. Descripción reducida.....	99
Anexo 10: CU Concluir Acta Procesal. Descripción reducida.....	99
Anexo 10: CU Remitir Actas Procesales a Fiscalía. Descripción reducida.....	100

Anexo 11: CU Revisar Diligencias Penales. Descripción reducida.	100
Anexo 12: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Análisis.....	101
Anexo 13: CU Aprobar Diligencias Penales. Diagrama de Clases del Análisis.....	101
Anexo 14: CU Autorizar Diligencias. Diagrama de Clases del Análisis.	102
Anexo 15: CU Comprobar Punibilidad del Hecho. Diagrama de Clases del Análisis.	102
Anexo 16: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Análisis.	102
Anexo 17: CU Concluir Acta Procesal. Diagrama de Clases del Análisis.	102
Anexo 18: CU Remitir Acta Procesales. Diagrama de Clases del Análisis.	103
Anexo 19: CU Revisar Diligencias Penales. Diagrama de Clases del Análisis.	103
Anexo 20: CU Asignar / Reasignar Actas Procesales. Diagrama de Colaboración del Análisis.....	104
Anexo 21: CU Aprobar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Aprobar.....	104
Anexo 22: CU Aprobar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Anular.....	105
Anexo 23: CU Autorizar Diligencias. Diagrama de Colaboración del Análisis.....	105
Anexo 24: CU Comprobar Punibilidad del Hecho. Diagrama de Colaboración del Análisis.....	106
Anexo 25: CU Comprobar Repetición de Denuncia. Diagrama de Colaboración del Análisis.....	106
Anexo 26: CU Concluir Acta Procesal. Diagrama de Colaboración del Análisis. ..	107
Anexo 27: CU Remitir Actas Procesales. Diagrama de Colaboración del Análisis.	107
Anexo 28: CU Revisar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Aprobar.....	108
Anexo 29: CU Revisar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Rechazar.....	108

Anexo 30: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Diseño. Primera Parte.	109
Anexo 31: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Diseño. Segunda Parte.	110
Anexo 32: CU Aprobar Diligencias Penales. Diagrama de Clases del Diseño. Primera Parte.	111
Anexo 33: CU Aprobar Diligencias Penales. Diagrama de Clases del Diseño. Segunda Parte.	112
Anexo 34: CU Autorizar Diligencias. Diagrama de Clases del Diseño. Primera Parte.	113
Anexo 35: CU Autorizar Diligencias. Diagrama de Clases del Diseño. Segunda Parte.	114
Anexo 36: CU Comprobar Punibilidad del Hecho. Diagrama de Clases del Diseño. Primera Parte.	115
Anexo 37: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Diseño. Primera Parte.	115
Anexo 38: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Diseño. Segunda Parte.	116
Anexo 39: CU Concluir Acta Procesal. Diagrama de Clases del Diseño. Primera Parte.	117
Anexo 40: CU Concluir Acta Procesal. Diagrama de Clases del Diseño. Segunda Parte.	118
Anexo 41: CU Remitir Actas Procesales. Diagrama de Clases del Diseño. Primera Parte.	119
Anexo 42: CU Remitir Actas Procesales. Diagrama de Clases del Diseño. Segunda Parte.	120
Anexo 43: CU Revisar Diligencias Penales. Diagrama de Clases del Diseño. Primera Parte.	121
Anexo 44: CU Revisar Diligencias Penales. Diagrama de Clases del Diseño. Segunda Parte.	122

Anexo 45: CU Asignar / Reasignar Actas Procesales. Diagrama de Contratos entre Paquetes.....	123
Anexo 46: CU Aprobar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Aprobar.....	123
Anexo 47: CU Aprobar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Anular.....	123
Anexo 48: CU Autorizar Diligencia. Diagrama de Contratos entre Paquetes.	124
Anexo 49: CU Comprobar Punibilidad del Hecho. Diagrama de Contratos entre Paquetes.....	125
Anexo 50: CU Concluir Acta Procesal. Diagrama de Contratos entre Paquetes... ..	125
Anexo 51: CU Remitir Actas Procesales. Diagrama de Contratos entre Paquetes.	125
Anexo 52: CU Revisar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Aprobar.....	126
Anexo 53: CU Revisar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Rechazar.....	126
Bibliografía.....	128

ÍNDICE DE FIGURAS.

Ilustración 1. Propuesta de Solución.....	21
Ilustración 2. Patrón n capas.....	21
Ilustración 3. Dirigido por casos de usos.....	22
Ilustración 4. Flujos y Fases de RUP.	23
Ilustración 5. Ciclo de vida bajo la tecnología JSF.	29
Ilustración 6. Estructura de Spring.	30
Ilustración 7. Arquitectura básica de un servicio web.	31
Ilustración 8. Estructura de Hibernate.	33
Ilustración 9. Descripción reducida. Iniciar Investigación.	38
Ilustración 10. Análisis. Iniciar Investigación. Diagrama de Clases	38
Ilustración 11. Interfaz. Iniciar Investigación.....	39
Ilustración 12. Descripción reducida. Gestionar Auto de Inicio.....	40
Ilustración 13. Análisis. Gestionar Auto de Inicio. Diagrama de Clases.....	40
Ilustración 14. Interfaz. Incluir Auto de Inicio.....	41
Ilustración 15. Descripción reducida. Consultar Libro de Control de investigación.	42
Ilustración 16. Análisis. Consultar Libro de Control de Investigación. Diagrama de Clases.	42
Ilustración 17. Interfaz. Consultar Libro de Control de Investigación. Búsqueda Avanzada.....	43
Ilustración 18. Descripción reducida. Gestionar Solicitud de Autorización de Diligencia Penal.	44
Ilustración 19. Análisis. Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases.	44
Ilustración 20. Incluir Solicitud de Autorización de Diligencia Penal.	45
Ilustración 21.Descripción reducida. Transferir copia de Denuncia policial SIGEPOL. 46	
Ilustración 22.Análisis. Transferir copia de Denuncia policial SIGEPOL. Diagrama de Clases.	46
Ilustración 23. Interfaz. Transferir copia de denuncia policial SIGEPOL.....	47
Ilustración 24. Análisis. CU Iniciar Investigación. Diagrama de Colaboración.	48
Ilustración 25. Análisis. CU Gestionar Auto de Inicio. Diagrama de Colaboración. Escenario Incluir.	48
Ilustración 26. Análisis. CU Consultar Libro de Control de Investigación.....	49
Ilustración 27. Análisis. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Colaboración. Escenario Incluir.	49
Ilustración 28. Análisis. CU Transferir copia de denuncia policial SIGEPOL. Diagrama de Colaboración.....	50
Ilustración 29. Estructura Modular de la Solución.	51
Ilustración 30. Diseño. CU Iniciar Investigación. Diagrama de Clases. Parte I.	54
Ilustración 31. Diseño. CU Iniciar Investigación. Diagrama de Clases. Parte II.	55
Ilustración 32. Diseño. CU Gestionar Auto de Inicio. Diagrama de Clases. Parte I.	56
Ilustración 33. Diseño. CU Gestionar Auto de Inicio. Diagrama de Clases. Parte II. ...	57
Ilustración 34.Diseño. CU Consultar Libro de Control de Investigación. Diagrama de Clases. Parte I.	58
Ilustración 35.Diseño.CU Consultar Libro de Control de Investigación. Diagrama de Clases. Parte II.	59

Ilustración 36. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases. Parte I.	60
Ilustración 37. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases. Parte II.	61
Ilustración 38. Diseño. Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Clases. Parte I.	62
Ilustración 39. Diseño. Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Clases. Parte II.	63
Ilustración 40. Diseño. CU Iniciar Investigación. Diagrama de Contratos entre Paquetes.	66
Ilustración 41. Diseño. CU Gestionar Auto de Inicio. Diagrama de Contratos entre Paquetes. Escenario Incluir.	66
Ilustración 42. Diseño. CU Consultar Libro de Control de Investigación. Diagrama de Contratos entre Paquetes.	67
Ilustración 43. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Contrato entre Paquetes. Escenario Incluir.	67
Ilustración 44. Diseño. CU Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Contratos entre Paquetes.	68
Ilustración 45. Diseño. Acta Procesal. Modelo de Datos.	69
Ilustración 46. Diseño. Denuncia. Modelo de Datos.	70
Ilustración 47. Diseño. Delito. Modelo de Datos.	71
Ilustración 48. Diseño. Denuncia SIGEPOL. Modelo de Datos.	72
Ilustración 49. Diseño. Diligencia. Modelo de Datos.	72
Ilustración 50. Diseño. Conexión entre las Principales Tablas. Modelo de Datos.	73
Ilustración 51. Diagramas de Subsistemas de Implementación.	74
Ilustración 52. Implementación. Páginas. Sub Módulo Denuncia. Diagrama de Componentes.	75
Ilustración 53. Implementación. Web. Sub Módulo Denuncia. Diagrama de Componentes.	76
Ilustración 54. Implementación. Páginas. Sub Módulo Control de Investigación. Diagrama de Componentes.	77
Ilustración 55. Implementación. Web. Sub Módulo Control de Investigación. Diagrama de Componentes.	78
Ilustración 56. Implementación. Negocio y DAO. Diagrama de Componentes.	79
Ilustración 57. Relación de las no conformidades en relación a los casos de usos.	87
Ilustración 58. Comportamiento de las no conformidades.	88

INTRODUCCIÓN.

El aumento de la criminalidad ha tenido importantes repercusiones en América Latina desde inicios de los años noventa. Los costos económicos y sociales que ha implicado el incremento sostenido de las tasas de criminalidad y práctica de la violencia en usualmente todos los países de la región son sustanciales. Pero esta crisis ha servido también para evidenciar las profundas carencias institucionales de los organismos encargados de enfrentarla tales como Justicia y Policía.

Según un estudio realizado por la Alcaldía de Chacao (1) en relación a la situación de violencia en la República Bolivariana de Venezuela, se detectó que el homicidio es la principal causa de muerte entre personas de 15 a 29 años así como que los índices de asesinatos han aumentado en un 128% desde 1998 al 2005 y más alarmante aún, las muertes por enfrentamiento armado en un 253% en igual período de tiempo. Esta situación coloca al país entre uno de los más violentos del mundo, siendo necesario por parte del gobierno una atención diferenciada.

El Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC), es el organismo de investigación criminal contemporáneo (*antiguamente Cuerpo Técnico de Policía Judicial*), consecuencia del Proyecto de Seguridad Ciudadana, llevado por el ejecutivo nacional al tomar la presidencia el Teniente Coronel Hugo Rafael Chávez Frías.

Desde su surgimiento, el CICPC ha tenido como misión, *garantizar la eficiencia en la investigación del delito, mediante su determinación científica, asegurando el ejercicio de la acción penal que conduzca a una sana administración de justicia* (2) y entre sus objetivos estratégicos, optimizar las acciones de investigación criminal tendentes a lograr el esclarecimiento de los hechos delictivos. Para todo esto contaba con el Sistema Integrado de Información Policial (SIIPOL).

Como consecuencia de los convenios de colaboración de la 7^{ma} mixta, Cuba y Venezuela firmaron un contrato con el objetivo de automatizar el CICPC. Como parte de la planificación general, se pretende desarrollar el nuevo Sistema de Investigación e Información Policial (SIIPOL), tomando como referencia el sistema existente, que incremente las posibilidades de uso y gestión de la información, con tiempos de respuesta mejores.

En el levantamiento de requisitos realizados por los analistas, se recolectaron una serie de procesos candidatos a automatizar, los cuales no estaban contemplados en el antiguo SIIPOL. Entre los módulos identificados estaba el de Investigación Penal.

Los Sub Módulos Denuncia y Control de Investigación pertenecientes al Módulo de Investigación Penal, recogen una serie de casos de usos que son clave en los procesos de una averiguación, por una parte, el primero abarca el comienzo de esta, y el segundo aunque no menos importante, agrupa funcionalidades indispensables para el desarrollo de la misma.

Este SIIPOL no abarcaba procesos tan significativos como el inicio de la investigación; solo se almacenaba el “*Libro de Control de Investigación*” y las relaciones que tenía con los diferentes elementos, quedando fuera del sistema, valiosa información que se acopiaba en la denuncia.

Los procesos antes descritos eran lentos, recogían una serie de pasos que afectaba el trabajo interno de los funcionarios, muestra de ello, lo representa el inicio de una investigación, cuando una persona se dirigía a un despacho del CICPC a reportar un hecho supuestamente punible, en ese momento, el oficial encargado de tomar la denuncia llenaba tres modelos con la ayuda de alguna herramienta Office: *Denuncia Común*, *Oficio de Participación al Fiscal* y *El Libro de Control de Investigación*, el cual se le entregaba al denunciante como comprobante de su declaración y una copia pasaba al Área de Análisis y Seguimiento de información para su posterior incorporación al SIIPOL.

La no incorporación al Sistema de los documentos *Denuncia Común* y *Oficio de Participación* así como la falta de agilidad en la confección del *Libro de Control de Investigación* ya que el funcionario no tenía acceso directo al SIIPOL son una muestra de los problemas planteados anteriormente.

Estudiada la situación, los analistas se dispusieron a especificar un sistema conformado por módulos, algunos de estos aledaños a las áreas del CICPC, resultando un conjunto de requisitos funcionales y no funcionales con el fin de mejorar las prestaciones de la Institución.

Por todo lo anteriormente expuesto, se declara el siguiente **problema científico**:

¿Cómo garantizar el cumplimiento de los requisitos funcionales y no funcionales asociados a los Sub Módulos Denuncia y Control de Investigación del Módulo Investigación Penal en el SIIPOL?

Teniendo como **objeto de estudio**: el proceso de desarrollo del software de gestión SIIPOL y el **campo de acción** los Sub Módulos Denuncia y Control de Investigación pertenecientes al Módulo Investigación Penal del sistema SIIPOL.

En este trabajo se tiene como **objetivo general**: analizar, diseñar e implementar los Sub Módulos Denuncia y Control de Investigación del Módulo Investigación Penal en el SIIPOL de modo que soporten los requisitos funcionales y no funcionales de la aplicación.

Cómo objetivos específicos:

- Comprender la lógica del negocio.
- Efectuar un análisis de los Sub Módulos para lograr una mayor comprensión de sus procesos.
- Diseñar los artefactos necesarios del modelo de diseño para responder a los requisitos funcionales y no funcionales obtenidos como resultado de aplicar la Ingeniería de Requerimientos.
- Implementar las capas de acceso a datos, negocio y presentación relacionados con los Sub Módulos Denuncia y Control de Investigación del Módulo Investigación Penal en el SIIPOL.
-

Como preguntas científicas:

- 1- ¿Qué mecanismos y patrones de diseño son factible utilizar para el diseño de la solución final?
- 2- ¿Cuáles serían las buenas prácticas de programación conjugadas con los patrones de diseño que permitan implementar una solución factible?
- 3- ¿Cuáles son las pautas de arquitectura y diseño que se deben tener en cuenta para integrar los Sub Módulos Control de Investigación y Denuncia al Módulo Investigación Penal del SIIPOL?

Por todo lo antes planteado la idea a defender es:

Si se aplica correctamente la metodología de desarrollo de software para el análisis, diseño e implementación de los Sub Módulos Denuncia y Control de Investigación, posibilitará la elaboración de un sistema informático que cumpla con los requisitos funcionales y no funcionales definidos para el mismo.

Para dar cumplimiento a estos objetivos específicos se plantearon las siguientes **tareas investigativas**:

- Consultar bibliografía referente al negocio.
- Investigar la metodología, lenguaje de implementación y de modelado, plataforma, frameworks, y otras herramientas escogidas para el desarrollo.
- Estudiar las pautas de arquitectura definidas para el desarrollo del sistema.
- Realizar el modelo de análisis y diseño de los Sub Módulos Denuncia y Control de Investigación.
- Realizar el modelo de implementación así como implementar los Sub Módulos Denuncia y Control de Investigación.
- Integrar la solución al sistema a medida que se realice el desarrollo de la misma.

Métodos Científicos Utilizados.

- *Analítico Sintético*

Se hizo un análisis de los documentos generados en la captura de requisitos, extrayendo y analizando los principales elementos relacionados a nuestro objeto de estudio.

Técnicas de Investigación

- *Análisis Documental*

Se realizó conforme al Método Analítico Sintético una exploración de toda la documentación generada por los analistas de nuestro proyecto, dígase, especificación de casos de usos, glosario de términos, diagramas de entidades así como también las especificaciones de la Arquitectura establecidas por este grupo de trabajo.

- *Entrevista*

Se ha realizado desde el comienzo de la investigación entrevistas con diversos roles del proyecto dígase analistas, arquitectos y demás con el fin de extraer información útil referente a nuestro campo de acción.

El presente trabajo está estructurado de la siguiente manera:

Capítulo 1. Se describen los principales conceptos, producto de una fundamentación teórica correspondiente a este trabajo de diploma, estos van desde el estudio de algunos sistemas similares hasta el análisis de una propuesta de solución práctica.

Capítulo 2. Se efectúa un análisis y diseño e implementación de los casos de usos pertenecientes a los Sub Módulos Denuncia y Control de Investigación, haciendo énfasis, en caso del diseño, a la propuesta arquitectónica y los patrones que rigieron esta actividad.

Capítulo 3. Este capítulo tiene como principal objetivo validar la propuesta de sistema. En él se explican algunos conceptos relacionados a los tipos de pruebas y finalmente se muestran los resultados más significativos obtenidos como resultado de los diferentes test efectuados.

CAPÍTULO 1.

FUNDAMENTACIÓN TEÓRICA.

En el presente capítulo se pretende explicar los principales conceptos relacionados a nuestro objeto de estudio así como dar una base teórica sustentable sobre el presente trabajo, realizar el estudio de algunos sistemas similares, las tendencias actuales para resolver este tipo de problema y presentar una propuesta de solución práctica.

1.1. Sistemas de gestión de información.

Un sistema de gestión de información puede ser definido como un conjunto de elementos que interactúan entre sí, con el fin de apoyar las actividades que se realizan en una organización o para automatizar los procesos de trabajo que se efectúan dentro de esta. Un sistema de gestión de información policial, como dice el término, se restringe a automatizar los procesos en entidades policiales. Una de las características más comunes de los sistemas de gestión policial es su confidencialidad; el desarrollo de este tipo de software no promueve la difusión de su documentación. Los sistemas de gestión de información policial permiten efectuar búsquedas ágiles y eficaces, por lo que el tiempo de respuesta para combatir los delitos es significativamente breve, ya que poseen numerosas funcionalidades que hacen que el trabajo que se realizaba de forma manual sea considerablemente rápido y efectivo.

A medida del avance de las tecnologías y del incremento de las cifras en los índices delictivos, se ha hecho necesaria la incorporación de estos tipos de sistemas a la gestión policiaca, donde cada vez se hace más imprescindible la automatización de grandes volúmenes de información para lograr un enfrentamiento exitoso ante la infracción.

1.1.1. Sistema Integrado de Información Policial.

El Sistema Integrado de Información Policial (antiguo SIIPOL) es un software que la República Bolivariana de Venezuela utiliza para la gestión de las tareas fundamentales. Este está desarrollado sobre el lenguaje Natural-Adabas y servidores de base datos SUN 6500, tecnología obsoleta que dificulta cualquier tarea de mantenimiento; además no abarca todas las labores de la organización y aunque hace más fácil la tarea de los funcionarios no fue confeccionado para llevar los procesos y mantener la información referenciada.

1.1.2. Sistema Territorial de Emergencias y Gestión Policial (STEGPOL).

El principal objetivo de este sistema es actuar como 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 actualmente operan en Chile.

El Sistema Territorial de Emergencias y Gestión Policial identifica en forma veraz y efectiva dónde geográficamente se están realizando o se han cometido actos delictivos a nivel territorial, apoyado por sistemas de información en-línea desde el lugar de los hechos, que permitan la acción rápida y coordinada entre los diferentes encargados de la seguridad ciudadana a nivel de país.

1.1.3. Sistema de Gestión Penitenciaria.

El Sistema de Gestión Penitenciaria o "Prisiones" como comúnmente se llama en la Universidad de Ciencias Informáticas donde se desarrolla específicamente por la facultad 4 comenzó a desarrollarse primero que la modernización del SIIPOL y presenta grandes similitudes por ser destinado a la República Bolivariana de Venezuela y utilizar una propuesta de solución análoga: java con los frameworks Hibernate, Spring y SpringSecurity. Este sistema sirvió de punto de partida para la investigación por las características antes mencionadas.

1.2. Análisis de la Propuesta de Solución.

Un proceso de desarrollo del software debe construirse sobre tecnologías, computadoras, sistemas operativos, estructuras de red, entornos de desarrollo, lenguajes de programación, frameworks para ser más eficiente el trabajo y herramientas que de una forma u otra automaticen las labores de la producción en menor o mayor escala.

El desarrollo de este tipo de producto ha evolucionado a lo largo del último cuarto del siglo pasado y principios de este y la tendencia junto al desarrollo de Internet ha llevado a la construcción de sistemas más grandes y complejos donde se requiere de mayor eficiencia, facilidad de mantenimiento, facilidad de uso, Interoperabilidad, seguridad y mejor tiempo de respuesta.

Con esta idea y con el fin de cumplir todos los requerimientos funcionales y no funcionales que la modernización del SIIPOL requiere, se ha decidido por parte de la dirección del proyecto construir una aplicación cliente-servidor mediante el protocolo https contactado a servidores *HP Integrity* para la aplicación y la base de datos respectivamente.

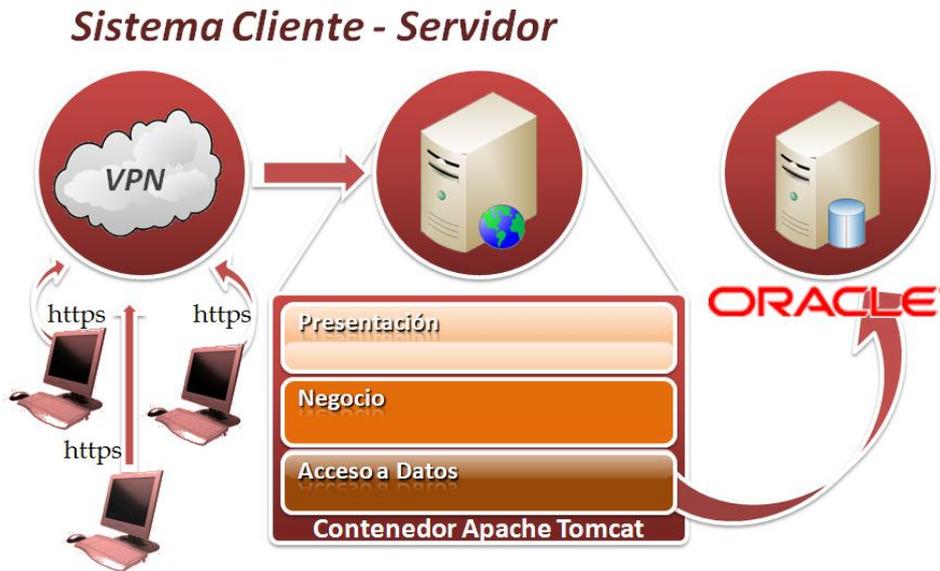


Ilustración 1. Propuesta de Solución.

Dicha aplicación será construida siguiendo el patrón arquitectónico *n* capas, donde cada capa solo conoce a su inmediata inferior.



Ilustración 2. Patrón *n* capas.

A continuación se describe la metodología, lenguajes, herramientas y frameworks utilizados para el desarrollo.

1.2.1. Metodología Utilizada.

Metodología se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Son vías que facilitan el descubrimiento de conocimientos seguros y confiables para solucionar los problemas que la vida plantea. En resumen son el conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal.

Una metodología de desarrollo del software define quien está haciendo qué, cuándo, y cómo alcanzar un objetivo específico. Esta proporciona normas para el desarrollo eficiente del software con calidad al captar las mejores prácticas que el estado actual de la tecnología permite.

Rational Unified Process

El Proceso Unificado del Software, RUP (Rational Unified Process), es un marco de trabajo genérico que puede especializarse en una gran variedad de sistemas de software (3), incorporando una serie de actividades que transforman los requisitos de un usuario en un producto informático.

Fue creado por Rational y en la actualidad constituye la metodología estándar más utilizada en el desarrollo Informático a escala mundial.

Sus características principales son:

Dirigido por casos de uso: Basándose en los casos de usos los desarrolladores crean una serie de modelos de análisis, diseño, implementación y demás que dan como consecuencia un producto de este tipo, convirtiéndose es un hilo conductor para todo el proceso.

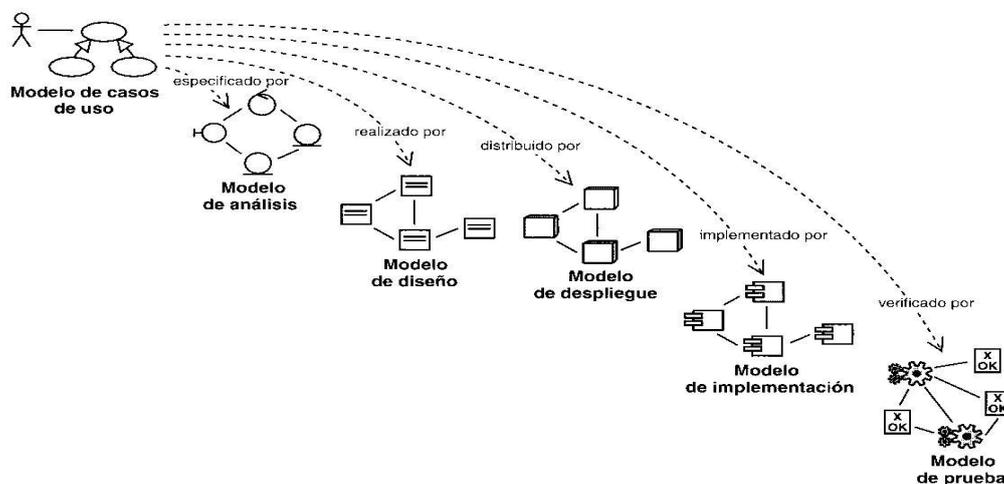


Ilustración 3. Dirigido por casos de usos.

Centrado en la arquitectura: La arquitectura tiene como objetivo ser de soporte a los casos de usos, es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado, incluye conceptos tanto estáticos como dinámicos y se ve influenciada por muchos factores como la plataforma de desarrollo, el sistema operativo, el gestor de base de datos, protocolos para la comunicación de red, etc.

Iterativo e incremental: La iteraciones referencian el transcurso entre los flujos de trabajos y los incrementos, al crecimiento del producto, permitiendo que en cada iteración se entregue una versión del software.

La táctica de este proceso es alcanzar su objetivo por medio del orden y documentación, lo que lo clasifica como una metodología robusta. Está definido por cuatro fases (inicio, elaboración, construcción y transición), y nueve flujos de trabajo (Modelado del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Gestión del Cambio y Configuraciones, Gestión de Proyecto y Entorno).

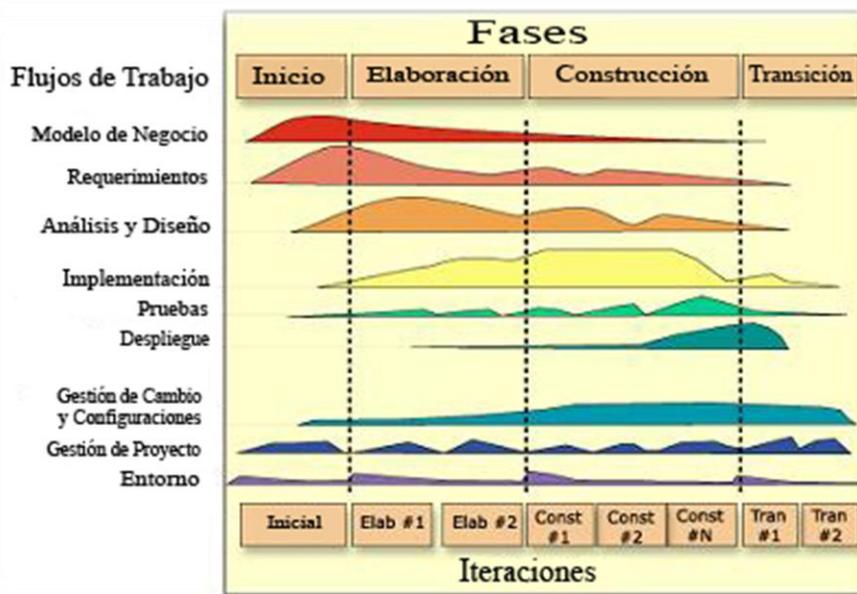


Ilustración 4. Flujos y Fases de RUP.

1.2.2. Lenguaje de Modelado.

Lenguaje Unificado de modelado. (UML)

La modelación ha sido una parte esencial tanto en la ingeniería, el arte y la construcción, tal es el caso que el diseño de un software supuestamente complejo podría complicarse si no se hace pleno uso de diagramas que lo identifiquen. La

modelación proporciona tres beneficios clave: visualización, complejidad de gestión y una comunicación clara. En la actualidad el auge de desarrollo de software o aplicaciones complicadas ha crecido hasta el punto de que un equipo de desarrollo de software ve la necesidad de la utilización de diagramas que ayuden al entendimiento pleno del software a desarrollar. Para ello, la solución se apoya en UML (Lenguaje Unificado de Modelado), lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Fue originalmente concebido por la Corporación Rational Software y tres de los más prominentes metodólogos en la industria de la tecnología y sistemas de información como lo son Grady Booch, James Rumbaugh e Ivar Jacobson, está respaldado por el OMG (Object Management Group). Es de saber que UML es un "lenguaje" para especificar y no para describir métodos o procesos o sea es el lenguaje en el que está descrito el modelo. Para la versión UML 2.0 hay 13 tipos diferentes de diagramas categorizados en: Diagramas de Estructura, Diagramas de Comportamiento y Diagramas de Iteración.

1.2.3. Herramientas Case.

CASE es una sigla, que corresponde a: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. El concepto de CASE es muy amplio; y una buena definición genérica, que pueda abarcar esa amplitud de conceptos, sería la de considerar a la Ingeniería de Software Asistida por Computación (CASE), como la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación. Concentrando nuestra atención en el uso de estas herramientas, para el desarrollo de proyectos informáticos que tengan como objetivo la automatización de procedimientos administrativos; se puede decir que: las herramientas CASE representan una forma que permite modelar los procesos de negocios de las empresas y desarrollar sistemas de información. En un sentido más amplio las Herramientas CASE se pueden ver como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: negocio, requerimiento, análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

- Ayuda a una rápida construcción de aplicaciones de calidad con un menor coste; permitiendo dibujar todos los tipos de diagramas así como generar código y documentación.
- Proporciona abundantes tutoriales de UML, demostraciones interactivas y proyectos.
- Cuenta con la posibilidad de diseñar los esquemas correspondientes al modelo Entidad-Relación.
- Apoya un conjunto de lenguajes, tanto en generación de código como ingeniería inversa: Java, C + +, CORBA IDL, PHP, XML Schema, Ada y Python, VB.NET, Flash ActionScript y archivos de mapeo de Hibernate.

1.2.4. Plataforma de Desarrollo.

Java es un lenguaje de programación que está diseñado tanto para Aplicaciones Web como de Escritorio. Fue desarrollado por la compañía Sun Microsystems en 1995 con el propósito de cubrir las necesidades tecnológicas de punta. Su principal característica y a su vez lo que lo diferencia ampliamente de los demás lenguajes de programación es la independencia que tiene a nivel de plataforma; esto se debe a que se le ha creado una máquina virtual para cada sistema operativo (SO). Lo cual significa que el desarrollador de software que programa en dicho lenguaje obvia la parte de preocuparse por el SO pues siempre el programa Java correrá. Es este uno de los motivos por el cual los desarrolladores de software preferentemente web optan por este lenguaje dada la necesidad de que en la web los clientes pueden tener diferentes aficiones en sus ordenadores, o sea, para una misma aplicación web unos trabajarán en Windows, Linux, Macintosh, etc. Haciendo un poco de historia este lenguaje originalmente fue pensado con la idea de utilizarse en cualquier tipo de equipo electrodoméstico pero dicha idea fracasó. Luego uno de los fundadores de Sun rescató la idea, esta vez su objetivo era utilizarlo en el ámbito de Internet y convirtieron a Java en un lenguaje potente, seguro y universal gracias a que lo puede utilizar todo el mundo pues es un lenguaje gratuito. El interés de esta propuesta consiste en que con Java se pueden programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee hacer con acceso a través de web se puede hacer utilizando Java. Este lenguaje incorpora muchos aspectos que en cualquier otro son extensiones, propiedad de empresas de software o fabricantes de ordenadores [threads (hilos de procesamiento), ejecución remota, componentes,

seguridad, acceso a bases de datos, etc.]. Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la informática moderna.

1.2.5. Entorno de Desarrollo.

Un entorno de desarrollo integrado: IDE, es un programa que incorpora un conjunto de herramientas y que tiene como objetivo ser de soporte a un programador para construir software.

Este puede incorporar un lenguaje o varios de ellos; es considerado un programa de aplicación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

A medida que ha avanzado el proyecto de modernización del SIIPOL, se han utilizado varios entornos de desarrollo para obtener los beneficios propios del impulso de la tecnología. Y aunque hasta la actualidad se utiliza el Red Hat Developer Studio se ha decidido describir las principales características de todos los utilizados.

Eclipse

Eclipse es un esqueleto donde se pueden montar herramientas para disímiles lenguajes, como lo expresa su sitio principal: un IDE para todo y para nada en particular”.

Presenta una arquitectura de plugins por lo que con los accesorios necesarios, se muestra como un potente y útil IDE para el desarrollo; es multiplataforma y brinda pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, wizards para la creación de proyectos, clases, test, componentes y refactorización.

Exadel Studio.

Es un entorno de desarrollo integrado *open source* basado en eclipse, para J2EE; entre sus bondades están:

1. Permite desarrollar aplicaciones empresariales con soporte de varias tecnologías de código abierto, incluyendo JSF, Struts, Hibernate, MyFaces, Oracle ADF, Shale, Spring, admitiendo a los desarrolladores trabajar fácilmente con múltiples frameworks, todos dentro de un mismo ambiente.
2. Está equipado con potentes funciones, incluyendo un editor para los archivos JSF y un asistente para Hibernate Object Relational Mapping (ORM).

Red Hat Developer Studio.

Entorno de desarrollo integrado open source basado en Eclipse, de la compañía Red Hat, que incluye JBoss Enterprise Middleware y Red Hat Enterprise Linux.

Developer Studio combina productos aportados a Red Hat por Exadel, Exadel Studio Pro, RichFaces y Ajax4jsf con software de middleware Jboss como JBoss Seam e Hibernate dentro de un potente entorno de desarrollo para aplicaciones SOA, Ajax y Java empresariales.

Entre sus características están:

1. Un modelo de programación unificado: Developer Studio aumenta y proporciona nuevas herramientas alrededor de JBoss Seam para construir aplicaciones de manera sencilla y consistente.
2. Potentes capacidades Ajax: proporciona un entorno de desarrollo Ajax integrado y potente con JBoss Seam y JBoss Ajax4jsf, componentes Web JBoss RichFaces, y herramientas What You See Is What You Get (WYSIWYG) para crear interfaces y páginas web que soportan Ajax.
3. Utilidades Java Platform Enterprise Edition (EE): Developer Studio hace más sencilla la construcción de aplicaciones Java EE, con capacidades como WYSIWYG y edición de JavaServer Faces FSF y páginas Facelets, asistencia de código dinámico y una paleta de componentes. Además, al incluir e integrar JBoss Application Server, Developer Studio simplifica el despliegue, la ejecución y la depuración de las aplicaciones Java EE.

1.2.6. Frameworks Utilizados.

Un framework, en el desarrollo de software es una estructura de soporte concretada en la cual un determinado proyecto de software puede afirmarse para su organización y desarrollado. Este en su mayor funcionalidad puede contener soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los componentes de un proyecto. Figura una arquitectura de software que modela las relaciones generales de las entidades del dominio. Suministra una estructura y una metodología de trabajo la cual desarrolla o utiliza las aplicaciones del dominio. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

Framework de la Capa de Presentación.

Framework para la generación de reportes.

Hoy día existen diversos framework para la generación de reportes, pero como sucede siempre resaltan unos más que otros tal como lo son JFreeReport, CrystalReport, BIRT y JasperReport que además de ser de los más populares son de las que más se acercan a cumplir con los objetivos de CICPC.

JasperReports.

JasperReports es ampliamente la presentación de reportes en Java. Presenta un acelerado desarrollo en esta especialidad, informes de alto rendimiento y escalabilidad masiva. JasperReports Professional incluye iReport que es un diseñador de informes gráfico muy fácil de usar y proporciona una cobertura completa de todas las capacidades de JasperReports. Los administradores y diseñadores de informes pueden apreciar la capacidad de utilizar el diseñador para obtener, almacenar y modificar los informes, es el más popular, potente y fácil de usar.

JasperReports en la actualidad es el más popular del mundo en sistemas Java de información gracias a las siguientes características:

- 100% puro Java, lo que asegura la portabilidad.
- Paneles, cuadros, y gráficos
- Pixel-perfecto, complejos diseños de pantalla a imprimir
- Flexibles y extensibles fuentes de datos, dando amplia gama de formatos de salida
- De alto rendimiento
- Solución abierta y basada en standards: Java y XML
- Más grande del mundo, en cuanto a la comunidad más activa de diseñadores y desarrolladores de informes.

Framework para la construcción y diseño de las páginas.

Java Server Faces.

JSF (Java Server Faces) es un marco de trabajo de interfaces de usuario del lado del servidor para aplicaciones Web desarrolladas en Java.

Está compuesto por un API y una implementación para representar componentes UI, manejar su estado, la validación del lado del servidor, conversión de datos y definir la navegación entre las páginas. JSF soporta internacionalización y accesibilidad, proporciona extensibilidad para todas sus características y permite el uso de simples clases java como controladores además de contar con librería de etiquetas para crear componentes dentro de una página JSP.

Este modelo de programación bien definido está regido por el patrón MVC (Modelo Vista Controlador) y facilita de forma significativa la construcción y mantenimientos de aplicaciones Web con UI del lado del servidor.

En la siguiente figura se muestra el ciclo de una petición, donde myUI representa la página creada bajo la tecnología JSF, por otro lado myForm.jsp dibuja los componentes de interface de usuario definido en Java Server Faces.

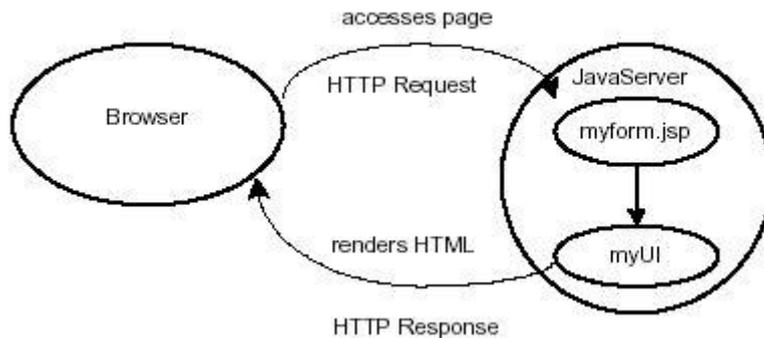


Ilustración 5. Ciclo de vida bajo la tecnología JSF.

Ajax4jsf.

Ajax4jsf es una librería open source integrable fácilmente con la arquitectura de JSF y con el objetivo de extender su funcionalidades, dotándolas de la tecnología Ajax de forma limpia y sin la necesidad de añadir código javaScrip. Mediante este framework se puede variar el ciclo de vida de una petición JSF, realizar peticiones al servidor de forma automáticas así como control de cualquier evento del usuario.

Su funcionamiento es similar a JSF, mediante sus propias etiquetas se generan eventos que envían peticiones al contenedor Ajax. Estos eventos se pueden producir por pulsar un botón, un enlace, una región de la pantalla o sencillamente cada cierto tiempo.

Framework de la Capa de Negocio.

Spring.

Spring es un framework contenedor de código abierto (que puede ser distribuido en un archivo .jar de poco más de un 1 Mega) para la plataforma de java aunque actualmente existe una versión para La plataforma .NET. Pertenece a la categoría de peso ligero por su capacidad de ensamblar componente de diferentes capas de una aplicación; es considerado por su popularidad un sustituto al modelo Enterprise Java Bean.

Sus Inicio se remontan a la versión escrita por Rod Johnson en su libro One-on-One 2003.

Spring soporta la programación orientada a aspecto, técnica que complementa la programación orientada a objeto, logrando dividir el programa en preocupaciones separadas de la lógica del negocio y que se ejecutan de manera transversal a la aplicación; esto evita la dispersión de código o código enmarañado, posibilitando una mejor separación de conceptos y reduciendo la dependencia entre cada uno de los módulos.

Otra de sus características al igual que los frameworks de peso ligero es que implementa el patrón inyección de dependencia mediante la técnica inversión de control, promoviendo el bajo acoplamiento entre los objetos relacionados.

Spring provee un contexto apropiado para el desarrollo de aplicaciones web e integración con frameworks como hibernate (Acceso a Datos), IBATIS (Acceso a Datos), Struts (Presentación), JSF (Presentación), Tapestry (Presentación), JUnit (Test), Asegi (Seguridad).

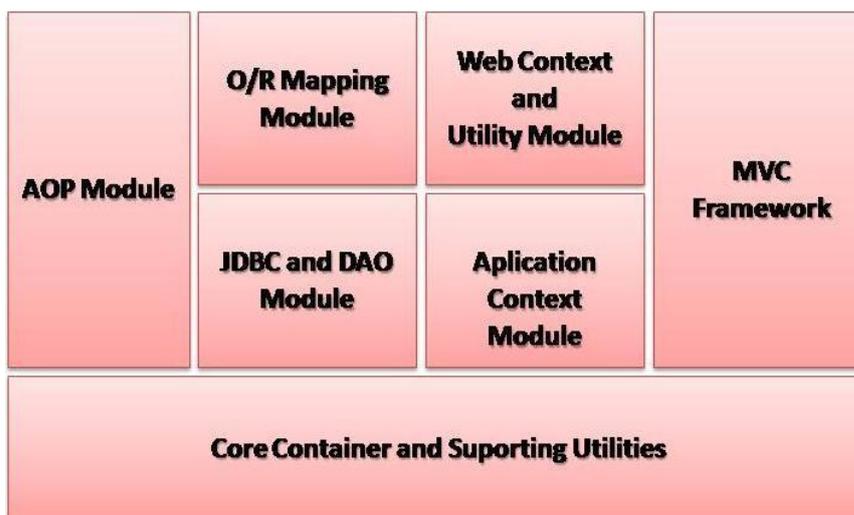


Ilustración 6. Estructura de Spring.

Framework para la construcción de Servicios Web.

Los servicios Web (Web Service en Ingles), representan un conjunto de aplicaciones o tecnologías con el fin de intercambiar datos en la Web. Estos proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones que pueden diferir desde el tipo de sistema operativo hasta el lenguaje donde fue desarrollado, y que al mismo

tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de servicios Web.

Para lograr las características antes plateadas la industria del Software ha adoptado algunos estándares o normas:

1. XML: El formato de datos usado en todas las capas de los Servicios Web.
2. SOAP: (Simple Object Access Protocol) El protocolo predefinido para empaquetar e intercambiar mensajes.
3. WSDL (Web Services Description Language): El idioma que describe los servicios de Web. Aunque basado en XML y entendible por los humanos, WSDL es principalmente para el consumo de la máquina, para ser leído y entendido por los programas del cliente.

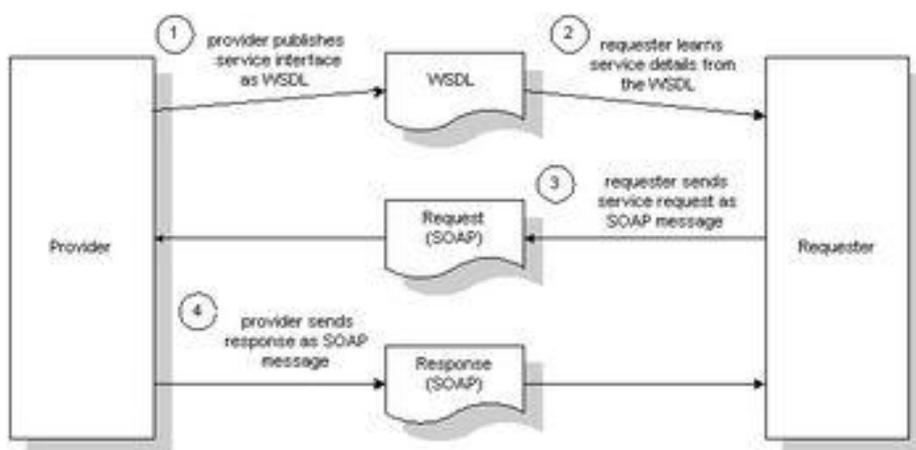


Ilustración 7. Arquitectura básica de un servicio web.

Aquí, Provider es el componente de la aplicación que proporcionaría el servicio, y Requester es el programa del cliente que lo consumiría. Muchas otras tecnologías pueden participar en las interacciones, pero esta figura muestra los componentes que deben estar en un Servicio Web.

XFire.

XFire es framework SOAP open Source, que nos permite implementar este tipo de ambiente con gran simplicidad y brinda muchas características avanzadas de los Servicios Web. Sigue la filosofía POJO por lo que cualquier clase puede ser un servicio Web siendo innecesario extender de alguna clase o implementar alguna interface. Es uno de los framework SOAP más rápidos disponibles y se caracteriza por su robustez; ha estado en desarrollo por alrededor de 5 años siendo utilizado por

muchas organizaciones del mundo. Se considera el de más fácil uso y su buena integración con Spring es otra de sus ventajas.

Framework de la Capa de Acceso a Datos.

Hibernate.

Hibernate es una poderosa herramienta de Mapeo objeto-relaciona Open Source, que al igual que los de su tipo están diseñados para cazar el modelo relacional con el objetual, permitiendo desarrollar clases persistentes siguiendo las pautas de la programación orientada a objeto tal como asociaciones, herencia, polimorfismo, composición y manejo de colecciones.

Entre sus bondades más significativas están:

1. Admite general una base de datos a partir del modelo de clases, para ello tiene compatibilidad con los motores de base de datos más actuales: Oracle, DB2, MySql, Postgres y más, esto se logra especificándole en un archivo XML el driver a usar en dependencia del motor escogido.
2. Mapeo entre las entidades persistentes y las tablas de la base de datos, soportando los diferentes tipos de relaciones y amplia gama de tipos de datos: string, boolean, double, integer, date etc.
3. Posee un lenguaje propio de consultas HQL (Object-Relational Mapping) y incorpora un API mediante el cual se pueden efectuar consultas programáticamente llamadas "criteria"; todo esto sin quitar el uso del SQL nativo.

Todas estas características liberan al desarrollador de las preocupaciones del tipo de Motor de Base de Datos, así como el lenguaje nativo a usar en cada caso, aparte de que facilita trabajar limpiamente en un modelo objetual y nos brinda la flexibilidad de usar SQL en caso que la situación lo permita, en la mayoría de los casos para buscar rendimiento.

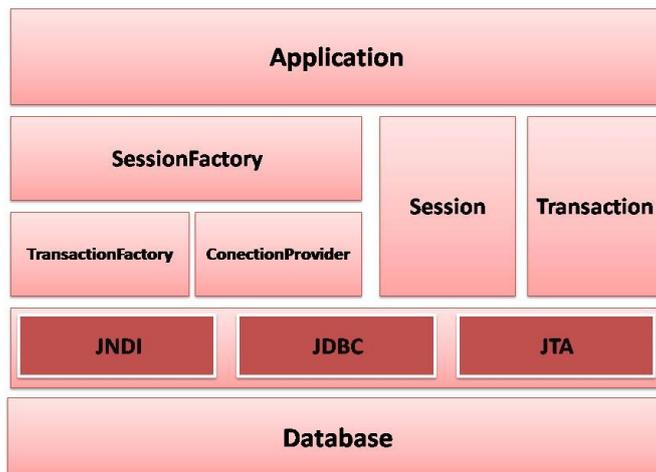


Ilustración 8. Estructura de Hibernate.

Framework auxiliares.

JUnit.

Cuando se prueba un programa, este se ejecuta con unos datos de entrada (casos de prueba) para verificar que el funcionamiento cumple los requisitos esperados. Se define prueba *unitaria* como la prueba de uno de los módulos que componen un programa. En los últimos años se han desarrollado un conjunto de herramientas que facilitan la elaboración de pruebas unitarias en diferentes lenguajes. Dicho conjunto se denomina *XUnit*. De entre dicho conjunto, JUnit es la herramienta utilizada para realizar pruebas unitarias en Java. Existe una gran variedad de razones por la cuales utilizar JUnit ejemplos de esto es que:

- **Los test JUnit te permiten escribir código más rápidamente e incrementan su calidad.**

E s decir cuando escribes test utilizando JUnit, pierdes menos tiempo depurando, y tendrás la confianza de que los cambios de tu código realmente funcionan. Esta confianza te permite ser más agresivo con la refactorización del código y la adición de nuevas características.

- **JUnit es elegantemente simple.**

Escribir test debería ser simple esto es muy importante. Con JUnit, puedes escribir rápidamente los test que ejerciten tu código e incrementalmente añadir test según va creciendo el software. Una vez que has escrito algunos tests, quieres ejecutarlos rápido y frecuentemente sin interrumpir el diseño creativo ni el proceso de desarrollo. Con JUnit, ejecutar test es tan fácil como compilar tu código. El compilador "testea" la sintaxis del código y los test "validan" la integridad del código.

- **Los test JUnit chequean sus propios resultados y proporcionan retroalimentación inmediata.**

Testear no es divertido si se tiene que comparar manualmente los resultados esperados y obtenidos del test, se frena un poco el proceso. Los test JUnit se pueden ejecutar automáticamente y chequean sus propios resultados. Cuando se ejecutan los test, se obtiene una retroalimentación visual de inmediato indicando si se ha pasado o fallado el test. No existe la necesidad de leer un informe para comparar los resultados.

- **Escribir test JUnit no es costoso.**

Utilizando el framework JUnit, puedes escribir test a bajo costo y disfrutar de las conveniencias ofrecidas por el framework. Escribir test es tan simple como escribir un método que pruebe el código que se va a testear y definir el resultado esperado. El framework proporciona el contexto para ejecutar los tests automáticamente y como parte de una colección de otros test. Esta pequeña inversión en testear se continuará beneficiando en tiempo y calidad.

- **Los test JUnit incrementan la estabilidad del software.**

Cuanto menos test escribas, menos estable será tu código. Los test validan la estabilidad del software y te dan la confianza de que los cambios no causarán efectos negativos en el software. Los test forman el pegamento de la integridad estructural del software.

- **¡JUnit es gratis!**

Es decir, JUnit es un conjunto de clases *open source* que nos permiten probar nuestras aplicaciones Java de la forma más simple, rápida y eficiente.

Conclusiones.

Debido al estudio previo realizado por parte de los directivos del proyecto en el que se ha llevado a cabo un análisis detallado de las tecnologías y herramientas más usadas en el campo de la informática, se opta por la decisión de desarrollar una aplicación web, sobre el lenguaje Java, con la integración de los frameworks Java Server Faces (JSF), Spring y Hibernate, lo cual daría como ventaja velocidad de desarrollo, además del uso del patrón n capas con tres niveles, facilitando el desarrollo del producto y disminuyendo el acoplamiento. Para lograr tal resultado se propone por parte de la dirección del proyecto el uso de las herramientas Eclipse y Visual Paradigm por las facilidades que estas brindan. El proceso como tal será orientado por la metodología RUP, la cual constituye una guía de cómo se debe desarrollar una aplicación de tal escala.

CAPITULO 2.

ANÁLISIS Y DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA DE SISTEMA.

En el siguiente capítulo se abordarán los procesos presentes en los sub módulos Denuncia y Control de Investigación y partiendo de los casos de usos respectivos se realizará un análisis y diseño sobre la base de la arquitectura implantada en el proyecto de modernización del CICPC y otros aspectos significativos como el uso de patrones.

2.1. Análisis.

El análisis es la primera representación técnica del sistema. Durante esta parte de flujo, se analiza los requisitos que se describieron anteriormente, refinándolos y estructurándolos. *El objetivo de hacerlo es 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 (3).*

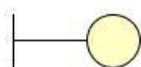
Unas de las tareas que se realizan es el análisis del comportamiento de los requisitos que significa:

1. Identificar las clases de análisis que son necesarias para el cumplimiento de los requisitos, conocidas también como las clases de análisis que intervienen para realizar un caso de uso.
2. Determinar la ubicación de estas clases de análisis dentro de la estructura lógica del sistema. Se refiere a como estas se agrupan en subsistemas de análisis.
3. Diseñar un prototipo de Interfaz de usuario.

Identificar las clases de análisis.

Las clases de análisis representan una abstracción de una o varias clases del diseño. Estas se agrupan en tres estereotipos básicos.

Clase Interfaz.



Las clases Interfaz se utilizan para modelar la interacción entre los actores (usuarios o sistemas externos) y el sistema. Esta interacción está vinculada al traspaso de información desde cualquiera de las partes implicadas. Estas representan

abstracciones de ventanas, formularios, interfaces de comunicaciones, interfaces de impresoras etc.



Clase Controladora.

Son las clases que efectúan el flujo de eventos en la realización de un caso de uso. Estas representan coordinación, secuencia, transacciones y control entre los objetos; son usadas para encapsular la lógica del negocio.



Clase Entidad.

Las clases Entidad se utiliza para modelar información que posee larga vida en el negocio, por lo que a menudo describen abstracciones de una o varias entidades persistentes. Describen objetos del mundo real como personas, objeto etc. Son usadas para almacenar la información.

2.1.1. Entrada al Análisis.

El módulo de Investigación Penal abarca los procesos claves para una investigación (Anexo 1). Estos están distribuidos por tres módulos:

Investigación: Principalmente agrupa los casos de usos referidos al tratamiento de las distintas diligencias penales.

Denuncia: Como su nombre lo indica se centra en los casos de usos referidos a la denuncia.

Control de Investigación: recoge las tareas efectuadas posteriores al comienzo de una investigación y vinculadas al control, asignación etc.

Los casos de usos de estos dos últimos sub módulos se describen brevemente a continuación. Para más información se podrán encontrar las descripciones reducidas así como los diagramas de casos de usos en los anexos del documento (Anexos 2-11).

Denuncia.

Iniciar Investigación: recoge los procedimientos para el inicio de una Investigación Penal.

Comprobar punibilidad del hecho: funcionalidad que permite saber si cierto hecho presenta amparo dentro del código penal como delito.

Comprobar repetición de denuncia: brinda la posibilidad de conocer si cierta denuncia ya se ha realizado.

Control de Investigación.

Aprobar Diligencias Penales: caso de uso con el fin de aprobar o anular las diligencias hechas sobre un acta procesal.

Asignar Reasignar Actas Procesales: caso de uso con el propósito de asignar o reasignar un caso penal.

Concluir Acta Procesal: funcionalidad que tiene como objetivo dar por cerrado un caso investigativo.

Consultar Libro de Control de Investigación: permite hacer consultas simples o avanzadas sobre los libros de control de Investigación.

Gestionar auto de Inicio: caso de uso que agrupa la funcionalidad de ver un auto de inicio e incluir uno nuevo.

Remitir Actas Procesales a fiscalía: caso de uso que permite remitir el caso investigativo a la fiscalía encargada posterior a estar cerrado.

Revisar Diligencias Penales: Caso de uso que permite revisar las diligencias realizadas sobre un acta procesal.

Autorizar Diligencias: caso de usos que permite autorizar las solicitudes de diligencias hechas por los investigadores.

Gestionar Solicitud de Autorización de Diligencia Penal: caso de uso que agrupa las funcionalidades de ver una Solicitud de Autorización de diligencia penal, incluir una nueva, así como modificar una existente.

Transferir Copia de Denuncia SIGEPOL: funcionalidad que permite transferir una denuncia desde SIGEPOL para incluirla dentro de un caso investigativo en el CICPC.

2.1.2. Análisis de casos de usos.

A continuación se muestra el análisis de los casos de usos junto a las realizaciones, que a consideración de los autores del presente documento representan significativamente los sub módulos Denuncia y Control de Investigación.

Caso de Uso : Iniciar investigación.	
Actores	Investigador.
Nivel	Usuario.
Objetivo	Dejar registrado en el sistema el inicio de una investigación.
Resumen	El caso de uso inicia cuando el actor accede al sistema con el objetivo de dar inicio a un caso investigativo. El sistema da la posibilidad de seleccionar la manera en que comienza la investigación, si es por denuncia o de oficio, el actor selecciona la opción correspondiente, en caso que sea por denuncia se permite obtener los datos del denunciante, los datos de la denuncia y además permite gestionar relación de elementos al caso en cualquier momento que se desee. El sistema permite seleccionar la naturaleza del delito, el tipo de delito y los demás datos referentes a la denuncia. El actor introduce los datos referentes a la denuncia y si así lo desea puede realizar modificaciones a la propuesta final, el sistema crea los documentos de denuncia, control de investigación y brinda la posibilidad de imprimirlos.

Ilustración 9. Descripción reducida. Iniciar Investigación.

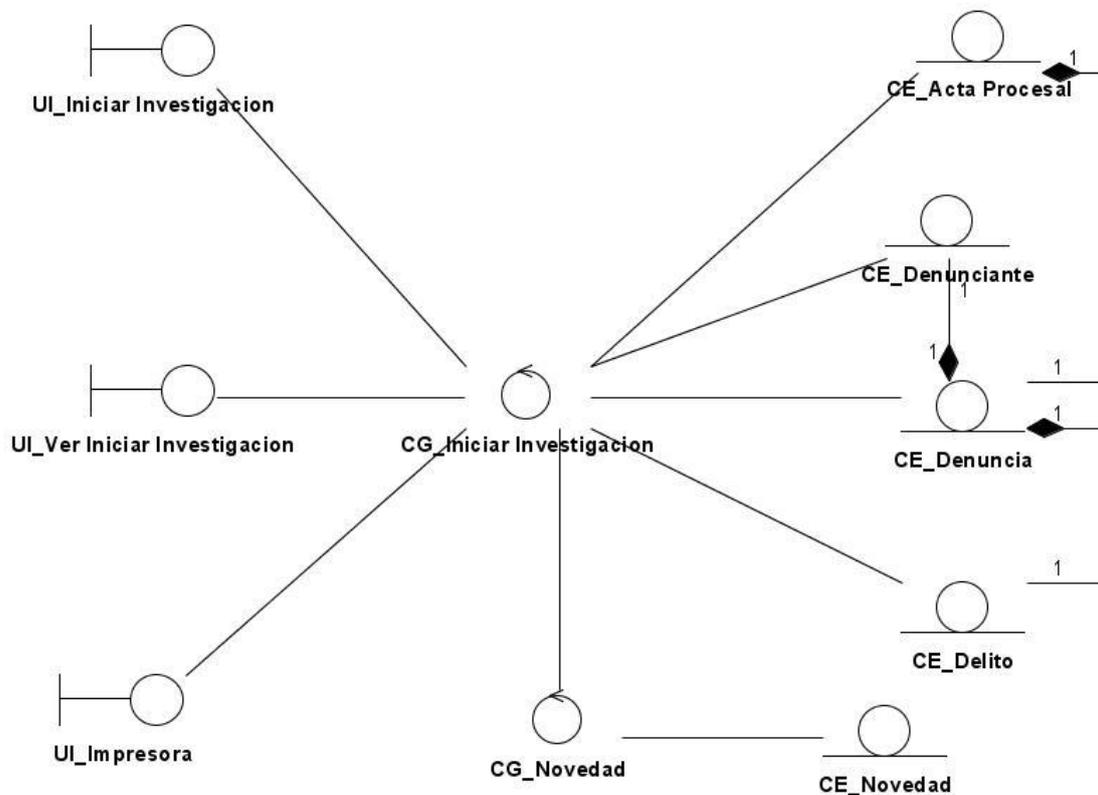


Ilustración 10. Análisis. Iniciar Investigación. Diagrama de Clases

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): DTVE YUBIDI ESMERALDA FLORES DE RODRIGUEZ 11/03/2009 ... 11:53 Inicio Ayuda Salir

AGENDA DE TRABAJO

- Notificaciones
- Asignaciones
- Recepciones
- Aprobaciones
- Remisiones
- Borradores
- Archivos

MENÚ PRINCIPAL

- ▶ Gestión de Despacho
- ▶ Atención Telefónica
- ▶ Investigación Penal
 - Registros
 - Denuncia
 - Notificación de Extravío ...
 - Notificación de Extravío ...
 - Notificación de Amenaza
 - Consultas
 - ▶ Aprehensión
 - ▶ Investigación Criminalística
 - ▶ Análisis de información
 - ▶ Evidencia

Ilustración 11. Interfaz. Iniciar Investigación.

Caso de Uso : Gestionar Auto de Inicio.	
Actores	Investigador.
Nivel	Usuario.
Objetivo	Incluir y ver un Auto de Inicio.
Resumen	El caso de uso se inicia cuando el Investigador selecciona la opción que le permite realizar una acción sobre Auto de Inicio. El actor puede incluir y ver un Auto de Inicio. Si el actor selecciona la opción de incluir un Auto de Inicio, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla. Si el actor elige la opción de ver un Auto de Inicio el sistema mostrará el contenido del Auto de Inicio en cuestión. El sistema permite ver una vista previa de Auto de Inicio, con posibilidad de imprimir y exportar a pdf, terminando así el caso de uso.

Ilustración 12. Descripción reducida. Gestionar Auto de Inicio.

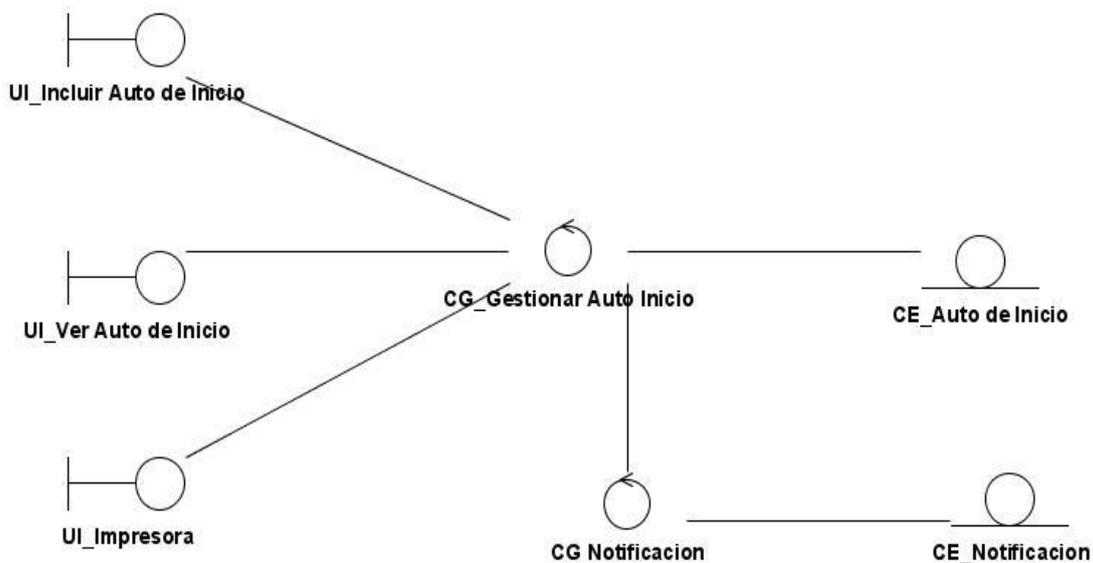


Ilustración 13. Análisis. Gestionar Auto de Inicio. Diagrama de Clases.

The screenshot shows the SIIPOL web application interface. The top navigation bar includes the SIIPOL logo, the user name 'Bienvenido(a): DTVE YUBIDI ESMERALDA FLORES DE RODRIGUEZ', the date '11/03/2009 ... 11:49', and links for 'Inicio', 'Ayuda', and 'Salir'. The left sidebar contains a menu with sections like 'AGENDA DE TRABAJO' and 'MENÚ PRINCIPAL', with 'Investigación Penal' selected. The main content area is titled 'INCLUIR AUTO DE INICIO' and contains a form for entering case details. A table shows 'Datos del Auto de Inicio' with fields for Fecha de Redacción, Hora de Redacción, No. Acta Procesal, Dependencia, Funcionario Redactor, Credencial, Cargo, and Fiscalía. Below this, there are dropdown menus for 'Fiscalía' (set to 'JUZGADO SUPERIOR SEPTIMO EN LO PENAL'), 'Fiscal', 'Fecha de Emisión', and 'Hora de Emisión'. There is a section for 'Diligencias a Realizar' with a 'Tipo Diligencia' dropdown and an 'Observación' text area. At the bottom, there are sections for 'Basamento Legal' with 'Ley' and 'Artículo' dropdowns, and a table for 'De Conformidad con lo establecido en la Ley' with columns for 'Artículo' and 'Eliminar'. 'Incluir' and 'Cancelar' buttons are at the bottom right.

Ilustración 14. Interfaz. Incluir Auto de Inicio.

Caso de Uso : Consultar Libro de Control de Investigación.	
Actores	Investigador.
Nivel	Consultor.
Objetivo	Consultar Libro de Control de Investigación.
Resumen	El caso de uso inicia cuando el actor accede a la opción de consultar el Libro de Control de Investigación, relativo a los casos atendidos en el despacho. El sistema muestra un listado de los casos que ha llevado el despacho, dando la posibilidad de filtrar, ordenar o ver el resumen de una investigación, brindando la posibilidad de imprimir el listado y de exportarlo a pdf, terminando así el caso de uso.

Ilustración 15. Descripción reducida. Consultar Libro de Control de investigación.



Ilustración 16. Análisis. Consultar Libro de Control de Investigación. Diagrama de Clases.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): DTVE, YUBIDI ESMERALDA FLORES DE RODRIGUEZ 11/03/2009 11:55 Inicio Ayuda Salir

AGENDA DE TRABAJO

- Notificaciones
- Asignaciones
- Recepciones
- Aprobaciones
- Remisiones
- Borradores
- Archivos

MENÚ PRINCIPAL

- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
 - Registros
 - Consultas
 - Libro de Control de Inve...
 - Denuncias
 - Punibilidad del Hecho
 - Notificación
 - Notificación por Oficio
 - Aprehensión
 - Investigación Criminalística
 - Análisis de Información
 - Evidencia

LIBRO DE CONTROL DE INVESTIGACIÓN

Criterios de Búsqueda

No. Acta Procesal: Naturaleza del Delito: Desde: Hasta:

Funcionario Actuante

Primer Nombre: Segundo Nombre: Primer Apellido: Segundo Apellido:

Credencial:

Denunciante

Número de Cédula: Primer Nombre: Segundo Nombre: Primer Apellido:

Segundo Apellido:

[Búsqueda Estándar](#)

Ilustración 17. Interfaz. Consultar Libro de Control de Investigación. Búsqueda Avanzada.

Caso de Uso: Gestionar Solicitud de Autorización de Diligencia Penal.	
Actores	Investigador.
Nivel	Usuario.
Objetivo	Incluir, ver o modificar una Solicitud de Autorización de Diligencias Penales en el Acta Procesal.
Resumen	El caso de uso inicia cuando el actor selecciona la opción que le permite realizar una acción sobre una Solicitud de Autorización de Diligencias Penales. El actor puede incluir, ver y modificar una Solicitud de Autorización de Diligencias Penales. En caso de que seleccione la opción de incluir Solicitud de Autorización de Diligencias Penales para un listado de diligencias penales, el sistema mostrará un listado de las posibles diligencias penales que puede solicitar el investigador (aquellas que no tenga autorizadas ya para ese caso), dando la posibilidad de seleccionar una o más diligencias y de introducir una nueva diligencia, distinta a las predeterminadas, así como brindará la posibilidad de introducir el resto de los datos de la Solicitud de Autorización de Diligencias Penales. Si el actor selecciona la opción de ver la Solicitud de Autorización de Diligencias Penales, el sistema muestra los datos de la misma brindando la posibilidad de imprimir o exportar a PDF. Si el actor selecciona la opción de modificar una Solicitud de Autorización de Diligencias Penales, el sistema muestra un listado de las diligencias penales solicitadas por el investigador, dando la posibilidad de seleccionar más diligencias o deseleccionar las que ya están, así como de introducir una nueva diligencia. El sistema mostrará además los datos que pueden ser editables dentro de Solicitud de Autorización de Diligencias Penales, y una vez realizados los cambios, guardará las modificaciones, terminando así el caso de uso.

Ilustración 18. Descripción reducida. Gestionar Solicitud de Autorización de Diligencia Penal.

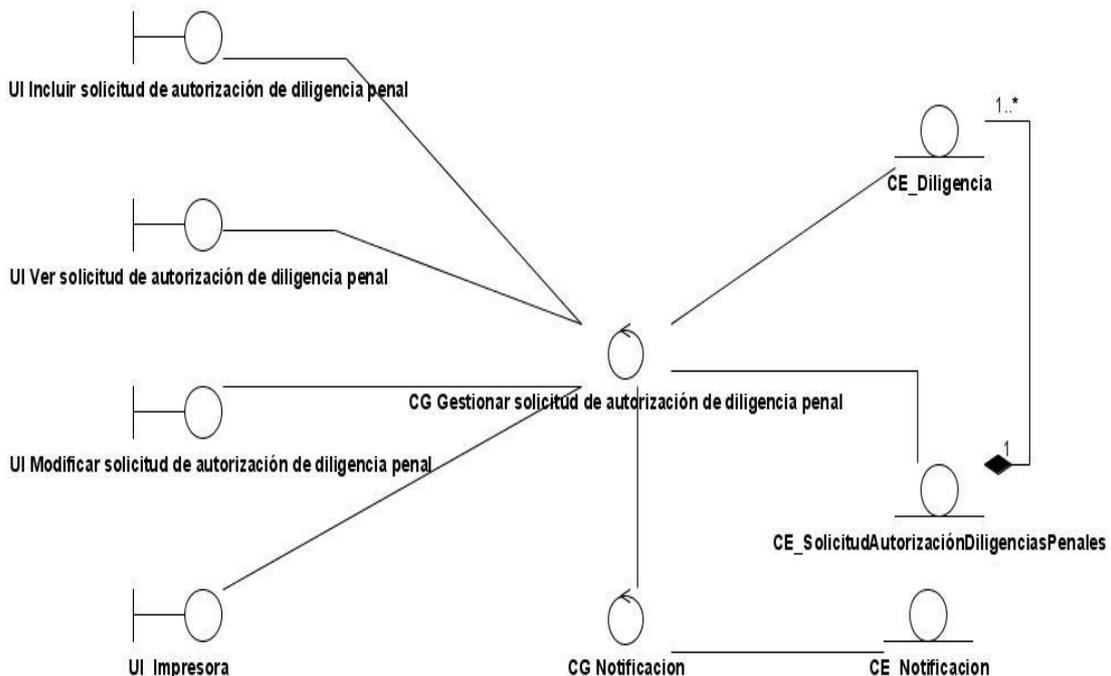


Ilustración 19. Análisis. Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): DTVE YUBIDI ESMERALDA FLORES DE RODRIGUEZ 11/03/2009 11:57 Inicio Ayuda Salir

AGENDA DE TRABAJO

- Notificaciones
- Asignaciones
- Recepciones
- Aprobaciones
- Remisiones
- Borradores
- Archivos

MENÚ PRINCIPAL

- ▶ **Gestión de Despacho**
- ▶ **Atención Telefónica**
- ▶ **Investigación Penal**
 - Registros
 - Consultas
 - Libro de Control de Inve...
 - Denuncias
 - Punibilidad del Hecho
 - Notificación
 - Notificación por Oficio
- ▶ **Aprehensión**
- ▶ **Investigación Criminalística**
- ▶ **Análisis de Información**
- ▶ **Evidencia**

INCLUIR SOLICITUD DE AUTORIZACIÓN DE DILIGENCIA PENAL

Solicitud de Autorización de Diligencia Penal

Fecha de Redacción	miércoles, 11/03/2009
Hora de Redacción	11:56
No. Acta Procesal	K-08-0051-00002
Dependencia	Sub-Delegación Simón Rodríguez
Funcionario	DTVE YUBIDI ESMERALDA FLORES DE RODRIGUEZ
Credencial	0018486
Cargo	Jefe de la Sala Técnica
Fiscalía	Fiscalía 27 Área Metropolitana
Fiscal	JURKY COROMOTO LINARES QUINTERO

Diligencias Penales

Tipo de Diligencias

Justificación

Listado de Diligencias Penales

No	Tipo de Diligencia	Justificación	Eliminar
No existen registros			

[Personalizar Comunicación](#)

Ilustración 20. Incluir Solicitud de Autorización de Diligencia Penal.

Caso de Uso: Transferir copia de Denuncia policial SIGEPOL	
Actores	Solicitante.
Nivel	Subfunción.
Objetivo	Incluir en el Acta Procesal una copia de una Denuncia que se encuentra en SIGEPOL.
Resumen	El caso de uso inicia cuando el actor accede a la opción de solicitar la transferencia de una Denuncia que se encuentra en SIGEPOL. El sistema brinda la posibilidad de introducir el número del caso de la otra policía que está asociado a la Denuncia. El actor introduce el número del expediente de la otra policía. El sistema consulta el SIGEPOL y obtiene la Denuncia. El sistema muestra una vista de la Denuncia y la incluye al Acta Procesal seleccionada, brindando la posibilidad de imprimirla y exportarla a PDF, dando fin al caso de uso.

Ilustración 21. Descripción reducida. Transferir copia de Denuncia policial SIGEPOL.

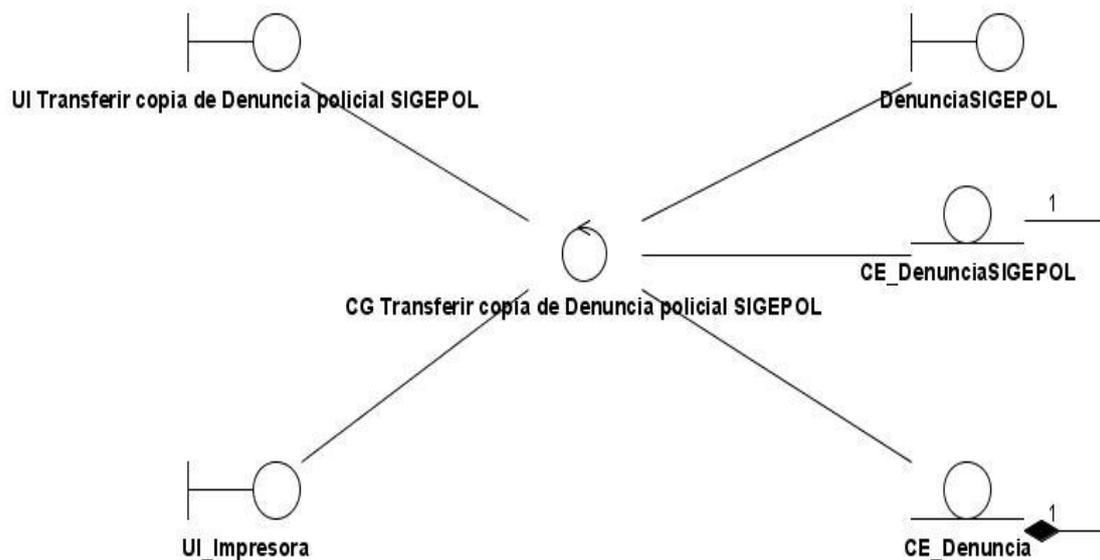


Ilustración 22. Análisis. Transferir copia de Denuncia policial SIGEPOL. Diagrama de Clases.

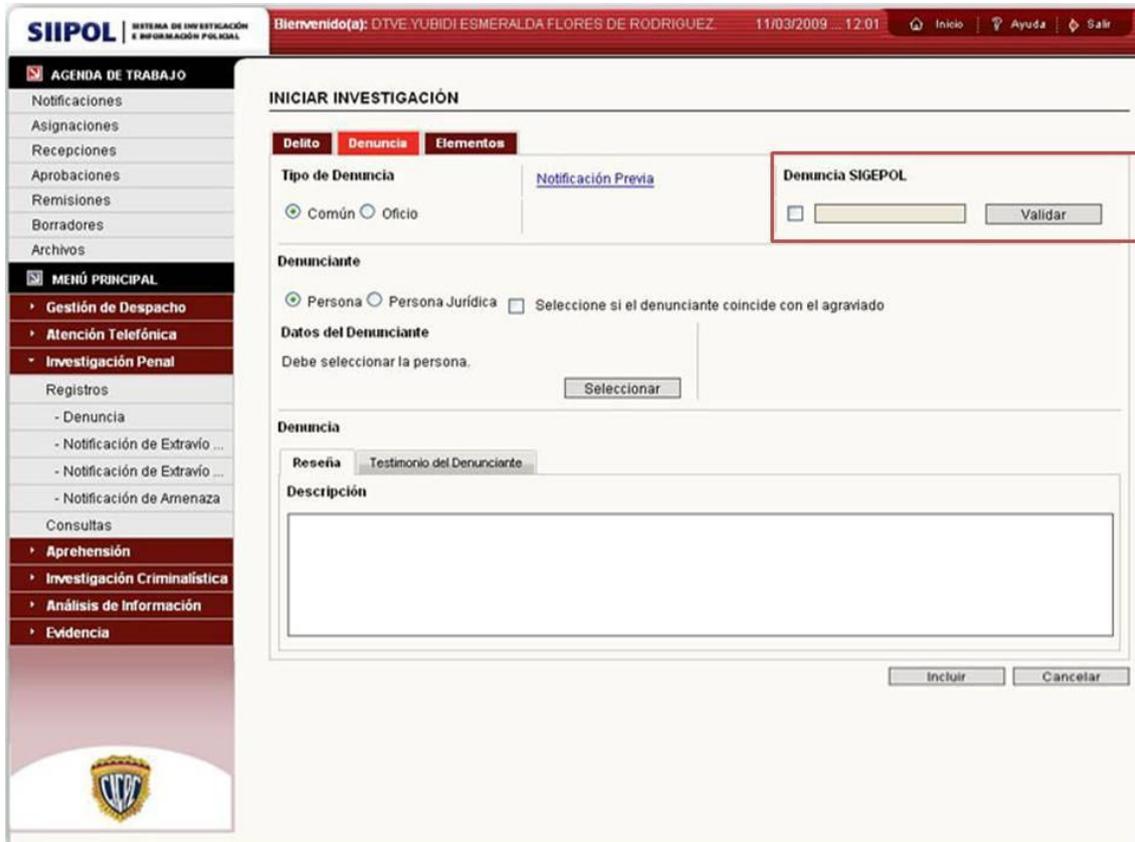


Ilustración 23. Interfaz. Transferir copia de denuncia policial SIGEPOL.

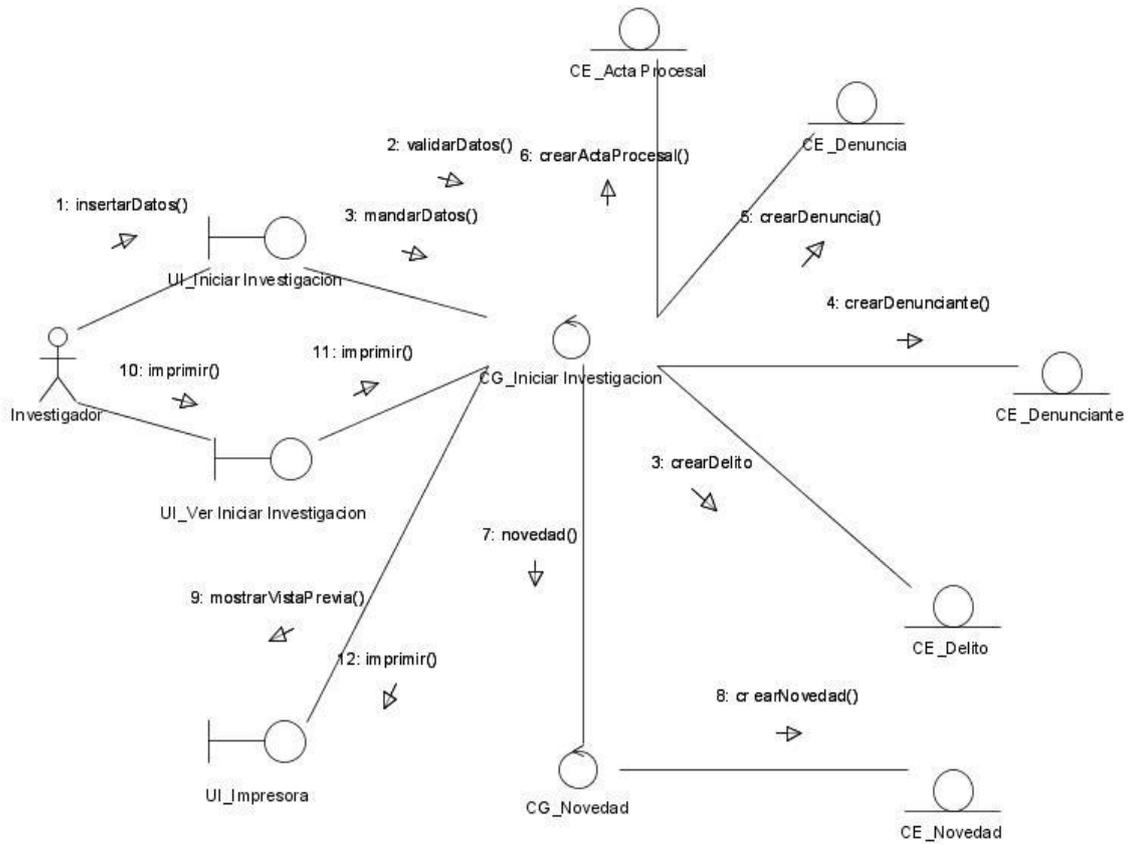


Ilustración 24. Análisis. CU Iniciar Investigación. Diagrama de Colaboración.

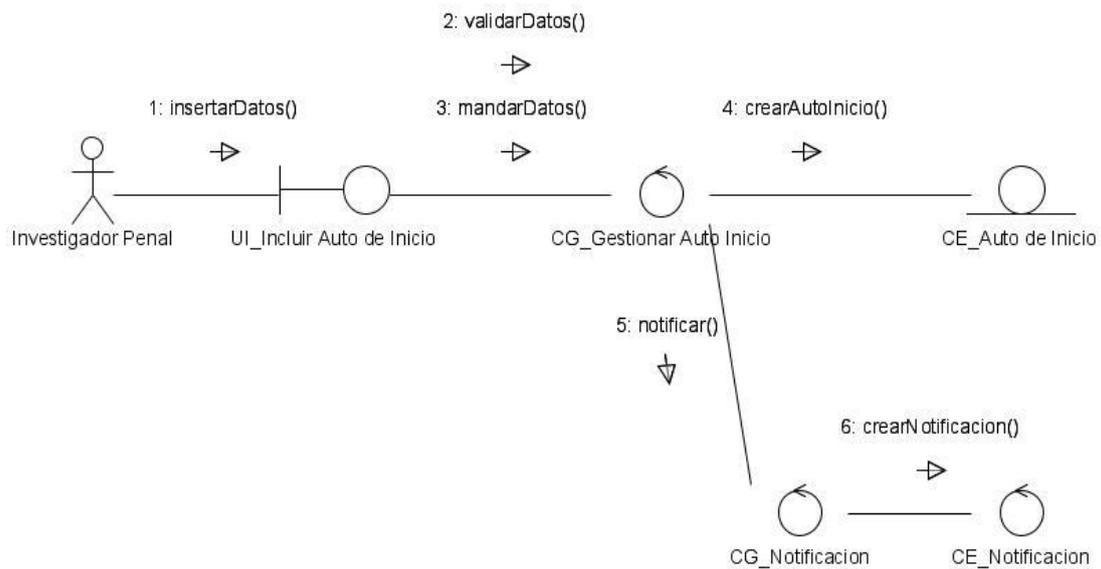


Ilustración 25. Análisis. CU Gestionar Auto de Inicio. Diagrama de Colaboración. Escenario Incluir.

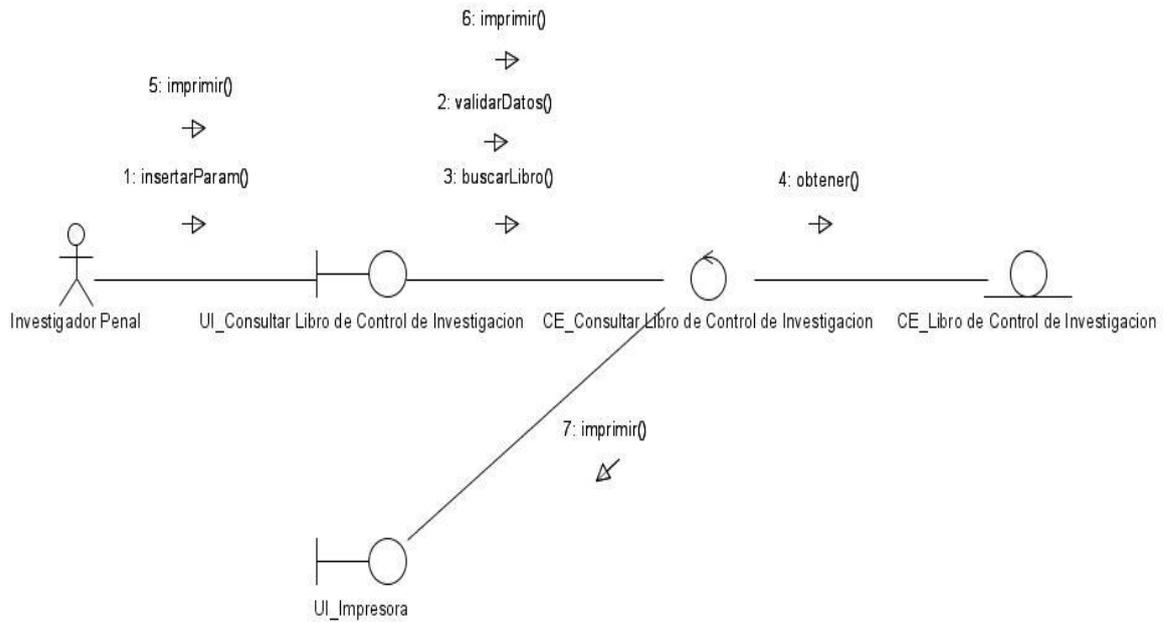


Ilustración 26. Análisis. CU Consultar Libro de Control de Investigación.

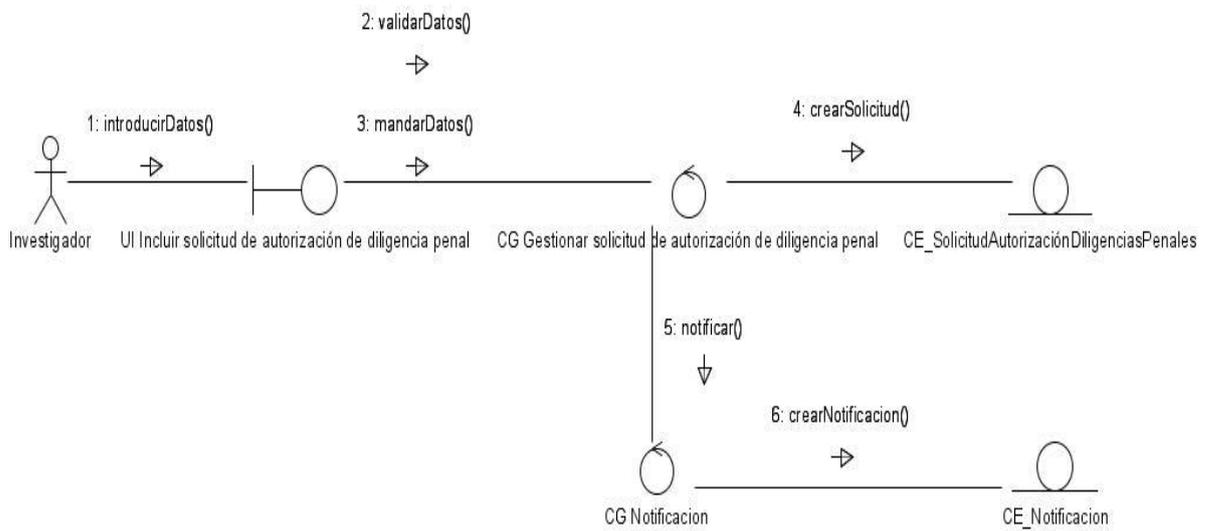


Ilustración 27. Análisis. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Colaboración. Escenario Incluir.

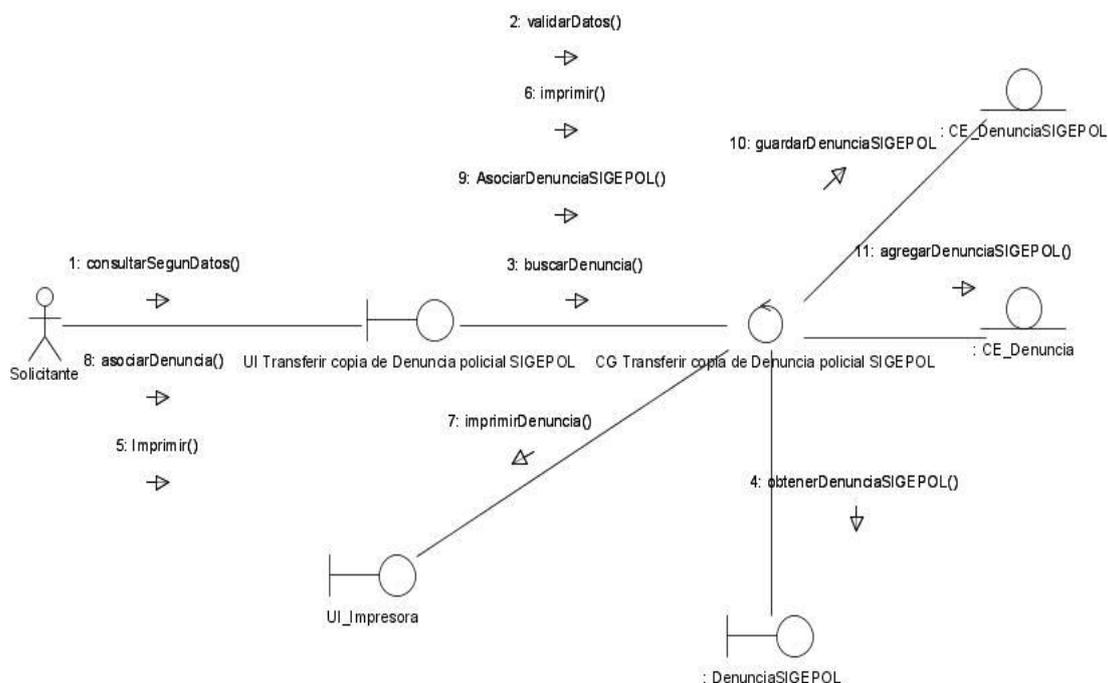


Ilustración 28. Análisis. CU Transferir copia de denuncia policial SIGEPOL. Diagrama de Colaboración.

2.2. Diseño.

El diseño es una representación significativa de ingeniería de algo que se va a construir (3). El objetivo es producir un modelo o representación del sistema que será implementado. Como macro actividad posterior al análisis esta se encarga de refinarlo por lo que este debe ser suficiente para el sistema pueda ser erigido sin ambigüedades. En este apartado cobra especial atención los patrones de diseño representando soluciones a problemas comunes en el desarrollo del software.

2.2.1. Aspectos significativos del Diseño.

El diseño de la solución de los sub módulos Denuncia y Control de Investigación se encuentra regido por una arquitectura común para todo el proyecto; esta, como se menciona anteriormente es de n-capas de tres niveles la cual ofrece desarrollo en paralelo, una aplicación más robusta debido al encapsulamiento, flexibilidad, escalabilidad y una mayor facilidad de mantenimiento y soporte.

Se propuso por parte de los arquitectos de la solución para la modernización del SIIPOL la estructura modular que se muestra a continuación.

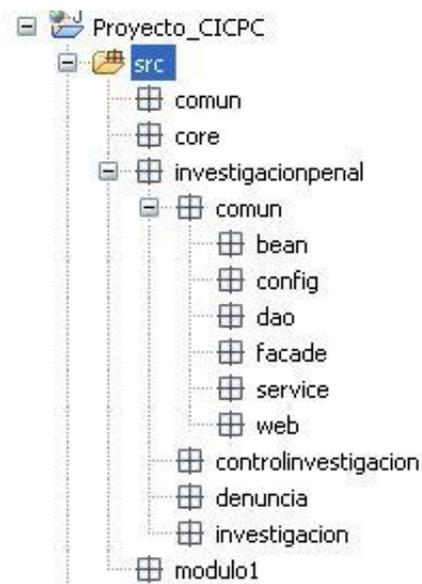


Ilustración 29. Estructura Modular de la Solución.

En cada módulo se encuentra un paquete común que almacenaría todo lo frecuente para el módulo en cuestión, igual papel jugaría el paquete común que se encuentra en la raíz de la carpeta de código del proyecto (src) que acumularía todo lo común para el proyecto. Todos los módulos, dígame sub módulos y el respectivo paquete común tienen una estructura conformada por:

config: almacena las configuraciones pertinentes de los diferentes framework.

bean: se agrupan las entidades.

dao : agrupa las clases encargadas del trabajo con la base de datos.

service: almacena las clases responsables de ejecutar la lógica del negocio.

facade: almacena la interfaz que sirve como fachada del módulo.

web: agrupa las clases para el trabajo con la presentación como beans manejados, convertidores, validadores y demás clases útiles.

2.2.2. Clases significativas de la Arquitectura.

Entre las clases que afectan directamente el diseño de los respectivos sub módulos y separándolas por los paquetes de incidencia están:

bean.

La clase EntidadPersistenteBase de la cual heredan todas las clases persistentes.

dao.

La clase DaoGenericoImpl que encapsula todo el trabajo con el framework de persistencia hibernate y de la cual heredan todas las clases con la responsabilidad de trabajar con la base de datos.

web

La clase BaseBean del cual heredan todos los *beans* manejado para el trabajo con jsf, esta clase agrupa funcionalidades afines como la navegación entre páginas, muestra de mensajes etc. La interfaz EntityViewer diseñada para ser implementada por aquellos beans de manejo de JSF que se encarguen de mostrar los datos de una entidad. La clase BeanBaseInvPenal que agrupa funcionalidades comunes de los beans del módulo de Investigación Penal, esta hereda de la clase BaseBean.

2.2.3. Patrones.

Para la construcción de dicha solución como se explica en el capítulo anterior se usaron diferentes framework. Dichos framework incorporan muchos patrones como es el caso de Jsf con el patrón arquitectónico MVC (Modelo Vista Controlador), que separa los datos, la interfaz de usuario y la lógica del control en tres elementos diferentes permitiendo construir aplicaciones más grandes así como fácil de modificar y mantener debido a la clara separación de tareas.

En el caso de Spring se puede mencionar algunas soluciones de diseño como el patrón Inversión de Control logrando que un objeto en cuestión defina sus propias dependencias, este patrón es una alternativa para enfrentar la creciente complejidad en la construcción de soluciones, a las cuales se les exige cada vez más flexibilidad y adaptabilidad a las cambiantes situaciones de la industria. Cabe mencionar otros patrones aunque conocidos como el Singleton que se encarga de asegurarse que el número de objetos creados sea controlado, en la práctica se usa cuando se quiere tener una instancia única de una clase a la cual se aplica y el Prototype que es exactamente lo contrario del Singleton.

En Hibernate, el framework encargado de la persistencia es importante reflejar el uso del patrón proxy, con la finalidad de tener una representación de un objeto, específicamente se utiliza cuando se van a obtener objetos de las base de datos y estos tienen relación con otros objetos, las relaciones se representan por medio de proxy para evitar cargar datos que no serán utilizados por el momento.

Entrando en el desarrollo, se reflejan otros patrones como ServiceLocator que consiste en utilizar un objeto Service Locator para abstraer toda la utilización JNDI y para ocultar las complejidades de la creación del contexto inicial, este es usado con frecuencias en la solución presentada para obtener una referencia a la implementación de las fachadas.

Uno de los patrones GOF usados es el Facade con el fin Proveer una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema, sobre la base de esto están construidas por cada módulo una clase interfaz que unifique todos los servicios que puede brindar dicho subsistema.

El acceso a datos está regido por el patrón DAO (data acces object) que consiste en utilizar un objeto de acceso a datos para abstraer y encapsular todos los accesos a la fuente de datos.

Además de los patrones explicados existen otros que pueden ser de interés para el lector y que para abordar sobre ellos pueden consultar la bibliografía adjunta.

Otros patrones usados:

Observer: Define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.

Transaction: agrupa una colección de métodos de forma que todos ellos finalicen correctamente o si alguno de ellos falla, todos fallen a la vez.

Composite view: Un objeto vista que está compuesto de otros objetos vista, es usado por ejemplo cuando se quiere incluir un subview .jsp dentro de una página .jsp.

Front Controller: Un objeto que acepta todos los requerimientos de un cliente y los direcciona a manejadores apropiados, es usado en la capa de presentación.

Abstract Factory (Fábrica abstracta): Permite trabajar con objetos de distintas familias de manera que las familias no se mezclen entre sí, es usado por el framework spring.

Builder (Constructor virtual): Abstrae el proceso de creación de un objeto complejo, centralizando dicho proceso en un único punto, es usado por el framework spring.

Bridge (Puente): Desacopla una abstracción de su implementación. Es el principio seguido para todas las implementaciones con respecto a las interfaces de la presente solución.

2.2.4. Diagrama de clases del diseño.

A continuación se muestra el diseño de los casos de usos escogidos por los autores del presente documento. Estos son los mismos a los cuales se les realizaron el análisis anteriormente.

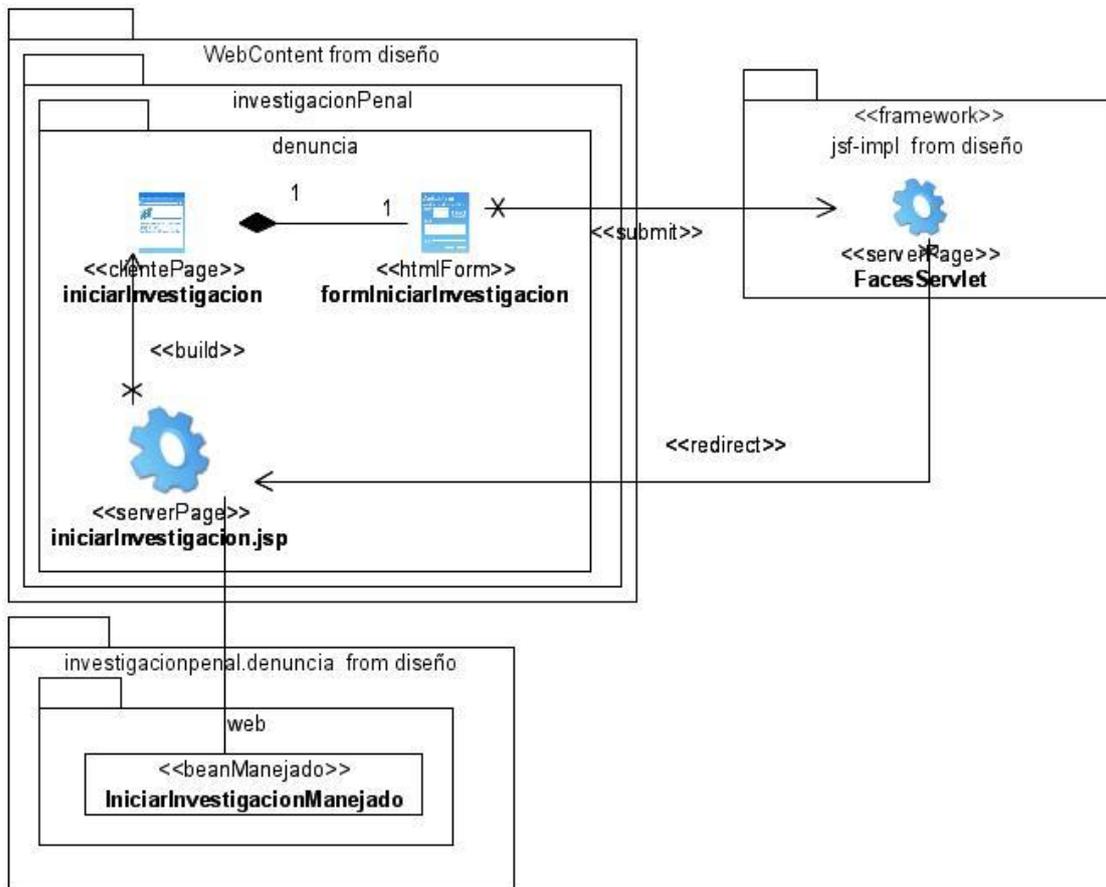


Ilustración 30. Diseño. CU Iniciar Investigación. Diagrama de Clases. Parte I.

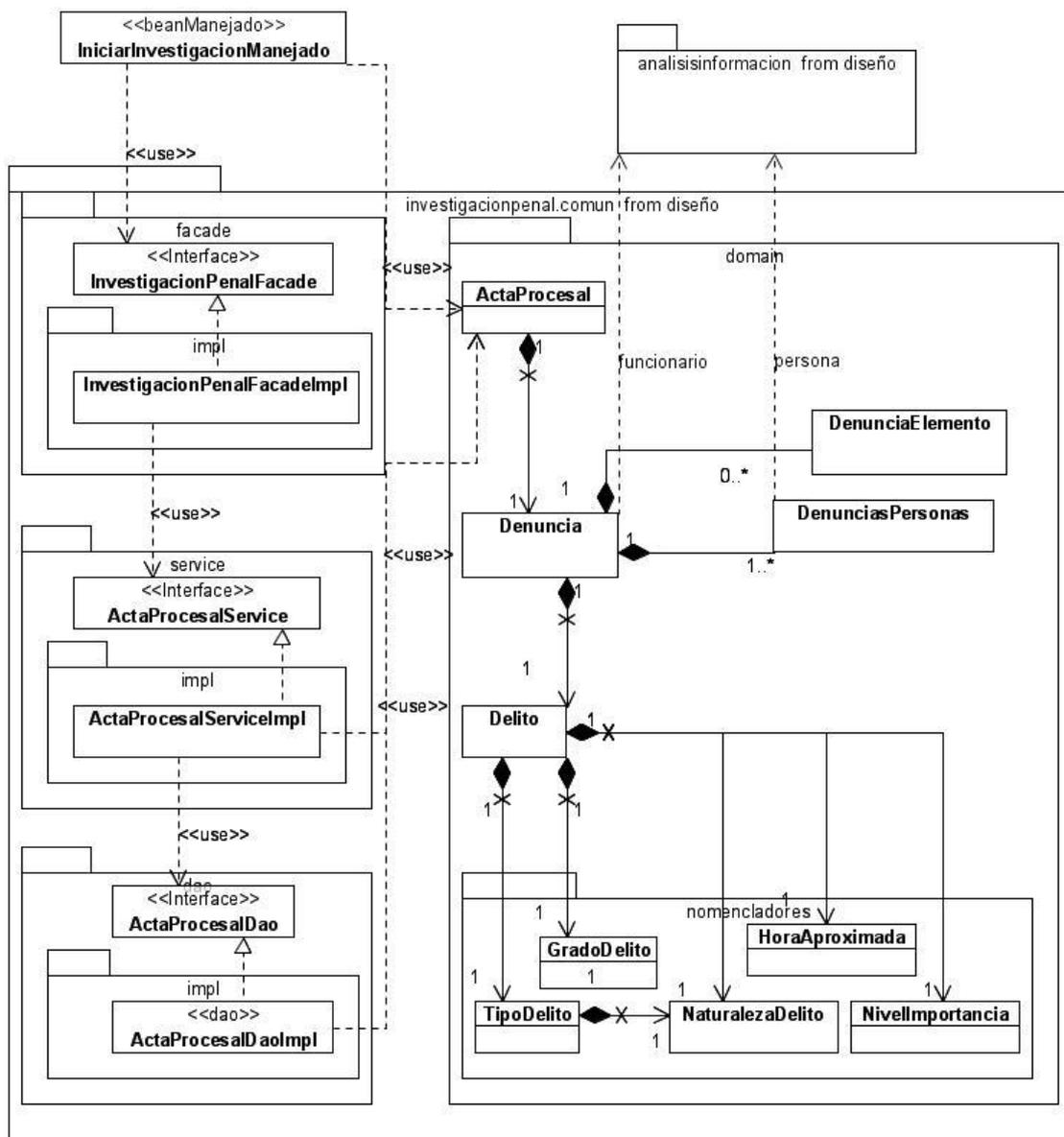


Ilustración 31. Diseño. CU Iniciar Investigación. Diagrama de Clases. Parte II.

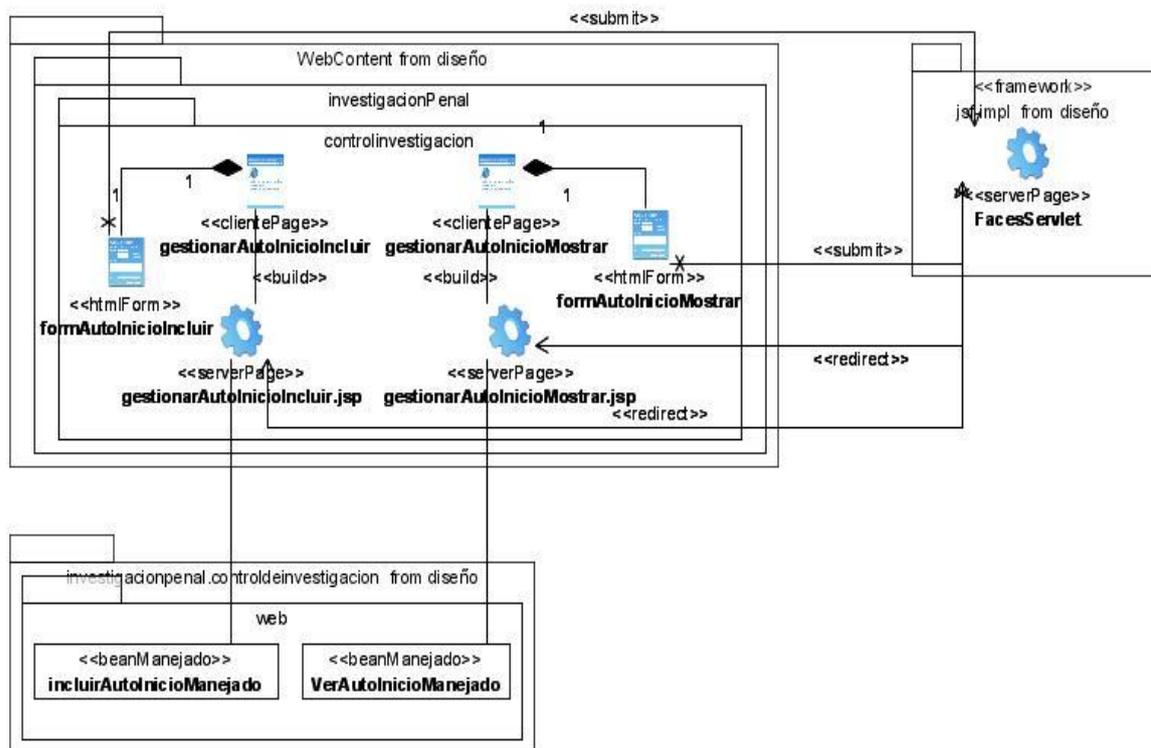


Ilustración 32. Diseño. CU Gestionar Auto de Inicio. Diagrama de Clases. Parte I.

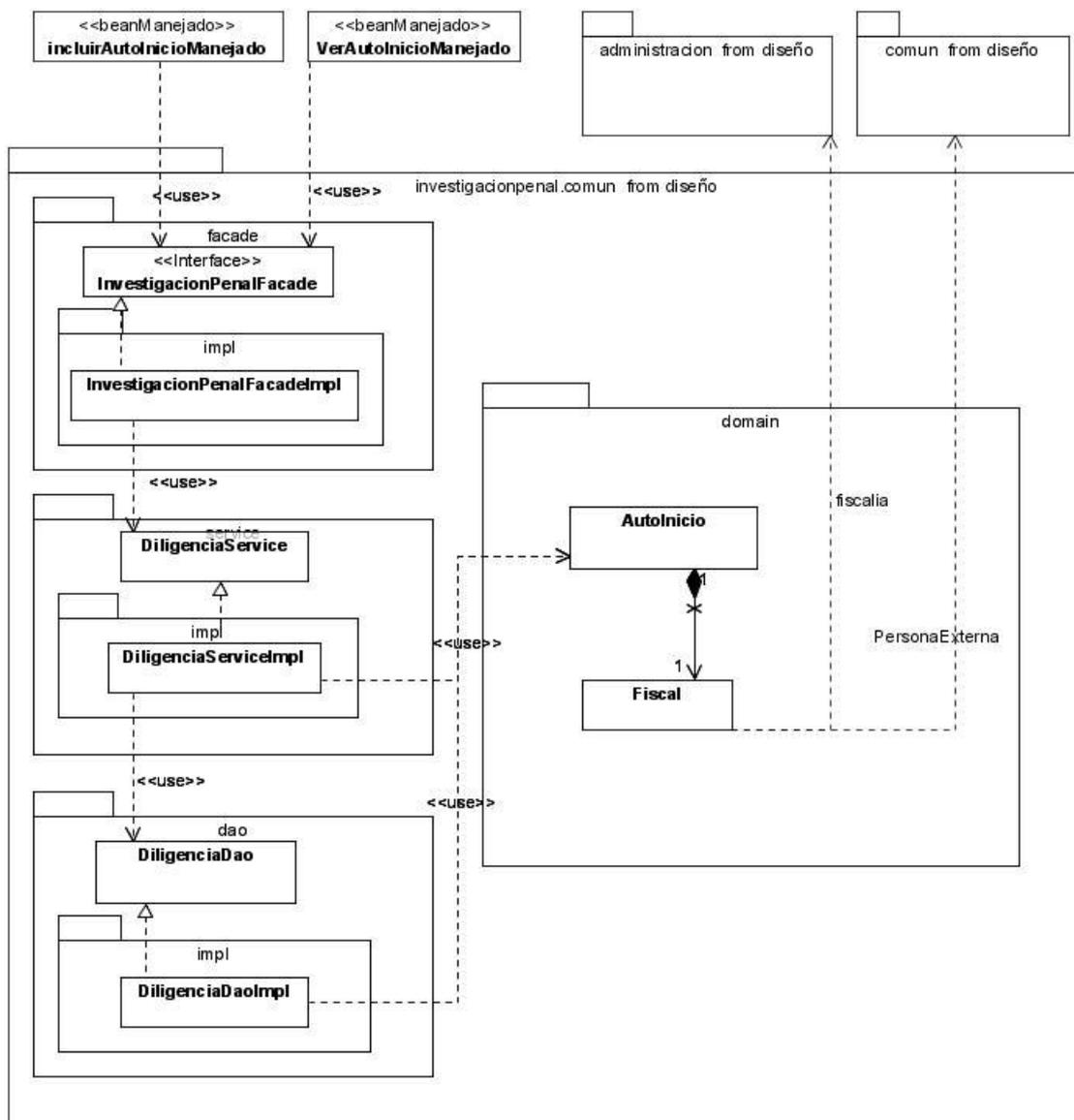


Ilustración 33. Diseño. CU Gestionar Auto de Inicio. Diagrama de Clases. Parte II.

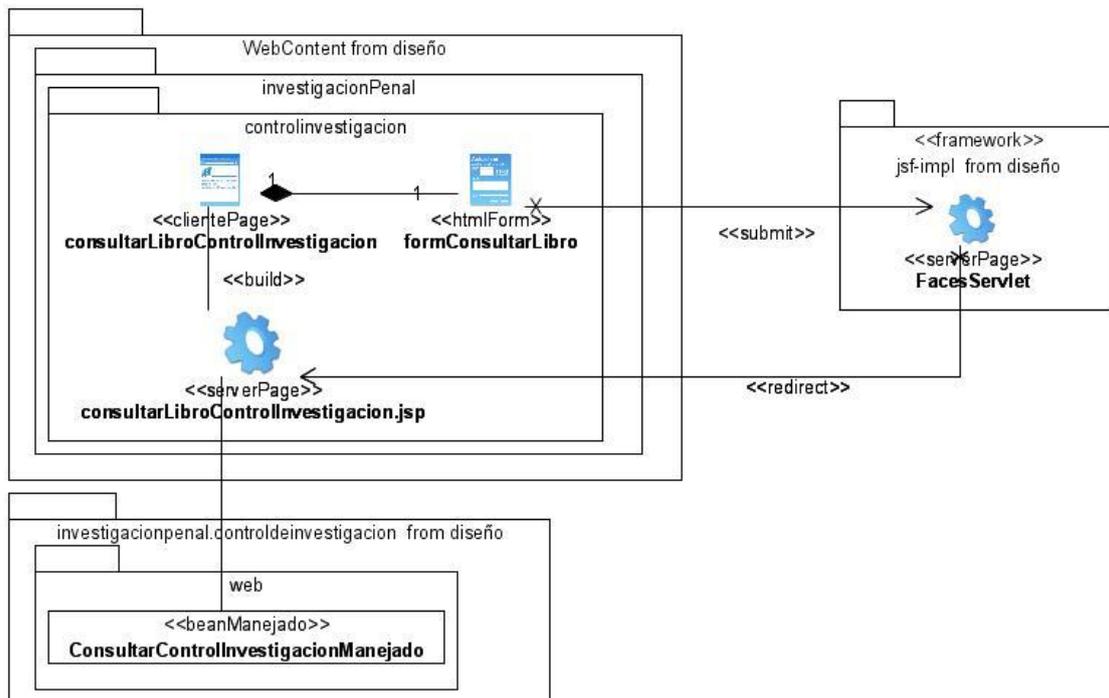


Ilustración 34. Diseño. CU Consultar Libro de Control de Investigación. Diagrama de Clases. Parte I.

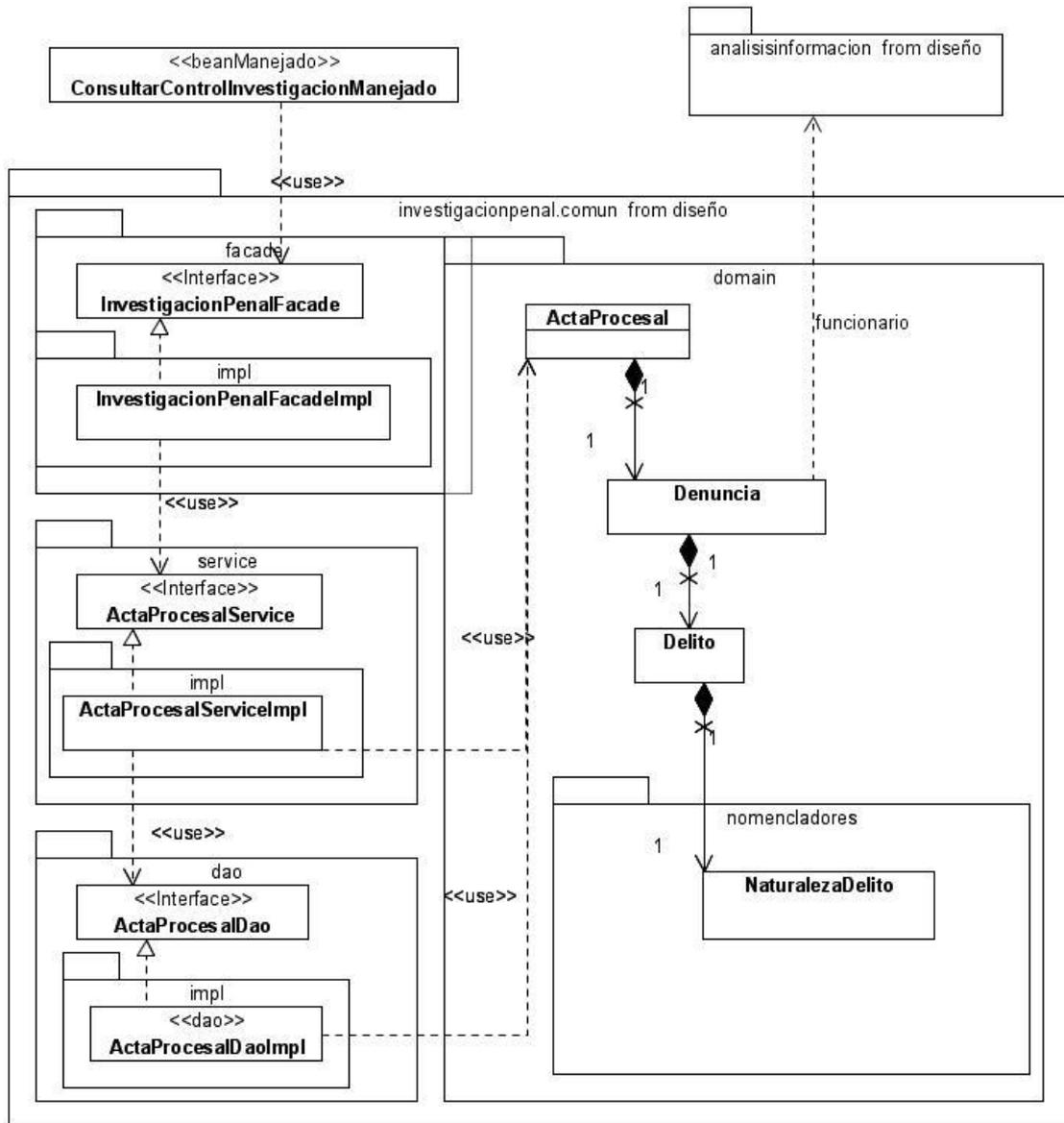


Ilustración 35. Diseño. CU Consultar Libro de Control de Investigación. Diagrama de Clases. Parte II.

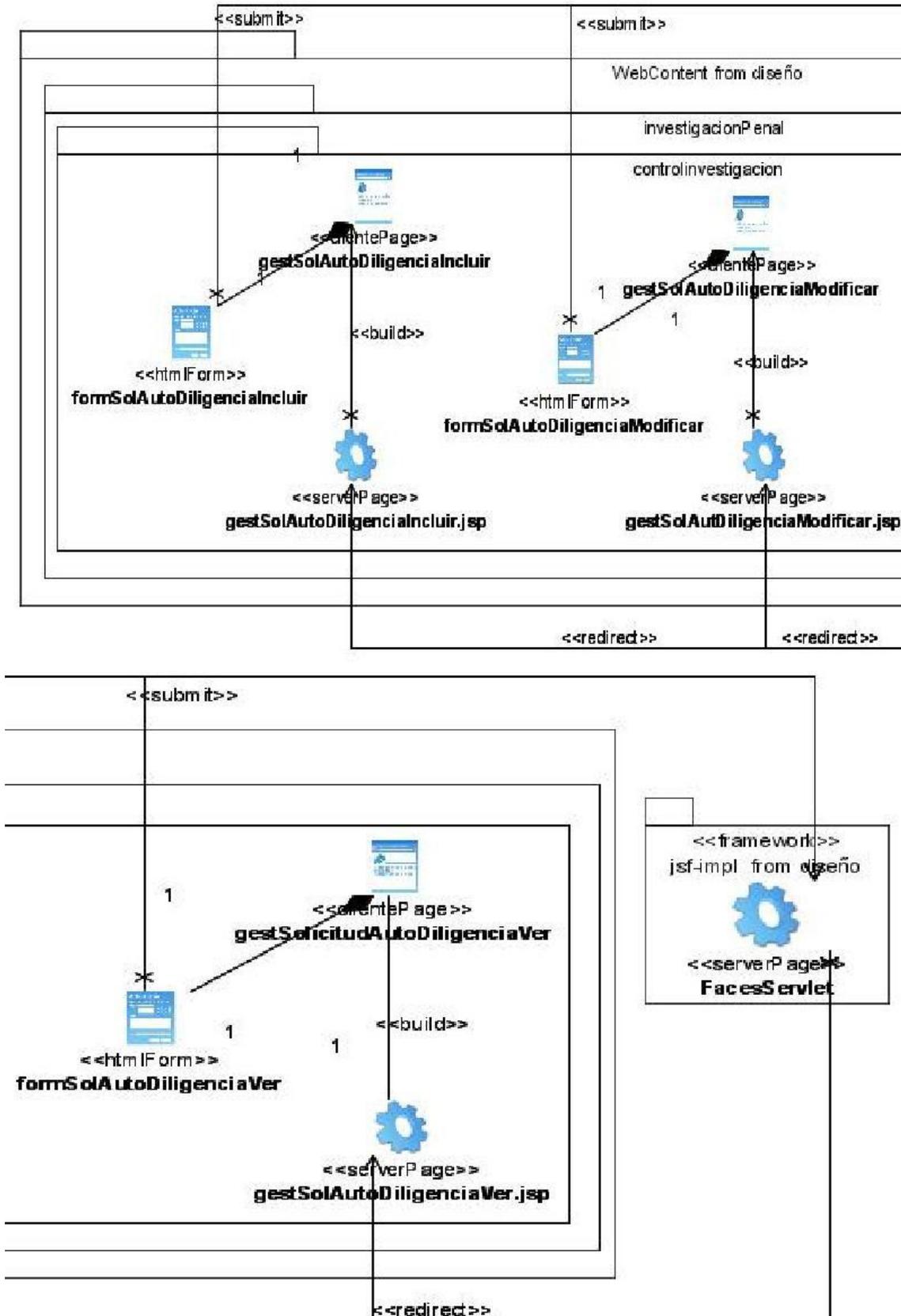


Ilustración 36. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases. Parte I.

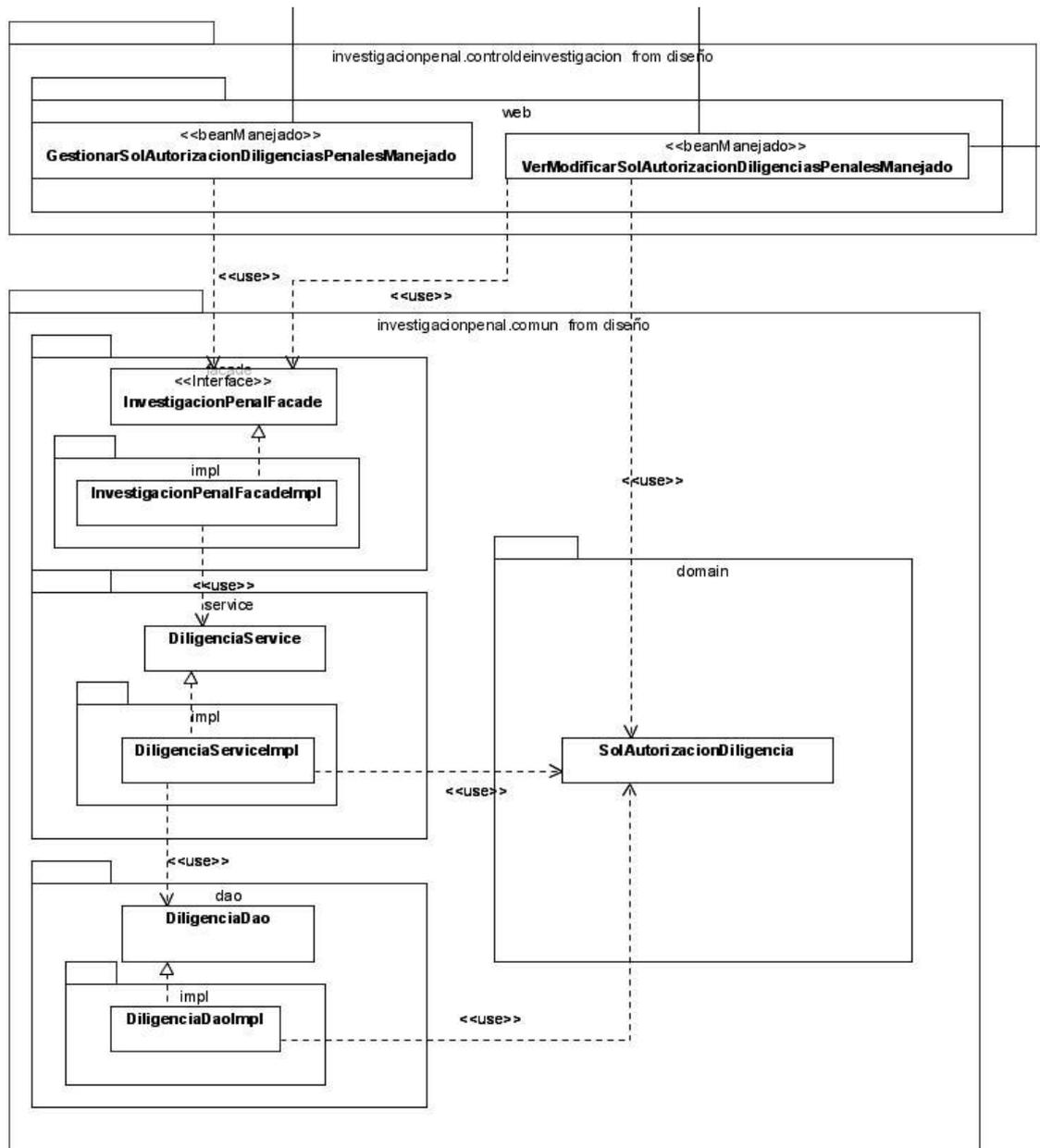


Ilustración 37. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Clases. Parte II.

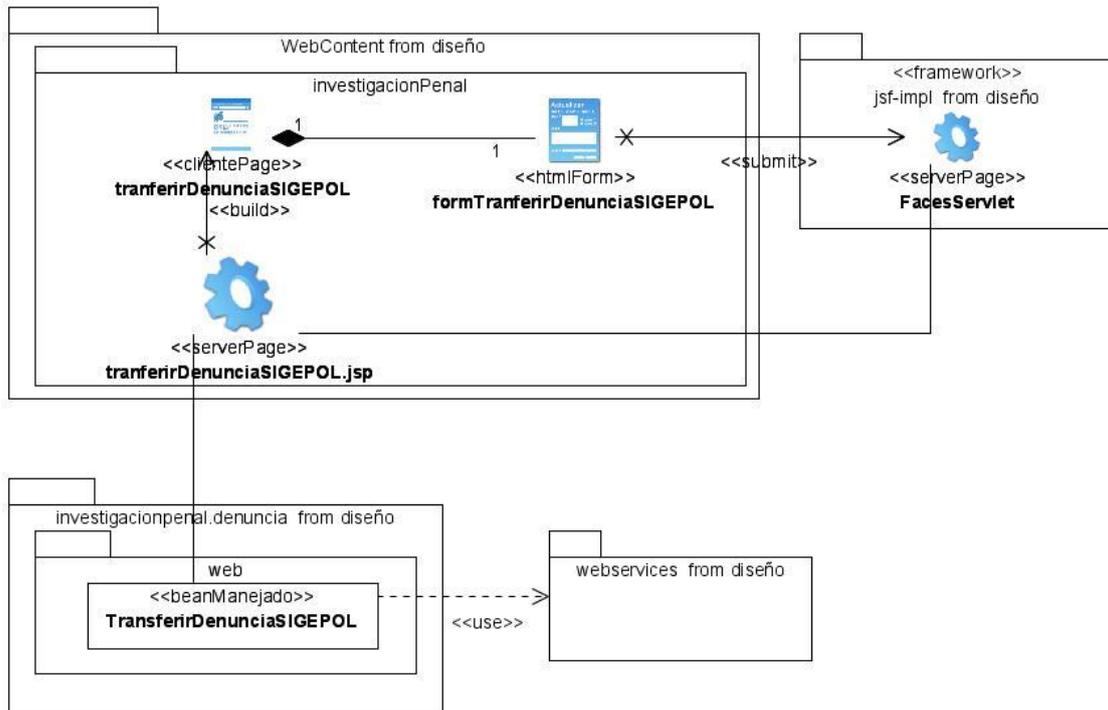


Ilustración 38. Diseño. Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Clases. Parte I.

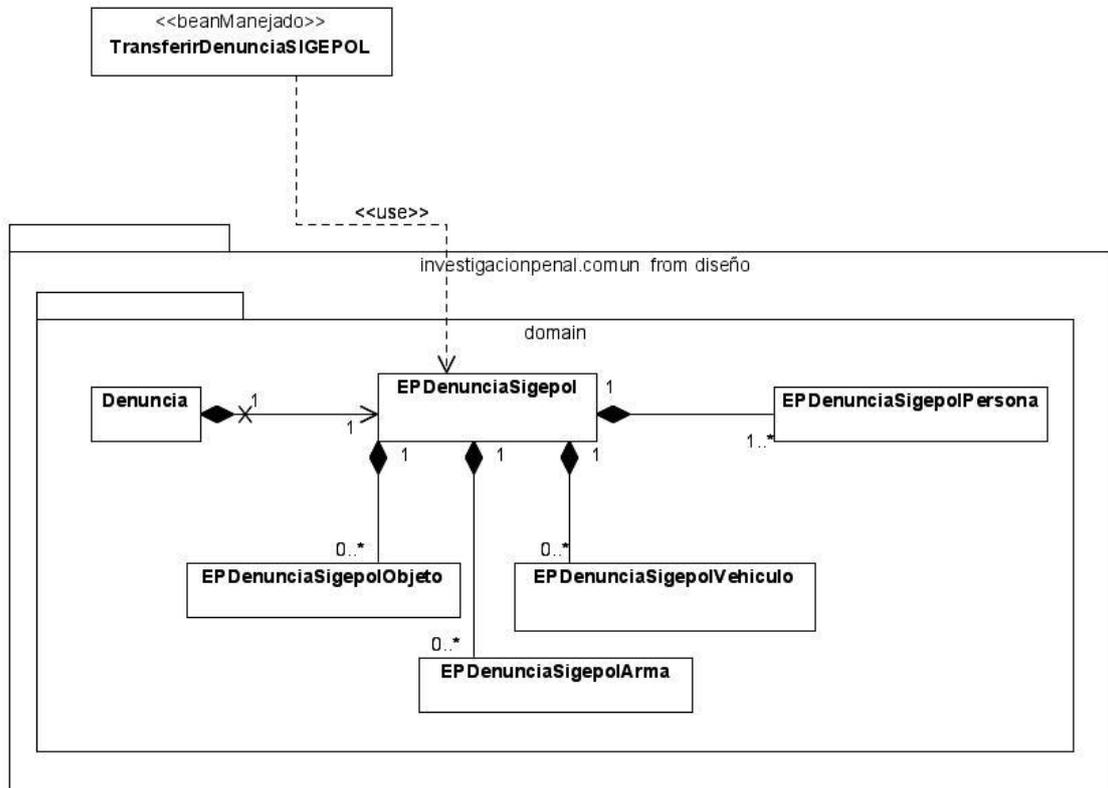


Ilustración 39. Diseño. Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Clases. Parte II.

2.2.5. Clases significativas propias de la solución.

A continuación se describen algunas de las clases más relevantes para la solución.

Nombre: ActaProcesal	
Tipo de clase: Bean Persistente que lleva consigo todo lo referente a un Caso Penal	
Atributo	Tipo
hora	Date
denuncia	Denuncia
fiscal	Fiscal
actaProcesalElemento	Set<ActaProcesalElemento>
actaProcesalPersona	Set<ActaProcesalPersona>
estado	EstadoActaProcesal
actaProcesalEstacionamiento	Set<ActaProcesalLugarEstacionamiento>
fiscalia	Fiscalia
historico	Boolean
Para cada responsabilidad:	
Nombre:	addActaProcesalPersona
Descripción:	Adicionar una nueva ActaProcesalPersona
Nombre:	obtenerTipoRelacionCaso
Descripción:	Obtener un TipoRelacionActaProcesalPersona según una persona
Nombre:	toString
Descripción:	Retorna un string que identifica el acta procesal
Nombre:	equals
Descripción:	Compara el objeto pasado por parámetro y retorna falso o verdadero

Nombre: Denuncia	
Tipo de clase: Bean Persistente que guarda los datos de una Denuncia	
Atributo	Tipo
testimonio	String
resenna	String
delito	Delito
tipo	boolean
funcionario	Funcionario
orgDenunciante	EnteExterno
denunciasPersonas	Set<DenunciasPersonas>
denunciasElementos	Set<DenunciaElemento>
articulos	Set<Articulo>
denunciaSigepol	EPDenunciaSigepol
Para cada responsabilidad:	
Nombre:	toString
Descripción:	Retorna un string que identifica el acta procesal
Nombre:	equals
Descripción:	Compara el objeto pasado por parámetro y retorna falso o verdadero

Nombre: Delito	
Tipo de clase: Bean Persistente que guarda los datos de un delito	
Atributo	Tipo
modusOperandi	String
horaAproximada	HoraAproximada
fecha	Date
hora	Date
sitioSuceso	SitioSuceso
naturalezaDelito	NaturalezaDelito
gradoDelito	GradoDelito
nivelImportancia	NivelImportancia
tipoDelito	TipoDelito
Para cada responsabilidad:	
Nombre:	toString
Descripción:	Retorna un string que identifica el acta procesal
Nombre:	equals
Descripción:	Compara el objeto pasado por parámetro y retorna falso o verdadero

Nombre: EPDenunciaSigepol	
Tipo de clase: Bean Persistente que guarda los datos de una denuncia SIGEPOL	
Atributo	Tipo
fecha	Date
descripcion	String
delitoFalta	String
fechaDelito	Date
sigepolPersonas	Set<EPDenunciaSigepolPersona>
sigepolVehiculo	Set<EPDenunciaSigepolVehiculo>
sigepolArmas	Set<EPDenunciaSigepolArma>
sigepolObjeto	Set<EPDenunciaSigepolObjeto>

Nombre: AutoInicio	
Tipo de clase: Bean Persistente que guarda los datos de un auto de inicio.	
Atributo	Tipo
fechaEmision	Date
horaEmision	Date
fiscal	Fiscal
Para cada responsabilidad:	
Nombre:	toString
Descripción:	Retorna un string que identifica el acta procesal
Nombre:	equals
Descripción:	Compara el objeto pasado por parámetro y retorna falso o verdadero

2.2.6. Realizaciones de casos de uso.

A continuación se muestra el diagrama de contrato entre paquetes de los casos de usos más relevantes escogidos anteriormente.

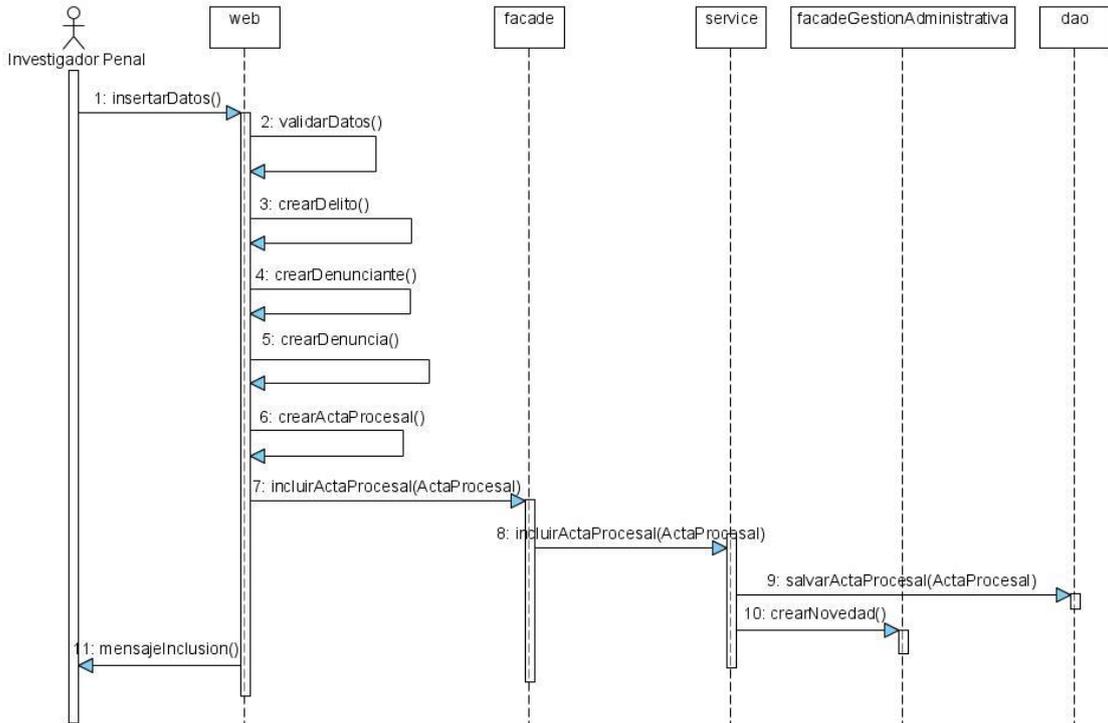


Ilustración 40. Diseño. CU Iniciar Investigación. Diagrama de Contratos entre Paquetes.

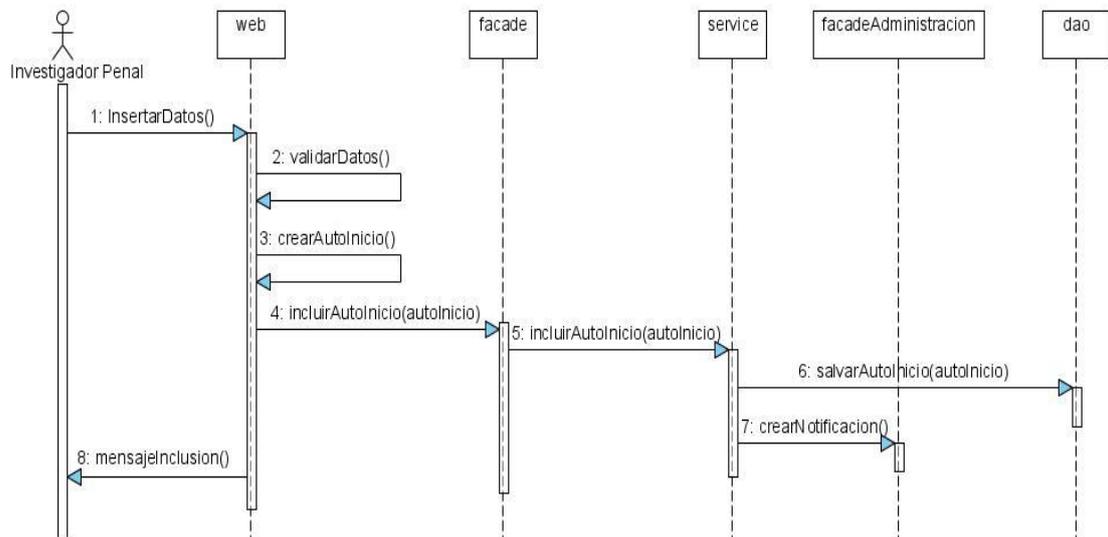


Ilustración 41. Diseño. CU Gestionar Auto de Inicio. Diagrama de Contratos entre Paquetes. Escenario Incluir.

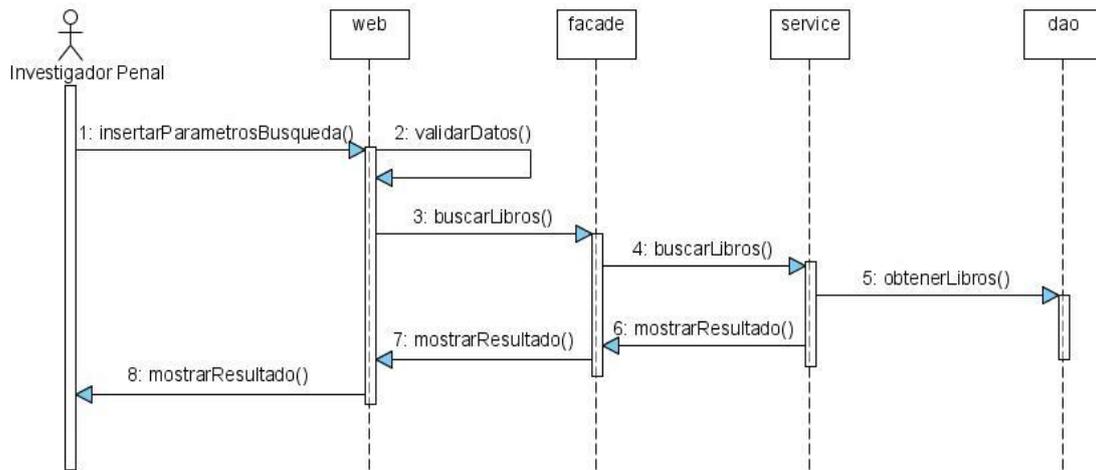


Ilustración 42. Diseño. CU Consultar Libro de Control de Investigación. Diagrama de Contratos entre Paquetes.

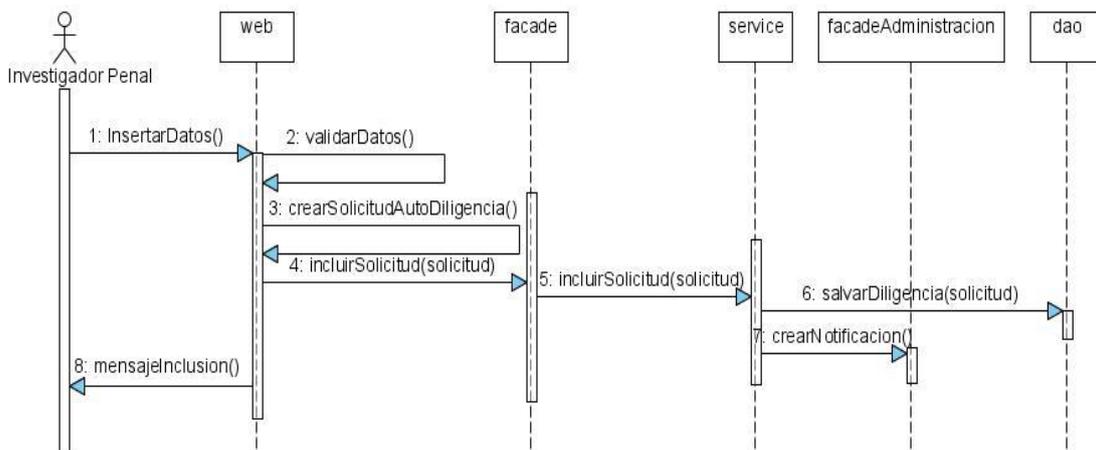


Ilustración 43. Diseño. CU Gestionar Solicitud de Autorización de Diligencia. Diagrama de Contrato entre Paquetes. Escenario Incluir.

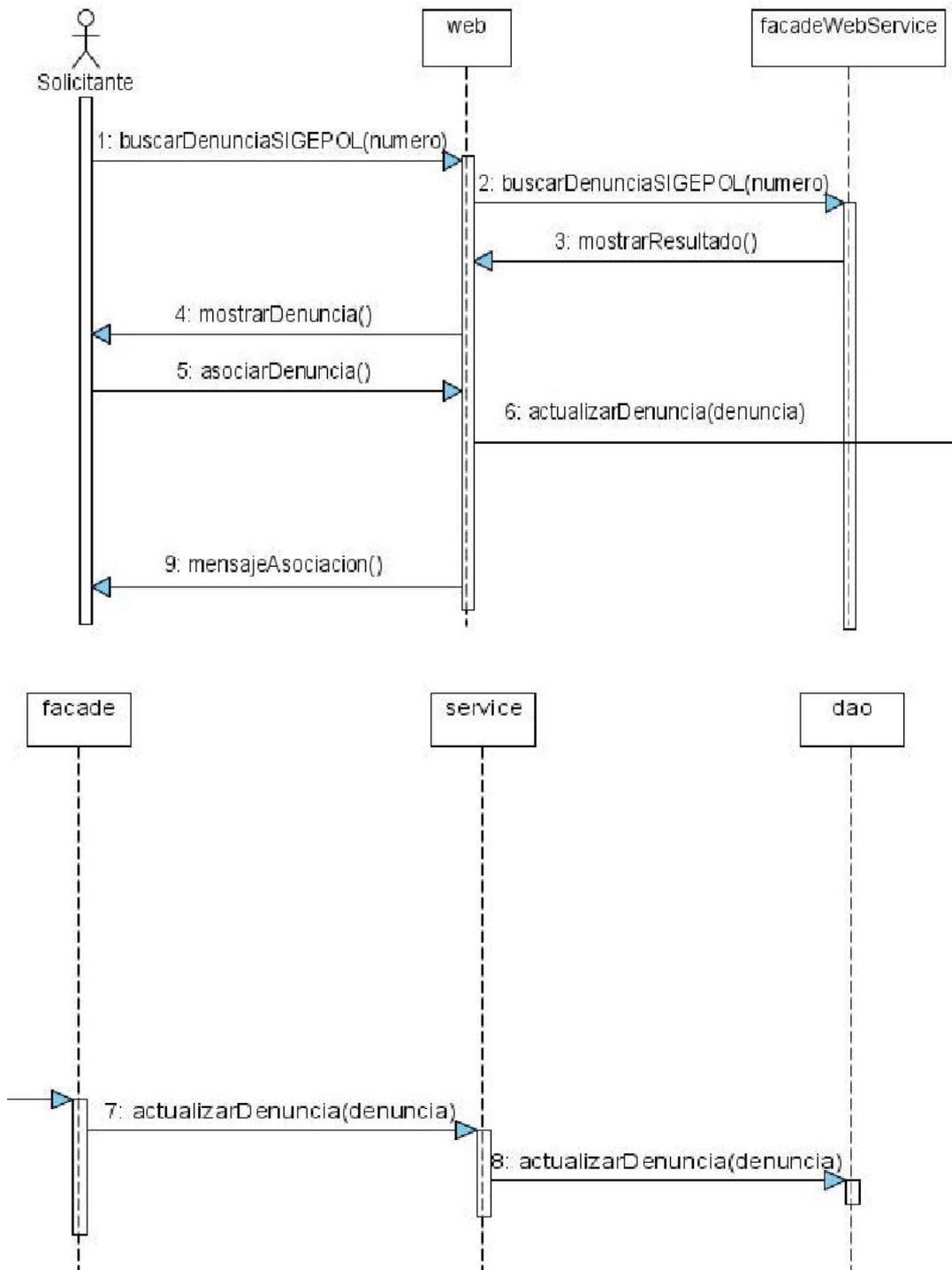


Ilustración 44. Diseño. CU Transferir Copia de Denuncia Policial SIGEPOL. Diagrama de Contratos entre Paquetes.

2.3. Modelo de Datos.

2.3.1. Diagramas de Tablas del Modelo Relacional.

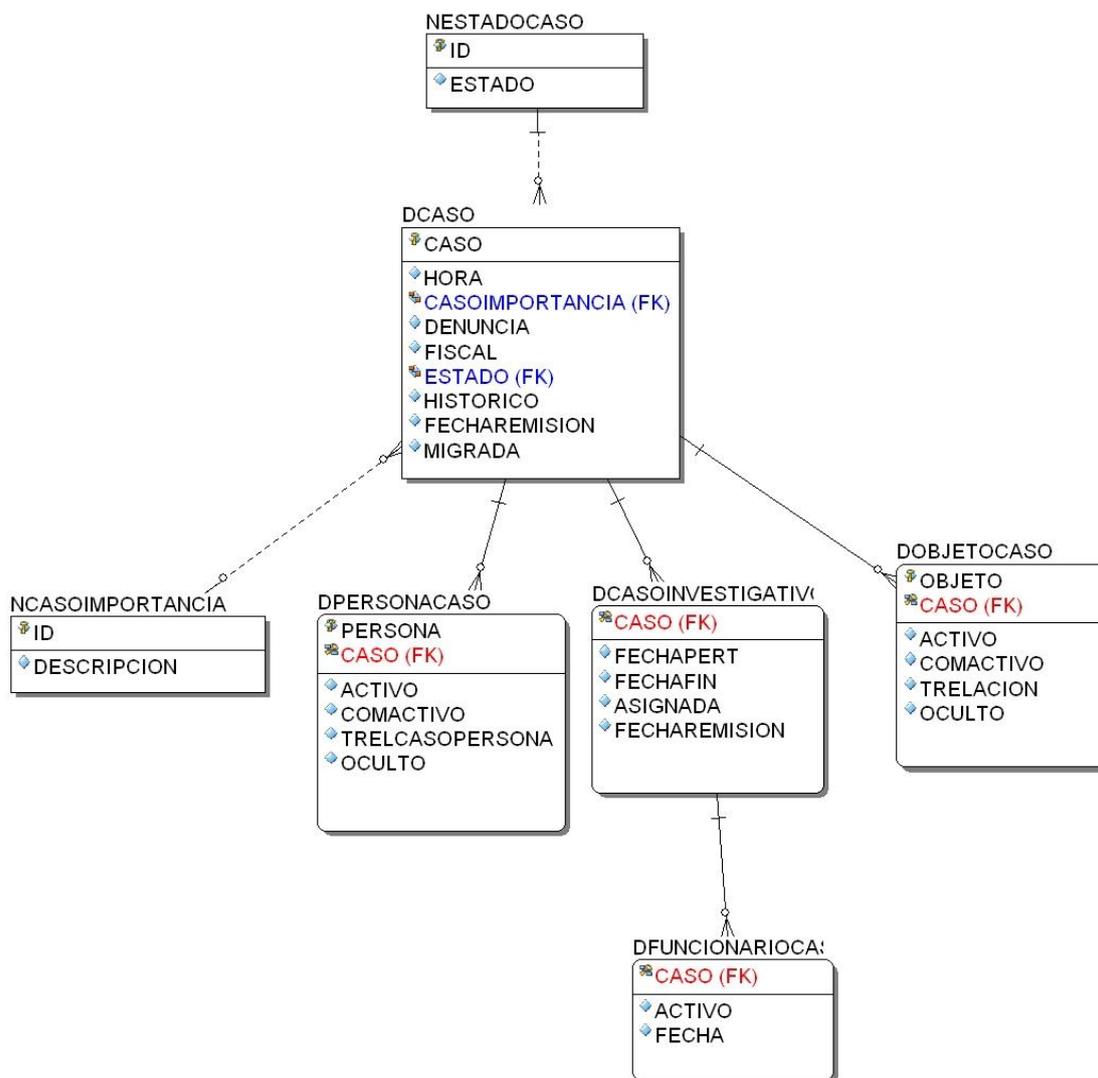


Ilustración 45. Diseño. Acta Procesal. Modelo de Datos.

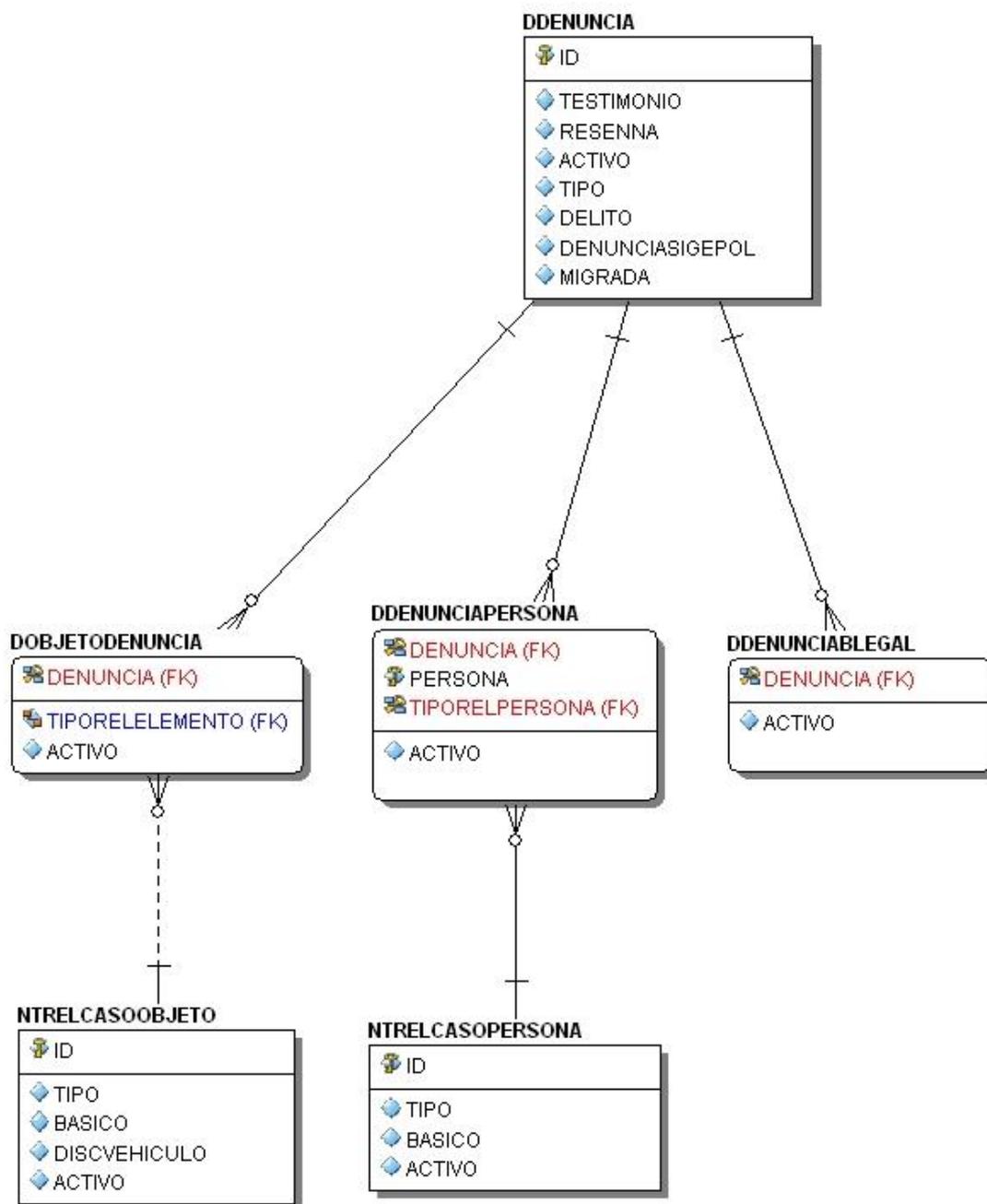


Ilustración 46. Diseño. Denuncia. Modelo de Datos.

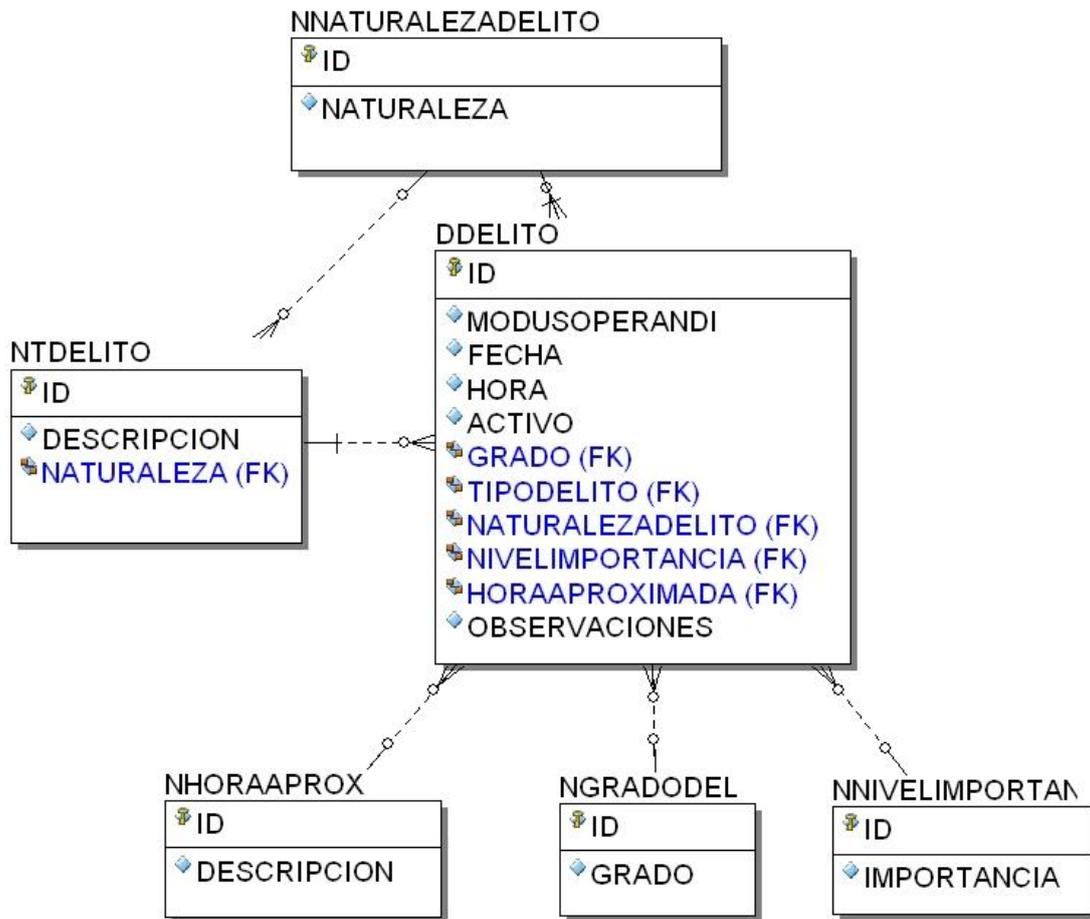


Ilustración 47. Diseño. Delito. Modelo de Datos.

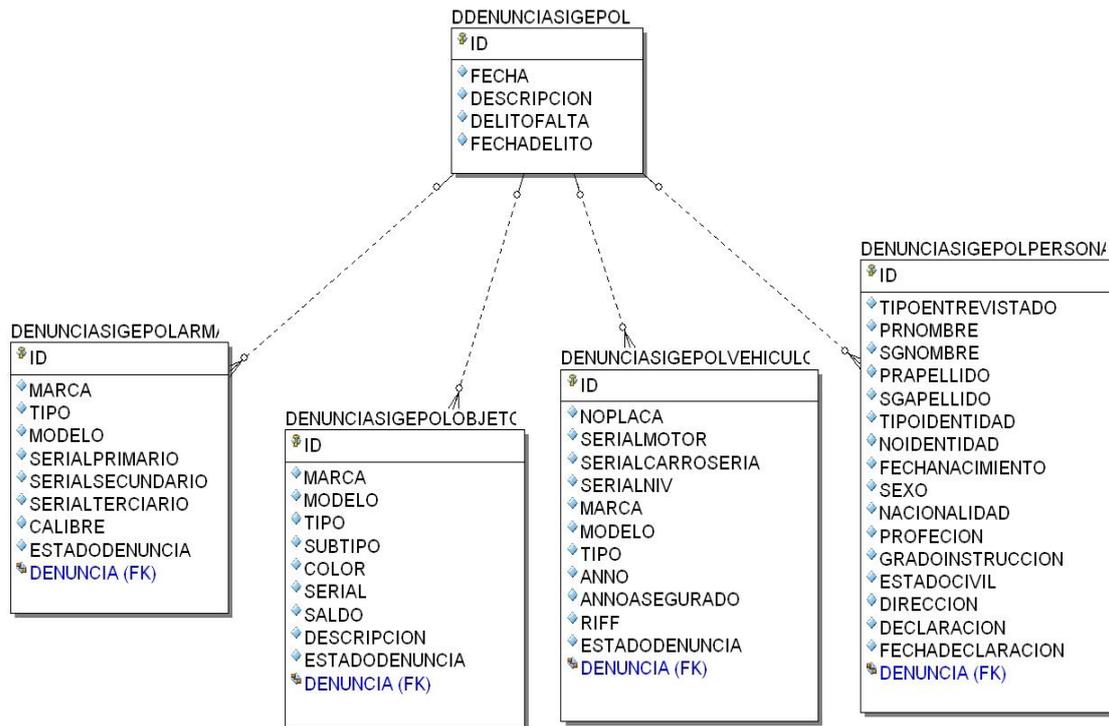


Ilustración 48. Diseño. DenunciaSIGEPOL. Modelo de Datos.

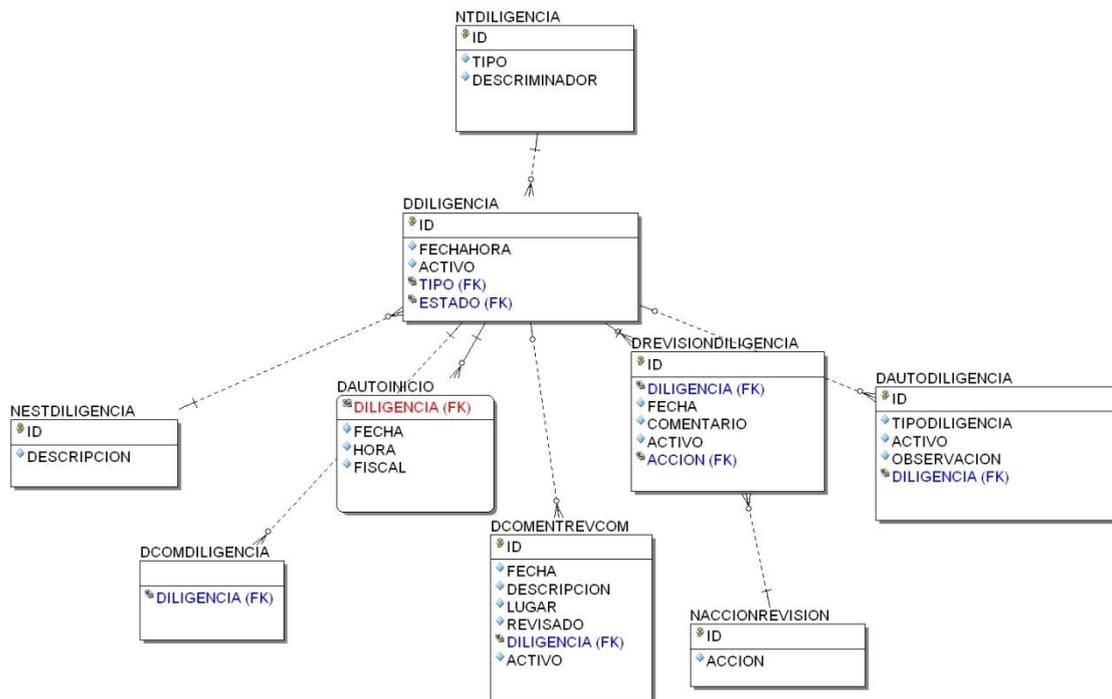


Ilustración 49. Diseño. Diligencia. Modelo de Datos.

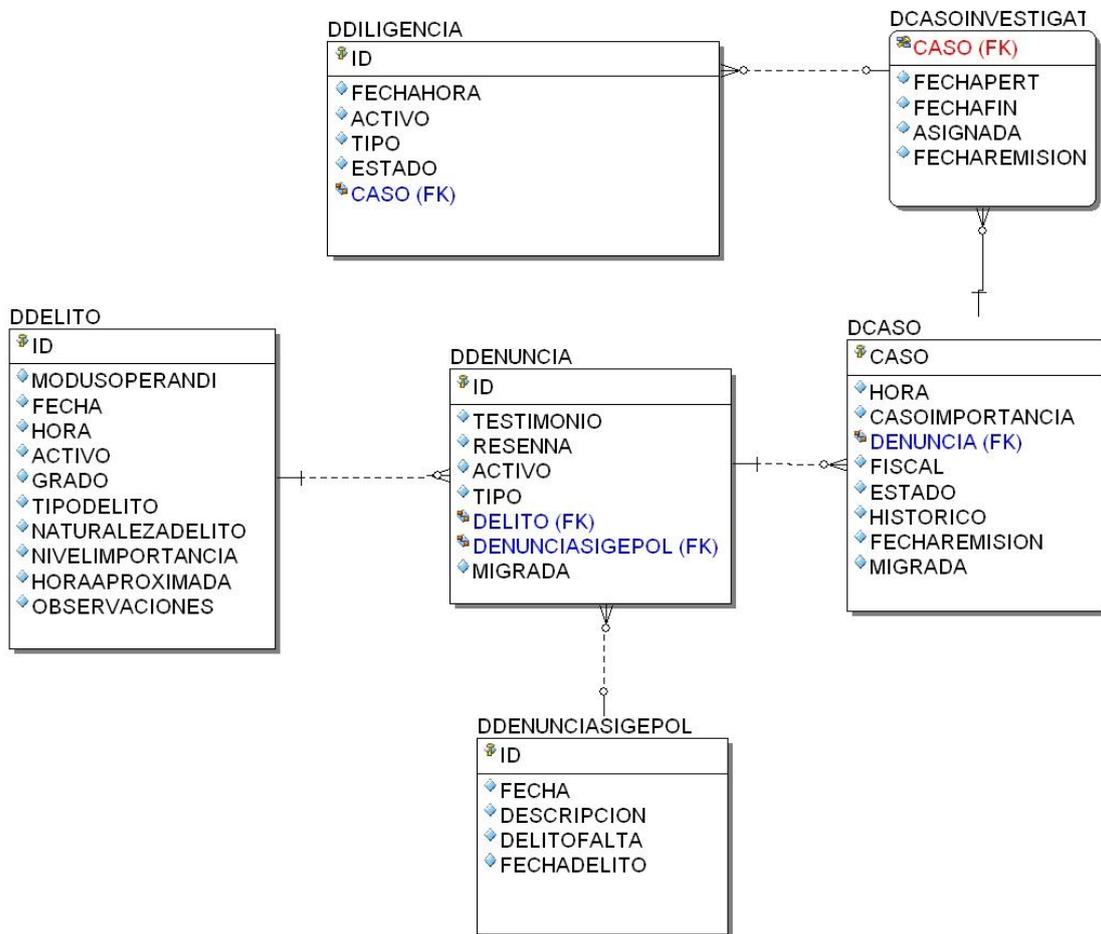


Ilustración 50. Diseño. Conexión entre las Principales Tablas. Modelo de Datos.

2.4. Modelo de Implementación.

2.4.1. Diagramas de subsistemas de implementación.

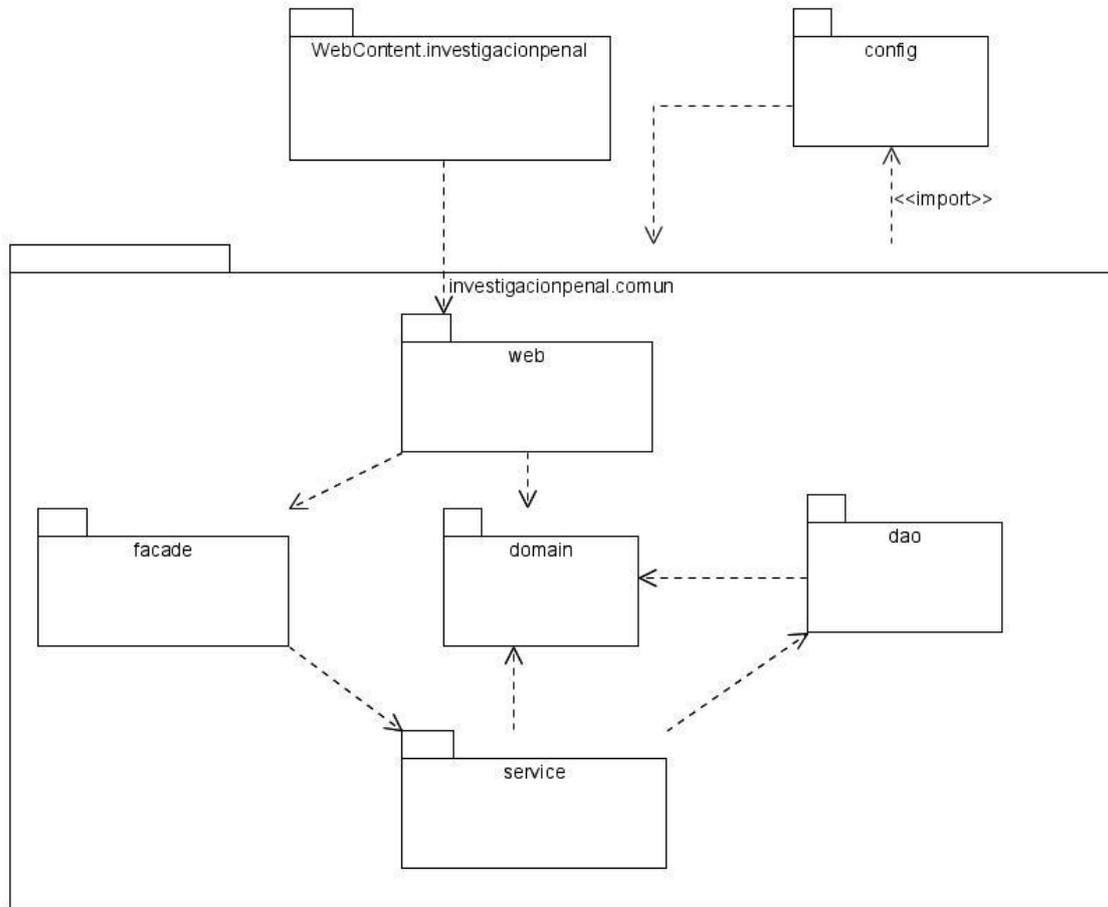


Ilustración 51. Diagramas de Subsistemas de Implementación.

2.4.2. Diagrama de componentes.

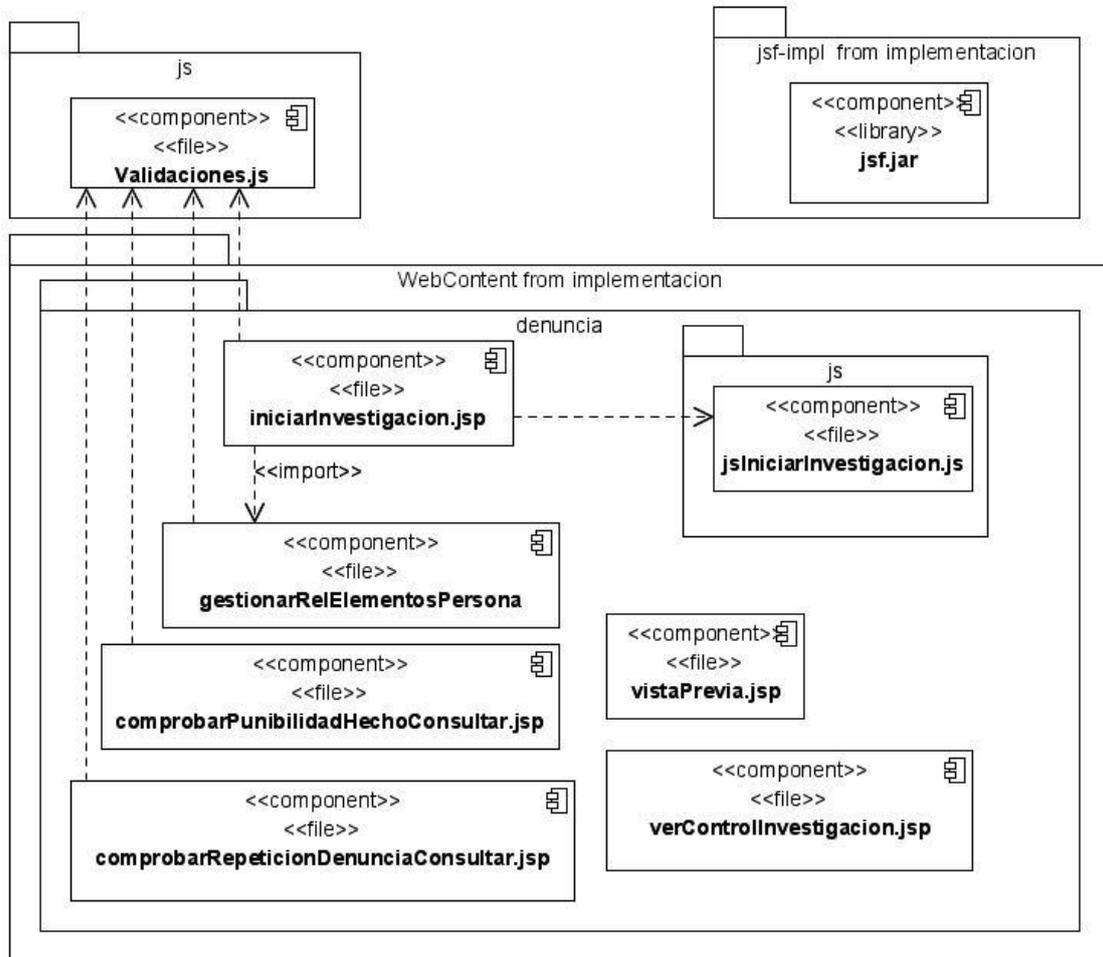


Ilustración 52. Implementación. Páginas. Sub Módulo Denuncia. Diagrama de Componentes.

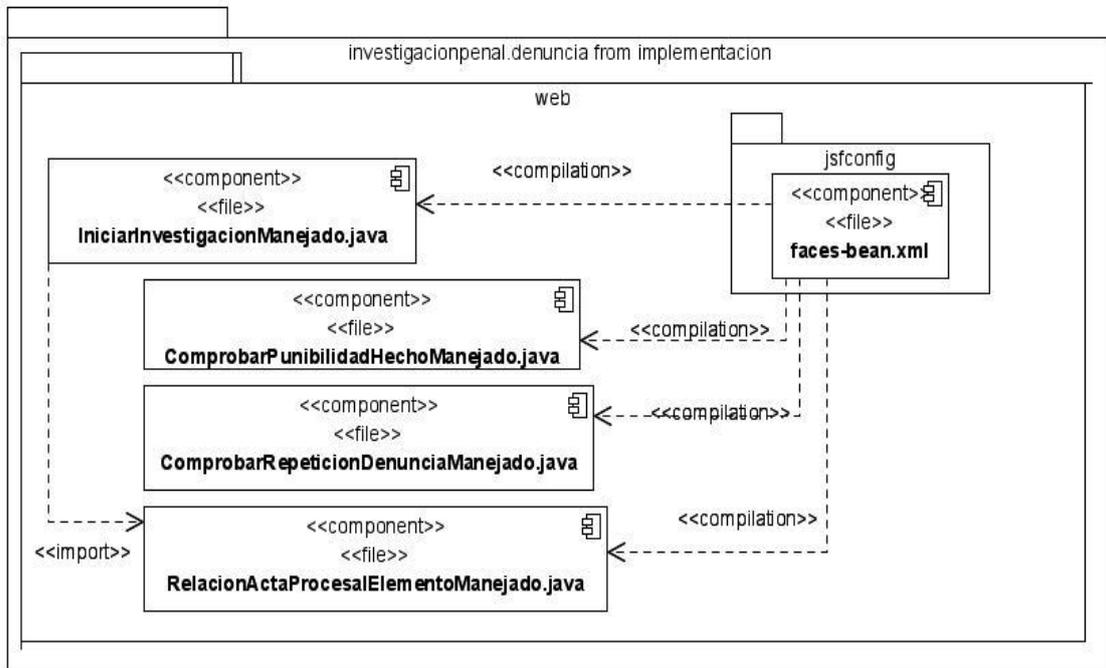


Ilustración 53.Implementación. Web. Sub Módulo Denuncia. Diagrama de Componentes.

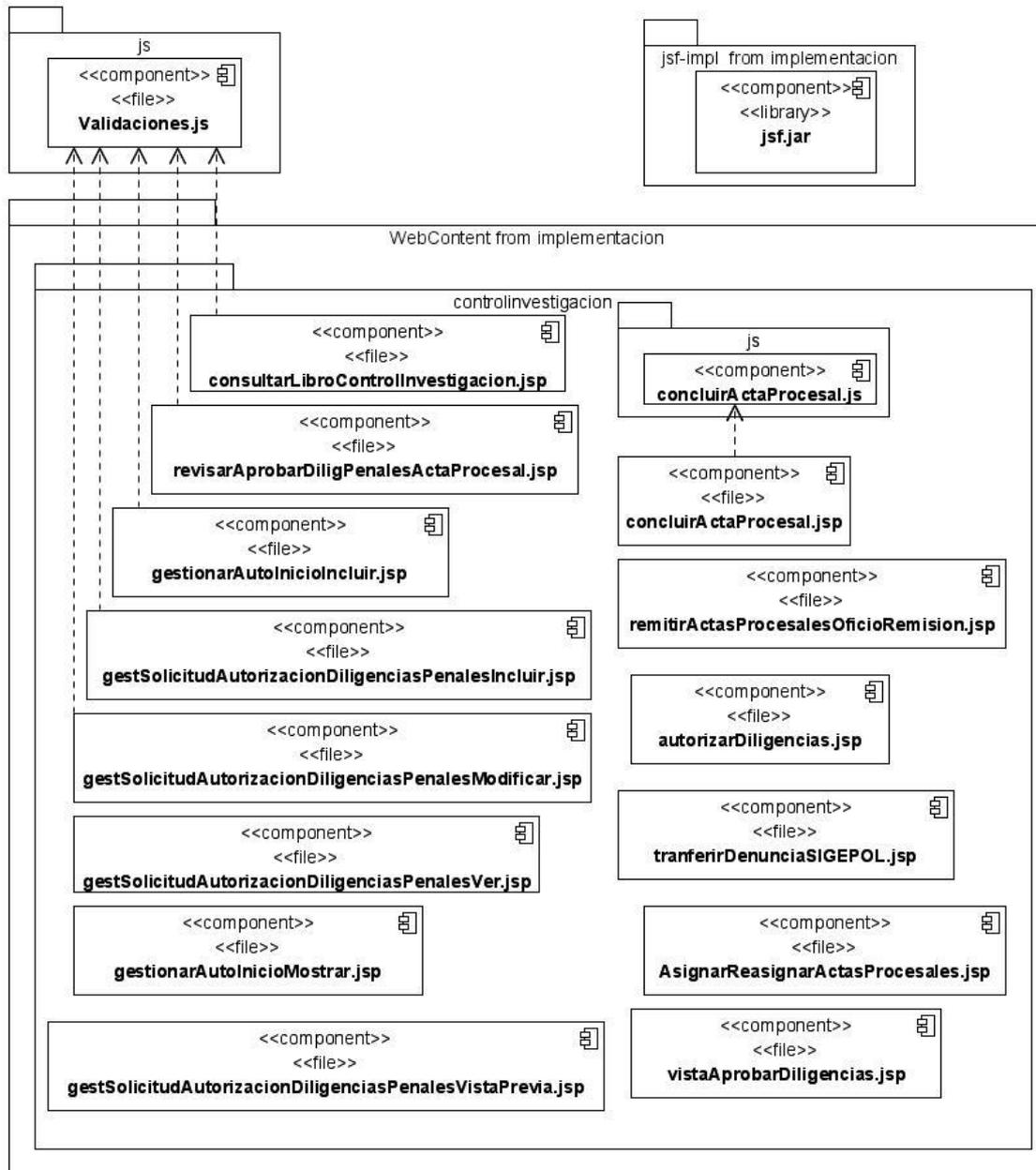


Ilustración 54. Implementación. Páginas. Sub Módulo Control de Investigación. Diagrama de Componentes.

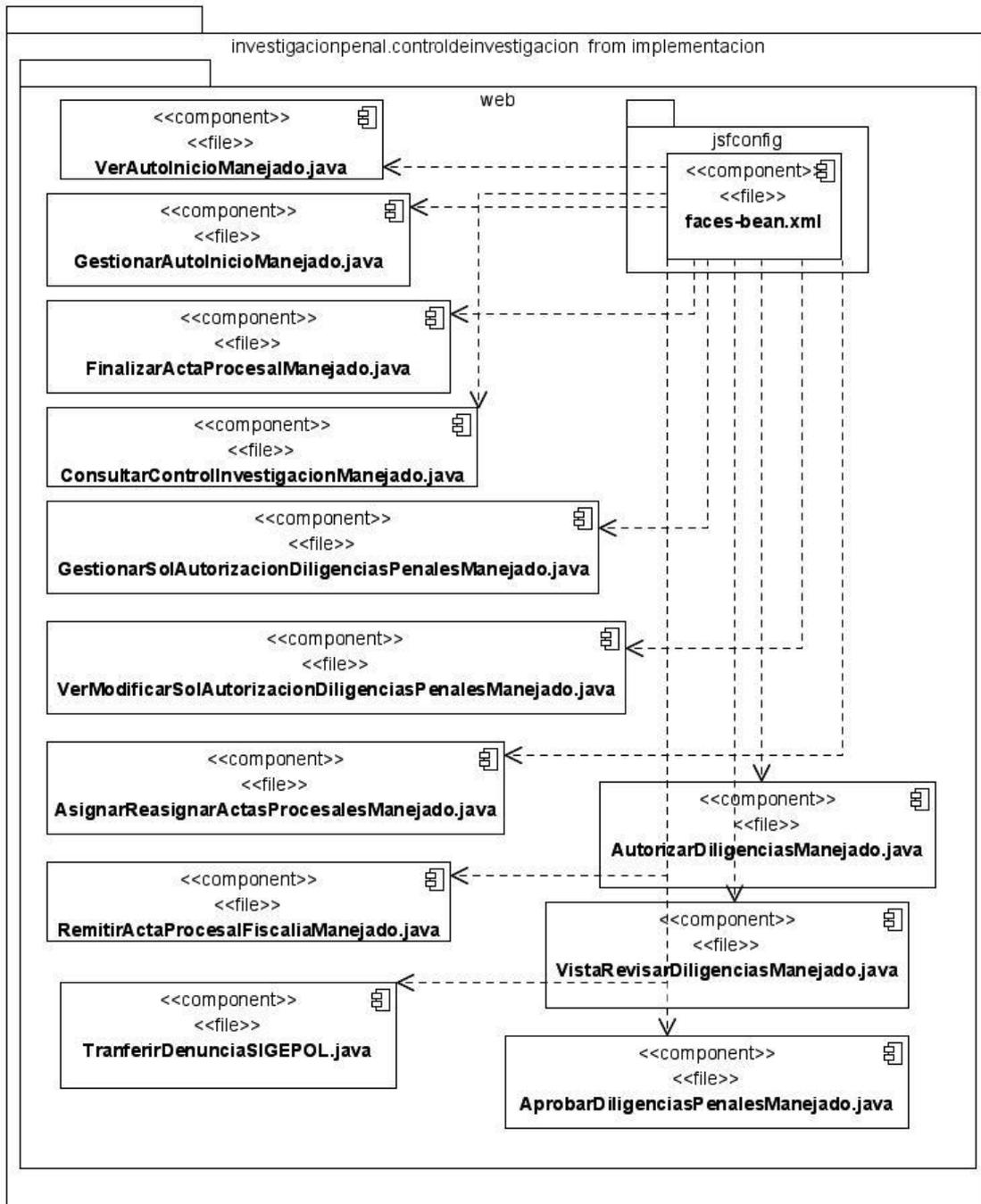


Ilustración 55. Implementación. Web. Sub Módulo Control de Investigación. Diagrama de Componentes.

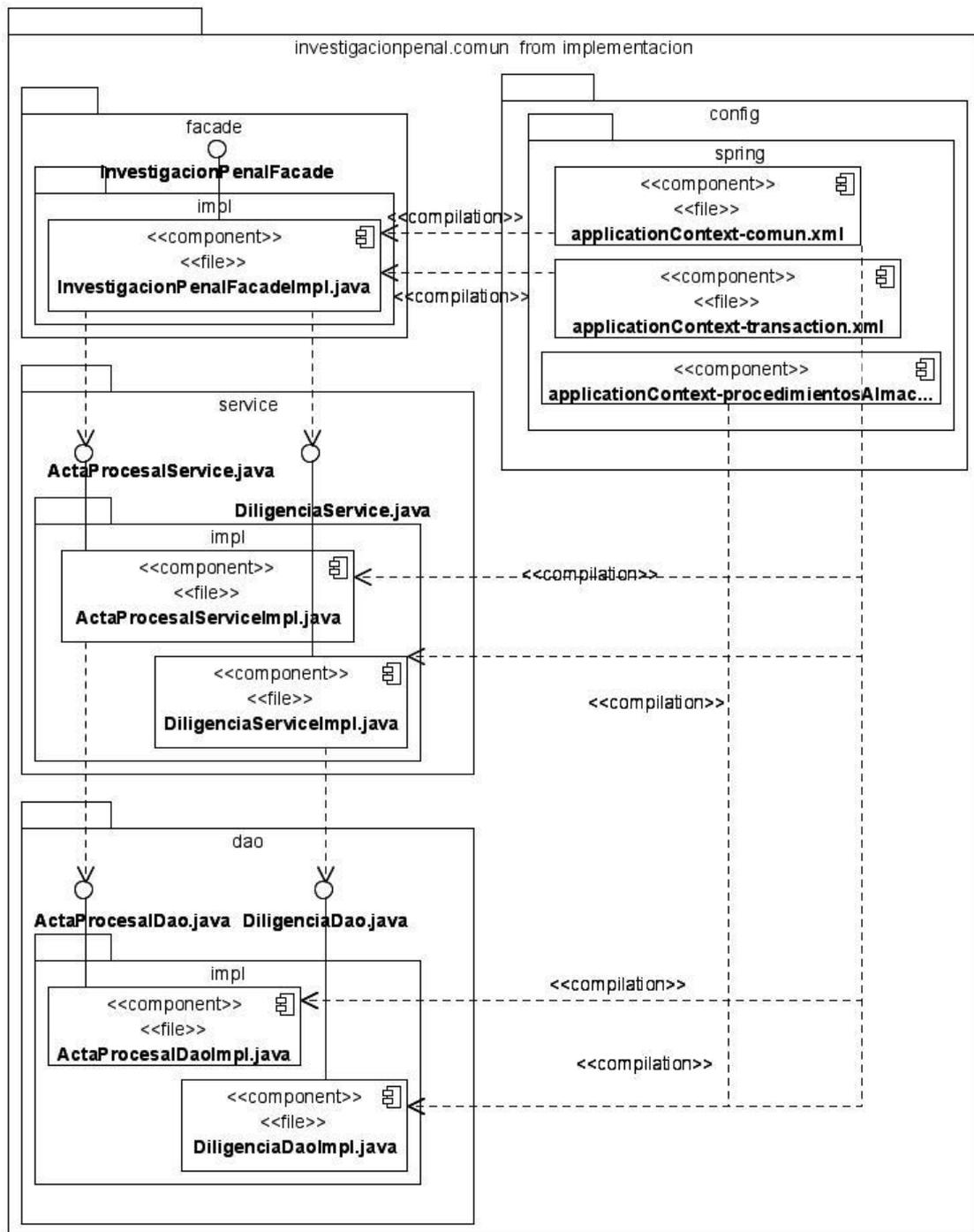


Ilustración 56. Implementación. Negocio y DAO. Diagrama de Componentes.

Conclusiones.

En este capítulo se trataron los temas referentes al análisis, diseño e implementación de los Sub Módulos Denuncia y Control de Investigación partiendo de los artefactos generados en el flujo de levantamiento de requisitos. Como resultado de las tareas se obtuvieron los diagramas de clases de análisis y diseño así como las realizaciones de los casos de usos; estos contribuyeron a la creación de los subsistemas de implementación y los diagramas de implementación.

CAPITULO 3.

VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

Las pruebas del software se integran dentro de las diferentes fases del ciclo de RUP, estas se centran principalmente en la evaluación o la valoración de la calidad del producto. A diferencia de otros flujos como requisitos, análisis, diseño e implementación. Su principal objetivo es exponer los puntos flacos del software (4).

A continuación se tratan temas relacionados a las pruebas y que desbordan en el objetivo del capítulo: Validar la propuesta de Sistema.

Entre los conceptos relacionados a las pruebas se distinguen:

- Métodos de prueba.
- Tipos de Pruebas.
- Niveles de Pruebas.

3.1. Métodos de Pruebas.

Entre los métodos de pruebas utilizados en la validación de la propuesta de solución están:

3.1.1. Prueba de Caja Blanca.

Este método permite examinar la estructura interna del programa, comprobando los caminos lógicos del software y examinando el estado del programa en varios puntos para determinar si el estado real coincide con el esperado.

3.1.2. Prueba de Caja Negra.

Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

3.2. Tipos de Pruebas.

Dentro de los tipos de prueba definidos para la propuesta están las pruebas de funcionalidad, Seguridad, Disponibilidad y Red, Rendimiento, Compatibilidad, Resistencia o Stress y Usabilidad. De estos tipos, solo las pruebas de funcionalidad están vinculadas al rol de programador – diseñador por lo que se explicará a continuación.

3.2.1. Prueba de Funcionalidad.

Estas pruebas tienen como objetivo verificar la funcionalidad del sistema al fijar la tensión en la validación de las funciones, métodos, servicios y casos de usos.

Metas

Validar que la aplicación:

- Cumpla con los requisitos funcionales especificados en el diseño de la solución por medio de casos de usos.

- Cumpla con los requisitos no funcionales especificados en el diseño de la solución.
- Cumpla con las restricciones de entrada y salida de la información especificada en el diccionario de Datos
- Cumpla íntegramente con la estructura referencial especificada en el Mapa de Navegación.

3.3. Niveles de Pruebas.

Los niveles de prueba están relacionados básicamente con la profundidad que cada rol de un proyecto le puede aportar. En el caso del rol de programador – diseñador las pruebas correspondientes son las Pruebas Unitarias.

3.3.1. Pruebas Unitarias.

Normalmente cabe distinguir una fase informal antes de entrar en la fase de pruebas propiamente dicha. La fase informal la lleva a cabo el propio codificador en su puesto de trabajo, y consiste en ir ejecutando el código para convencerse de que "básicamente, funciona". Esta fase suele consistir en pequeños ejemplos que se intentan ejecutar. Si el módulo falla, se suele utilizar un depurador para observar la evolución dinámica del sistema, localizar el fallo, y repararlo; a esto se refiere las pruebas de unidad. La automatización de estas pruebas es fundamental para agilizar el proceso, por tal motivo es utilizado el framework JUnit que se explicó en el primer capítulo de este documento.

Para la aplicación de las pruebas unitarias en el modulo de Investigación penal se construyó la clase *InvestigacionPenalFacadeTest* que probaba los métodos de la fachada del módulo. Estas pruebas fueron dirigidas por los mismos desarrolladores y se usaron en ocasiones para ensayar diferentes funcionalidades cruciales.

3.3.2. Pruebas de Integración.

Las pruebas de integración se llevan a cabo durante la construcción del sistema, involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Las pruebas funcionales de integración son similares a las pruebas de caja negra. Aquí se trata de encontrar fallos en la respuesta de un módulo cuando su operación depende de los servicios prestados por otro(s) módulo(s). Según se van acercando al sistema total, estas pruebas se van basando aún más en la especificación de requisitos del usuario.

Las pruebas de integración correspondientes a la presente solución fueron realizadas por calidad uci. A continuación se muestra los problemas detectados.

#	Descripción	Caso de uso	Clasificación
1	No valida los datos que se insertan (Campo: Hora de Ocurrencia)	Iniciar Investigación	media
2	No muestra el listado de preguntas que debe aparecer según	Iniciar Investigación	media

	el tipo de delito.		
3	No muestra la vista previa de la denuncia	Iniciar Investigación	media
4	Los radios y los checkbox tienen mal el estilo, salen con la cajita alrededor	Iniciar Investigación	media
5	Cuando voy a incluir el denunciante y vira para atrás me pone en la pestaña delito en vez de en la pestaña denuncia.	Iniciar Investigación	baja
6	La pestaña preguntas tiene una tabla roja oscura y vacía	Iniciar Investigación	baja
7	Cuando vas a asociar un objeto la tablita que sale ahí le falta la tilde a la palabra número en las dos últimas líneas.	Iniciar Investigación	baja
8	En el guardar da "lazy initialization" error.	Iniciar Investigación	alta
9	Hay un error en las validaciones de java script. Una vez que muestran un error, se echa a perder el resto del java script de la página.	Iniciar Investigación	
10	No tiene el menú arreglado como se estableció por arquitectura de información.	Iniciar Investigación	baja
11	Después de introducir ciertos datos la aplicación dio un error.	iniciar Investigación	alta
12	Aparece en la IU de Control de Investigación Artículo sin tilde.	Iniciar Investigación	baja
13	En la IU de control de investigación aparecen trocados los datos del Acta Procesal y el denunciante. Debería	Iniciar Investigación	baja

	ser al revés.		
14	Cuando se selecciona la opción de realizar una búsqueda avanzada abajo debería salir el vínculo para volver a la búsqueda estándar y en su lugar dice Búsqueda	Consultar Libro de Control de Investigación	baja
15	El campo Despacho que aparece en la especificación del CU, no está incluido en la interfaz de la búsqueda avanzada.	Consultar Libro de Control de Investigación	baja
16	Al modificar los datos personales de un funcionario y actualizarlos me muestra un mensaje diciendo que ha ocurrido un error interno.	Asignar /Reasignar Actas Procesales	media
17	Al intentar entrar a la opción En Curso, me muestra el mensaje de información "Valor requerido." Y no me permite entrar.	Asignar /Reasignar Actas Procesales	media
18	Al asignar un funcionario a un acta procesal me muestra el mensaje de error: "Valor requerido".	Asignar /Reasignar Actas Procesales	media
19	Al concluir un Acta Procesal me lanza una excepción.	Concluir Acta Procesal	alta
20	En el vínculo de Búsqueda Estándar, la palabra Estándar aparece sin tilde.	Consultar Libro de Control de Investigación	baja
21	Cuando se exporta a pdf los datos del Control de Investigación la última parte de la tabla aparece separada de la parte	Consultar Libro de Control de Investigación	baja

	de arriba. "Ver imagen"		
22	Cuando se selecciona la opción de ver una de las diligencias a aprobar y se quiere imprimir o exportar a pdf, lo que hace es refrescar la página.	Aprobar Diligencias Penales	media
23	En la descripción del caso de uso dado en el flujo alterno rechazar uno o varios elementos del listado de Solicitudes de Autorización de Diligencias Penales, hay que entrar un comentario de porque se rechaza, sin embargo no se entra ese comentario, la descripción no explica que pasa y el sistema si muestra un mensaje de error si este no es entrado.	Autorizar Diligencias	media
24	Cuando se acepta un acta y la misma queda remitida, el sistema no muestra el mensaje: "El estado del elemento ha cambiado".	Remitir Actas Procesales	baja

3.3.3. Pruebas de Aceptación.

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable de cara al cliente; sino una vez pasada todas las pruebas de integración por partes de los desarrolladores.

Posterior a las pruebas realizadas por calidad interna y calidad UCI, el software fue llevado a Venezuela donde se les realizo tres iteraciones correspondientes a las pruebas de aceptación y piloto.

A continuación se muestra los problemas detectados y los pedidos de cambios.

Tabla 1. Problemas detectados en las Pruebas de Aceptación.

#	Descripción	Caso de uso	Clasificación
1	Revisar las naturalezas de los delitos, que sean un poco más descriptivas.	Iniciar investigación	media

Tabla 2. Pedidos de cambios de las Pruebas de Aceptación.

#	Descripción	Caso de uso	Clasificación
1	Debe consultar por naturaleza del delito	Consultar Libro de Investigación.	media
2	Incluir todos los tipos de delitos correspondientes a las naturalezas	Iniciar investigación	media

Tabla 3. Problemas detectados en las Pruebas Piloto.

#	Descripción	Caso de uso	Clasificación
1	En el reporte de la denuncia, eliminar el campo testimonio que aparece debajo y colocarlo como parte del texto de la denuncia. Se entregó un ejemplo.	Iniciar Investigación	media
2	En el Reporte del Control de Investigación no marca la nacionalidad, la hora del delito y el día de la semana no sale correcto.	Iniciar Investigación	baja
3	En el reporte de la denuncia, el formato del nombre del denunciante debe coincidir con el del nombre del funcionario que toma la denuncia, falta el cintillo y el reporte no coincide con el formato definido.	Iniciar Investigación	baja

Tabla 4. Pedidos de cambios de las Pruebas Piloto.

#	Descripción	Caso de uso	Clasificación
---	-------------	-------------	---------------

1	Colocar una nueva naturaleza delito: <i>contra el robo y hurto de vehículo</i> , que es una nueva ley que salió.	Iniciar Investigación	baja
---	--	-----------------------	------

Tomando en cuenta los datos anteriores, se muestra la siguiente gráfica donde se representa la relación de las no conformidades en relación a los casos de usos.

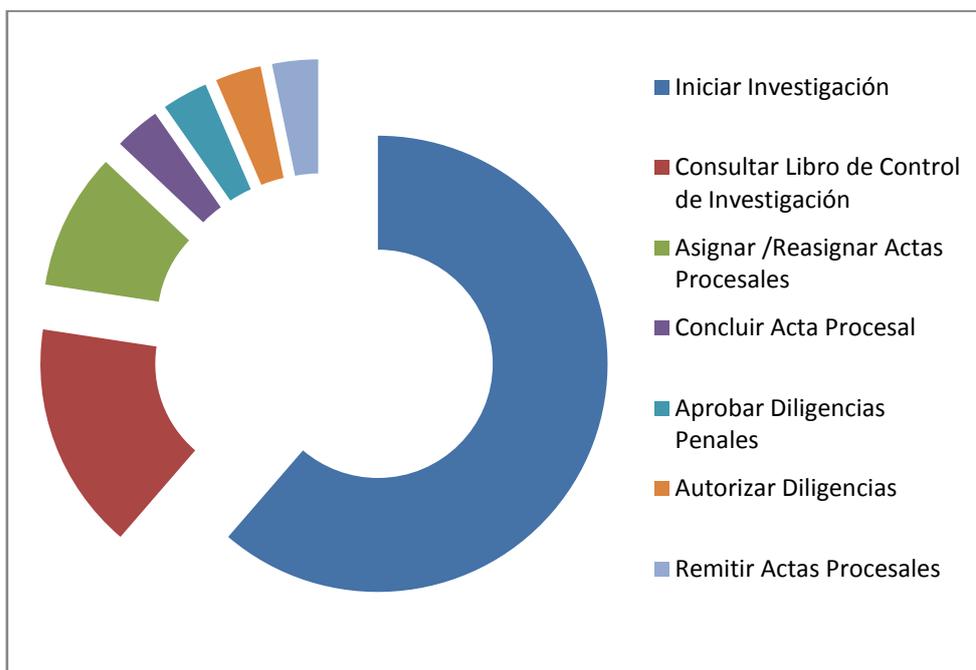


Ilustración 57. Relación de las no conformidades en relación a los casos de usos.

Se puede apreciar que en el Caso de Uso Iniciar Investigación están más del 50 % de los problemas encontrados. Este resultado está aparejado a la complejidad que presenta esta funcionalidad, relacionada principalmente con el enorme flujo de datos que maneja.

En la tabla siguiente se hace referencia a dos etapas de las pruebas, Integración (Calidad UCI) y Aceptación (Pruebas realizadas en Venezuela), se puede apreciar como las dificultades descubiertas disminuyeron drásticamente desde una etapa a otra.

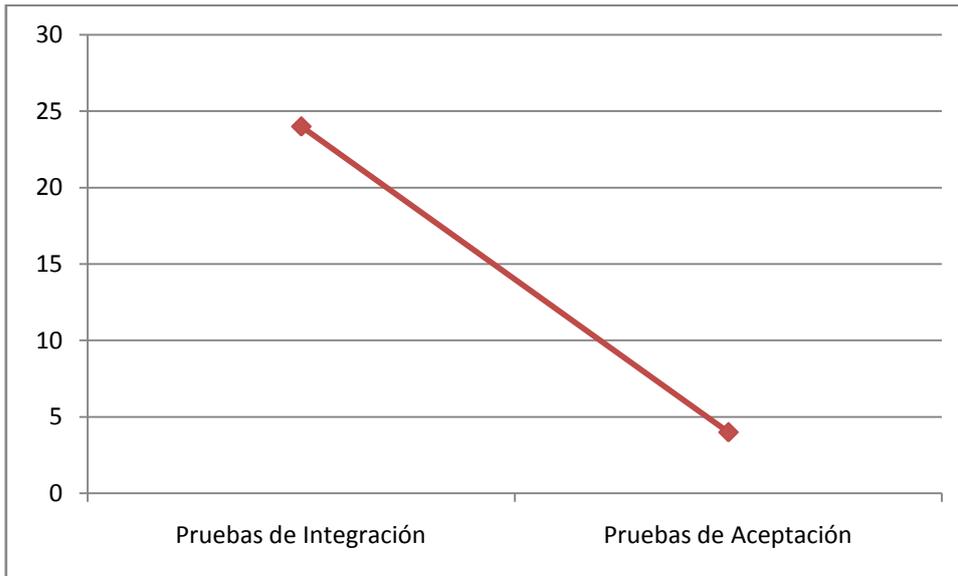


Ilustración 58. Comportamiento de las no conformidades.

En el presente, todas las no conformidades y los pedidos de cambios pactados con el cliente han sido resueltos satisfactoriamente.

Conclusiones.

En el actual capítulo se abordaron los temas referentes a las pruebas que se le efectuaron a la propuesta de solución con el fin de validar que cumpliera con los requisitos funcionales y no funcionales. Finalmente se realizó un estudio teniendo en cuenta los problemas detectados y como estos se distribuían por los diferentes casos de usos.

CONCLUSIONES.

- ✓ Se han logrado obtener dos sub módulos de software funcionales e integrados al SIIPOL como principal resultado de este trabajo, además de todo un conjunto de documentación que ha sido fruto de el análisis, diseño e implementación de dichos sub módulos, verificándose a su vez el cumplimiento de los objetivos en su totalidad.
- ✓ Se le fue dando cumplimiento a cada uno de los objetivos en el tiempo dedicado al desarrollo, confiriéndole gran importancia a la tarea de confeccionar un diseño teórico de la investigación por constituir la base a partir de la cual se obtienen con gran claridad los puntos iniciales de la investigación.
- ✓ Una vez sentadas las bases de análisis se diseñó e implementó el resultado de la ingeniería de requerimientos. Posteriormente a su implementación, los sub módulos han logrado pasar diferentes iteraciones del flujo de trabajo de Pruebas de RUP corrigiéndose una serie de errores existentes, aprobándose el sistema por parte del equipo de trabajo de control de la calidad, permitiendo que la primera versión estuviera lista para pasar las pruebas de aceptación; logrando a su vez que una representación del cliente acepte al sistema propuesto.
- ✓ Dicho modelo de sistema, ha sido estudiado detalladamente, por cada caso de uso y de manera iterativa como se propone en RUP, el resultado permitió que se corrigiera todo detalle pasado por alto en el trabajo que se antecede al anteriormente descrito en este documento. Lográndose en su totalidad un gran conocimiento de los procesos que se automatizaron así como un conjunto de todos los conceptos que se manejaron por parte de los desarrolladores dígase tanto diseñadores como programadores. A partir de este momento con todo el conocimiento adquirido, fue una tarea de dedicación y esmero diseñar e implementar todo un conjunto de clases dando cumplimiento a los requisitos funcionales y no funcionales asociados a los casos de usos pertenecientes a los Sub Módulos Denuncia y Control de Investigación del Módulo Investigación Penal.
- ✓ Otra tarea que no se debe pasar por alto ha sido la integración al sistema, que se logró llevando a cabo las buenas prácticas de diseño e implementación, como la utilización de patrones, convenciones de código y documentación, llevando consigo a los objetivos propuestos como prueba de cumplimiento de las actividades planificadas. Por todos los resultados obtenidos, parte que se expone en este trabajo y el resto puede ser consultado en la documentación del proyecto, se ha concluido que los objetivos del trabajo han sido cumplidos satisfactoriamente.

RECOMENDACIONES.

Realizar una refactorización de la implementación de los casos de uso del sistema para optimizar el rendimiento de la aplicación de cara al cliente.

Consultar a los diseñadores y programadores a la hora de la creación de medios visuales, para la documentación de apoyo al usuario, debido al alto nivel de familiarización con el sistema que obtuvieron en el proceso de desarrollo.

Capacitar nuevos desarrolladores para las próximas etapas del sistema, apoyándose en la experiencia de los actuales.

Tener en cuenta el resultado de la implementación de módulo y del sistema en general a la hora de realizar proyectos de iguales características, siempre verificando que se cumpla con las reglas de confidencialidad con la información tratada.

Sugerir un proyecto similar para la aplicación de las Ciencias Policiales en Cuba.

GLOSARIO.

AJAX: Java Script asíncrono y XML es la unión de varias tecnologías que juntas pueden lograr cosas realmente impresionantes como GoogleMaps, Gmail, el Outlook Web Access o algunas otras aplicaciones muy conocidas: AJAX, en resumen, es el acrónimo para *Asynchronous JavaScript + XML* y el concepto es: Cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma.

Ajax4JSF: Ajax4jsf es una librería *open source* que se integra totalmente en la arquitectura de JSF y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax.

AOP: *Aspect-Oriented Programming*, traducido al español: Programación Orientada a Aspectos, paradigma de programación que separa la lógica del negocio de aspectos de servicios, tales como la seguridad, las transacciones, entre otros.

Bean: En el lenguaje Java es un componente software reusable que evita programar los distintos componentes uno a uno. Existen con la finalidad de ahorrar tiempo al programar.

CICPC: Cuerpo de Investigaciones Científicas, Penales y Criminalísticas, es el organismo venezolano responsable de llevar a cabo las investigaciones asociadas a cada uno de los hechos delictivos.

CVS: *Concurrent Versions System*, es un sistema de control de versiones que permite mantener todo el código y ficheros de un proyecto en un servidor, del cual los desarrolladores pueden acceder y hacer sus modificaciones de acuerdo a los permisos pertinentes.

DAO: *Data Access Object*, traducido al español: Objeto de Acceso a Datos, es un patrón de diseño de clases en ingeniería de software que permite a quien lo aplique suministrar una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos.

Framework: Los frameworks orientados al objeto (llámense simplemente frameworks) son la piedra angular de la moderna ingeniería del software. El desarrollo del framework está ganando rápidamente la aceptación debido a su capacidad para promover la reutilización del código del diseño y el código fuente (*source code*). Los frameworks son los Generadores de Aplicación que se relacionan directamente con un dominio específico, es decir, con una familia de problemas relacionados.

Hibernate: Framework de Mapeo - Objeto – Relacional para la plataforma java, con una versión para .NET que permite mapear los atributos de las clases con las entidades de las base de datos, abstrayendo al programador del trabajo con el paradigma relacional.

HQL: Hibernate Query Language, lenguaje de consultas de Hibernate, similar al SQL pero trabaja a nivel de objetos no de tablas.

JNDI: La Interfaz de Nombrado y Directorio Java es una Interfaz de Programación de Aplicaciones para servicios de directorio. Esto permite a los clientes descubrir y buscar objetos y nombres a través de un nombre.

JUnit: Framework de java para efectuar pruebas unitarias.

J2EE: Java 2 Enterprise Edition, define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems. Simplifica las aplicaciones empresariales basándolas en componentes modulares y estandarizados (5).

IoC: *Inversion of Control*, traducido al español Inversión de Control, técnica usada por Spring donde en lugar de los objetos buscar sus dependencias, el propio contenedor se responsabiliza por buscarlas.

Natural-Adabas: Es una base de datos jerárquica de alto rendimiento creada por la empresa alemana Software AG, en el año 1969.

OASIS: *Organization for the Advancement of Structured Information Standards*, es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web.

ORM: Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos.

Plugin: Complemento, aplicación informática que agrega o reemplaza funciones que no existen en el programa original.

POJO: *Plain Old Java Object*, utilizada por programadores java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

RUP: Metodología de desarrollo de software. Es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo.

SIIPOL: Sistema informático que actualmente se encuentra desplegado en algunas de las áreas del CICPC, las siglas responden a Sistema Integrado de Información Policial y con la nueva reestructuración pasa a nombrarse Sistema de Investigación e Información Policial.

SOAP: *Simple Object Access Protocol*, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

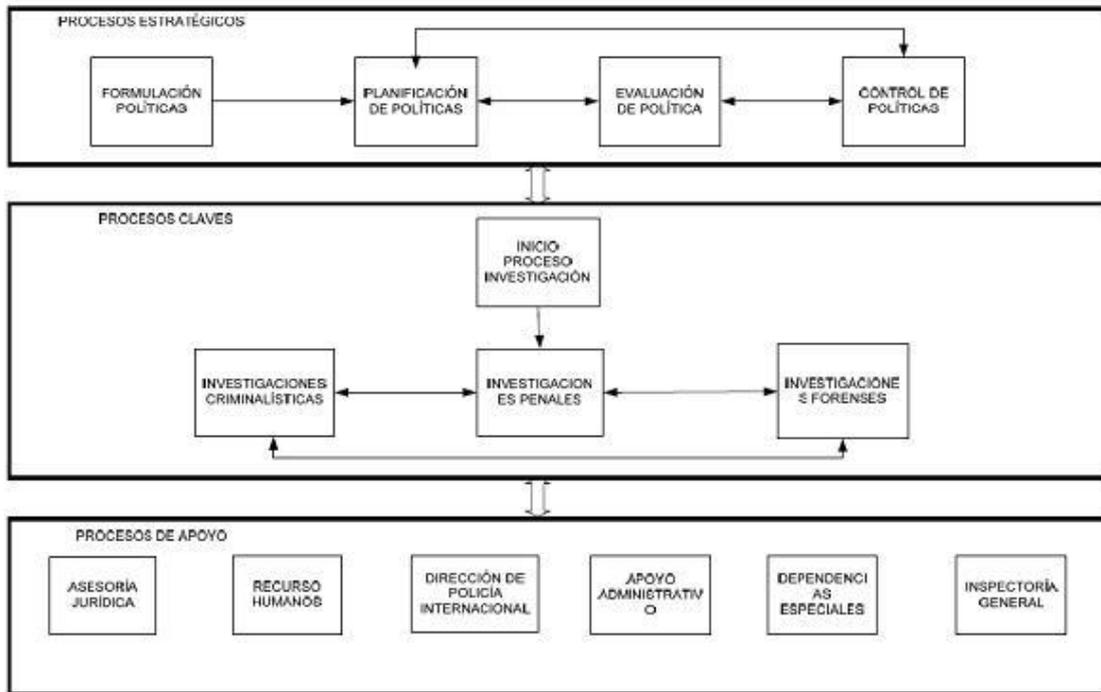
Sun Microsystems: Compañía estadounidense de computadoras creada en 1983, con sede central en Mountain View, California. La compañía ha sido líder en la fabricación de estaciones de trabajo de computadoras. En el campo del software además del lenguaje Java se le atribuye la creación del Sistema operativo Solaris (5).

UML: Lenguaje Unificado de Modelado, es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

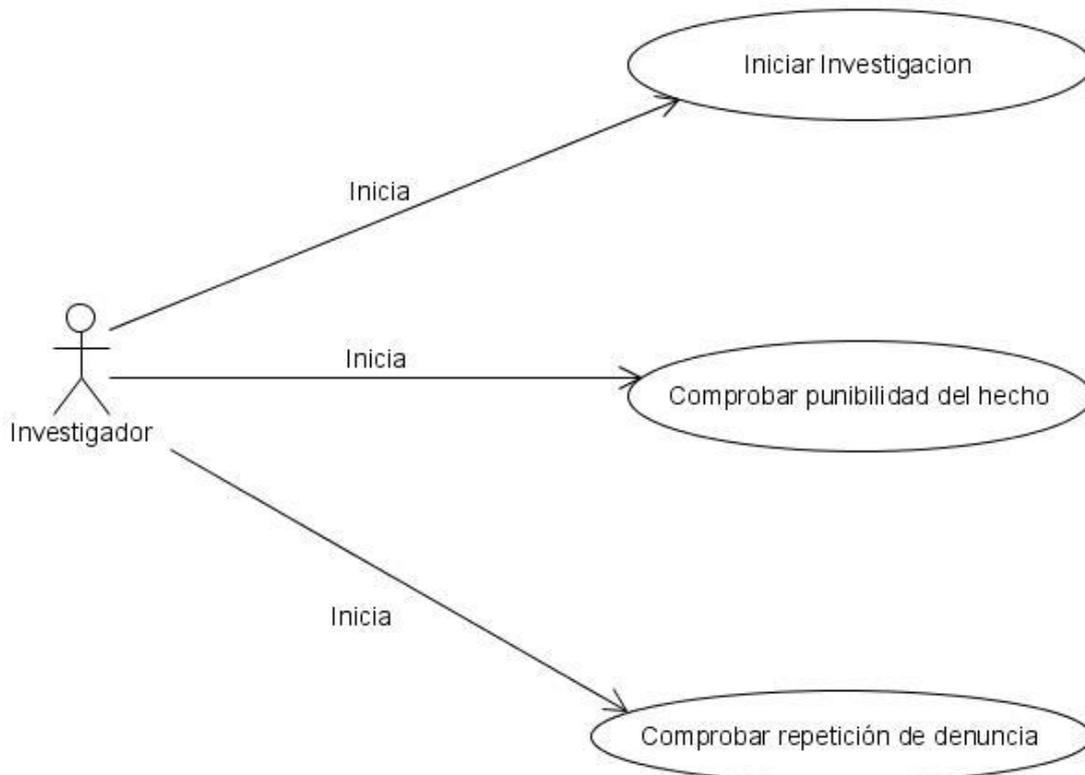
W3C: *World Wide Web Consortium*, es un consorcio internacional que produce estándares para la World Wide Web (6).

ANEXOS.

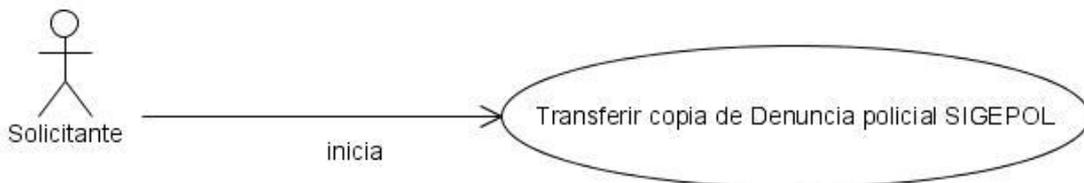
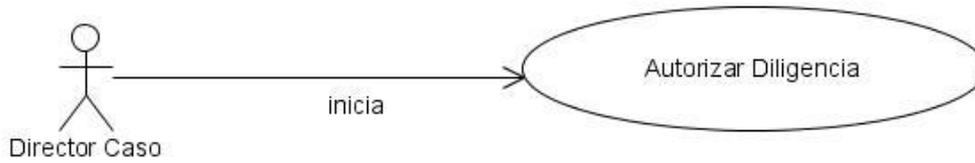
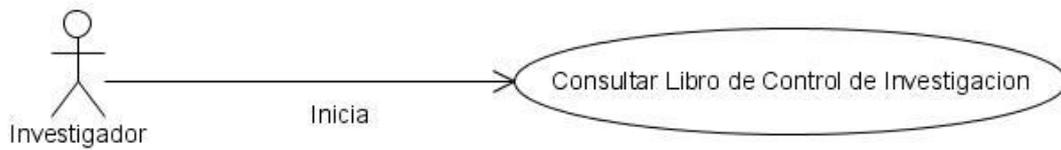
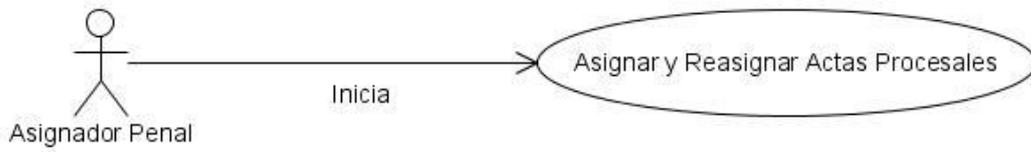
Anexo 1: Diagrama de Procesos del CICPC.



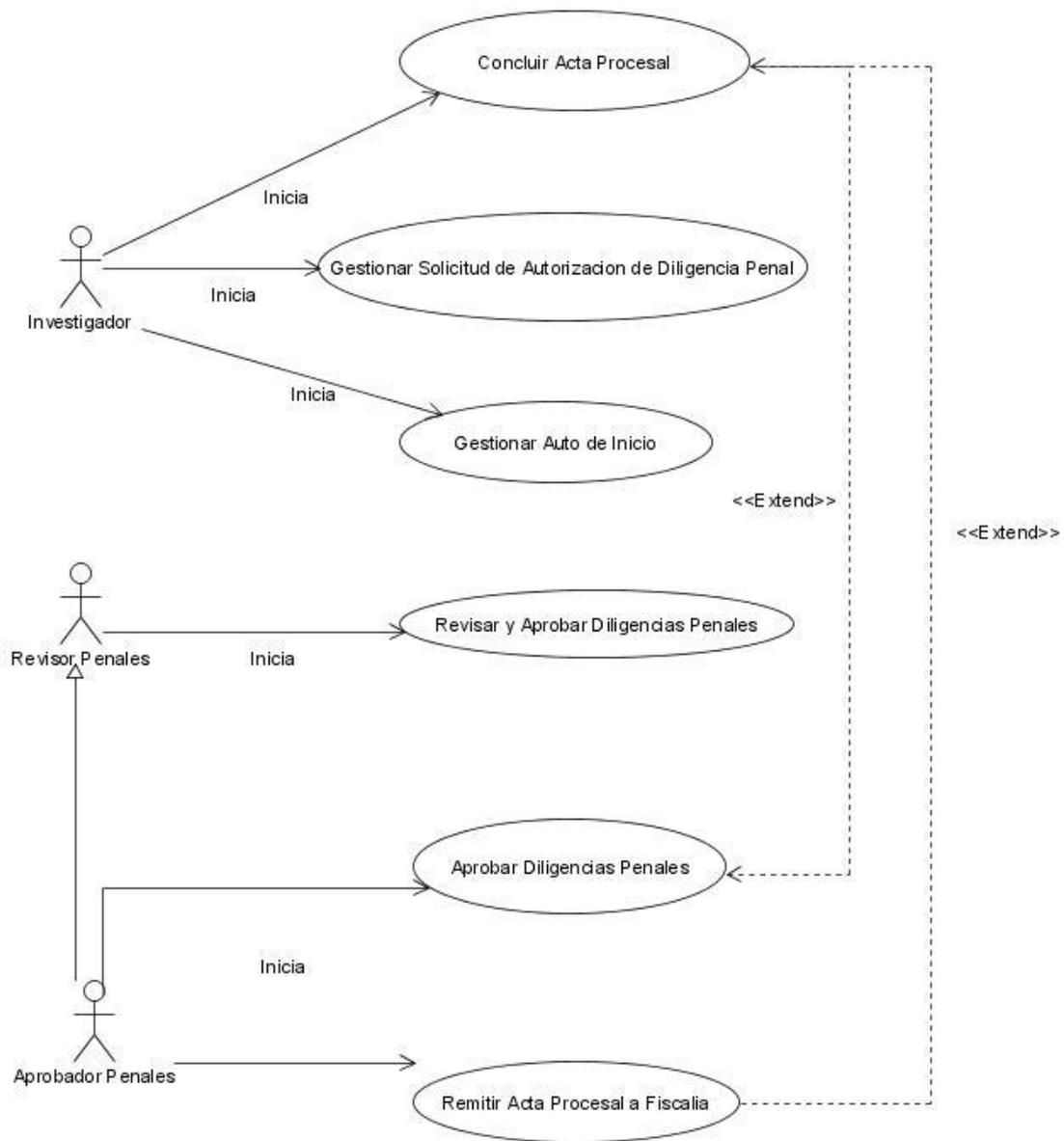
Anexo 2: Diagrama de casos de usos perteneciente al Sub Módulo Denuncia.



Anexo 3: Diagrama de casos de usos perteneciente al Sub Módulo Control de Investigación. Primera parte.



Anexo 4: Diagrama de casos de usos perteneciente al Sub Módulo Control de Investigación. Segunda parte.



Anexo 5: CU Asignar / Reasignar Actas Procesales. Descripción reducida.

<i>Caso de Uso : Asignar / Reasignar Actas Procesales.</i>	
<i>Actores</i>	Asignador.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Asignar o Reasignar el Acta Procesal a un investigador de la división.
<i>Resumen</i>	El caso de uso inicia cuando el actor accede desde la Agenda de Trabajo a la opción de asignar/reasignar un Acta Procesal que pertenece a su Despacho. El sistema muestra un listado de las Actas Procesales que posean el estado: "en Curso" y "Iniciado", y muestra si esta "Asignado" o "No Asignado" .El sistema permite reasignar el Acta Procesal a un funcionario diferente o seleccionar un nuevo funcionario responsable del caso. El caso de uso termina.

Anexo 6: CU Aprobar Diligencias Penales. Descripción reducida.

<i>Caso de Uso : Aprobar Diligencias Penales.</i>	
<i>Actores</i>	Aprobador de Diligencias Penales.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Aprobar una o más Diligencias Penales que ya hayan sido revisadas.
<i>Resumen</i>	El caso de uso inicia cuando el actor accede a la bandeja Aprobaciones desde la Agenda de Trabajo .El sistema lista todas las Diligencias Penales que han sido revisadas y en espera de ser aprobadas. Una vez que el Aprobador de Diligencias Penales accede a la Diligencia Penal que desea aprobar, el sistema permite seleccionar si la anula, aprueba o aprueba y envía, enviando una Notificación informando a los interesados de la acción realizada.

Anexo 7: CU Autorizar Diligencias. Descripción reducida.

<i>Caso de Uso : Autorizar diligencias.</i>	
<i>Actores</i>	Director del Caso.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Incluir una autorización de una serie de diligencias en un Acta Procesal.
<i>Resumen</i>	El caso de uso se inicia cuando el actor selecciona la opción de autorizar diligencias penales de un Acta Procesal seleccionada. El sistema muestra un listado de Solicitudes de Autorización de Diligencias Penales que se realizaron al Acta Procesal seleccionada. El sistema brinda la posibilidad de seleccionar una o varias para ser aprobadas o rechazadas, en este último caso brinda la posibilidad de justificar el rechazo. El caso de uso se termina.

Anexo 8: CU Comprobar Punibilidad del Hecho. Descripción reducida.

<i>Caso de Uso : Comprobar punibilidad del hecho.</i>	
<i>Actores</i>	Investigador.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Consultar el Código Orgánico Procesal Penal.
<i>Resumen</i>	El caso de uso se inicia cuando el actor accede a la opción de comprobar punibilidad del hecho. El sistema brinda la posibilidad de buscar un artículo relacionado con un delito, permitiendo incluir un criterio de búsqueda específico, o por palabras calientes. El actor introduce el criterio de búsqueda que desea y se le muestra un listado con los Artículos que respondan al criterio de búsqueda a manera de resumen con la opción de ver sus detalles. Si el actor acceda a la opción de ver los detalles de un Artículo, se muestra su contenido completo dando fin al caso de uso.

Anexo 9: CU Comprobar Repetición de Denuncia. Descripción reducida.

<i>Caso de Uso : Comprobar repetición de denuncia.</i>	
<i>Actores</i>	Investigador.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Consultar denuncias.
<i>Resumen</i>	<p>El caso de uso se inicia cuando el actor accede a la opción de consultar otras denuncias registradas en el sistema.</p> <p>El sistema brinda la posibilidad de introducir el número de cédula del denunciante.</p> <p>El sistema busca alguna coincidencia y le muestra al investigador un resumen con el listado de todas las Denuncias realizadas por la Persona, brindando la posibilidad de ver los detalles de una de ellas.</p> <p>En caso de no encontrar ninguna, el sistema muestra un mensaje indicando que esa persona no tiene denuncias realizadas.</p> <p>El actor accede a la opción de ver los detalles de una de las denuncias encontradas.</p> <p>El sistema muestra el estado de la denuncia, nombre del investigador que la tomó, nombre del investigador que está llevando el caso o que lo llevó y el despacho en el que se tomó la denuncia ordenadas por fecha de toma de Denuncia.</p>

Anexo 10: CU Concluir Acta Procesal. Descripción reducida.

<i>Caso de Uso : Concluir Acta Procesal</i>	
<i>Actores</i>	Investigador.
<i>Nivel</i>	Sub función.
<i>Objetivo</i>	Concluir una investigación penal.
<i>Resumen</i>	<p>El caso de uso inicia cuando el actor accede a la opción de concluir un Acta Procesal. El sistema registra los datos y genera una Notificación Interna para el Aprobador del despacho, indicando que el expediente está pendiente de ser revisado, terminando así el caso de uso.</p>

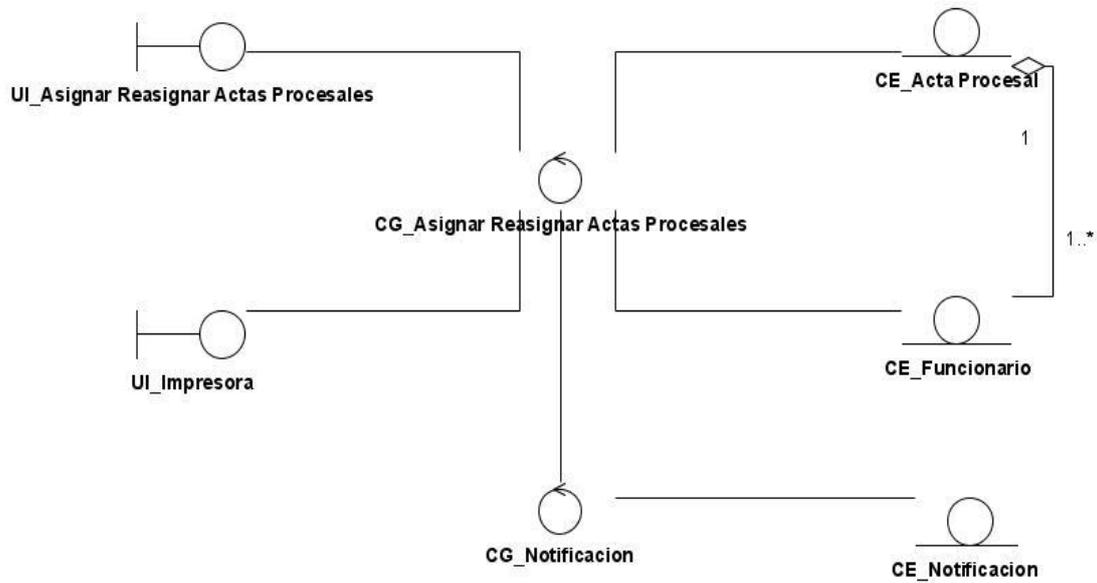
Anexo 10: CU Remitir Actas Procesales a Fiscalía. Descripción reducida.

<i>Caso de Uso</i> : Remitir Actas Procesales a Fiscalía	
<i>Actores</i>	Aprobador.
<i>Nivel</i>	Sub función.
<i>Objetivo</i>	Remitir un Acta Procesal, concluida y aprobada a Fiscalía.
<i>Resumen</i>	El caso de uso inicia una vez que se revisaron todas las diligencias de un Acta Procesal preliminar o definitiva. El actor tiene la opción de remitir el expediente a Fiscalía y el sistema cambia el estado del Acta Procesal a Remitida, terminando así el caso de uso.

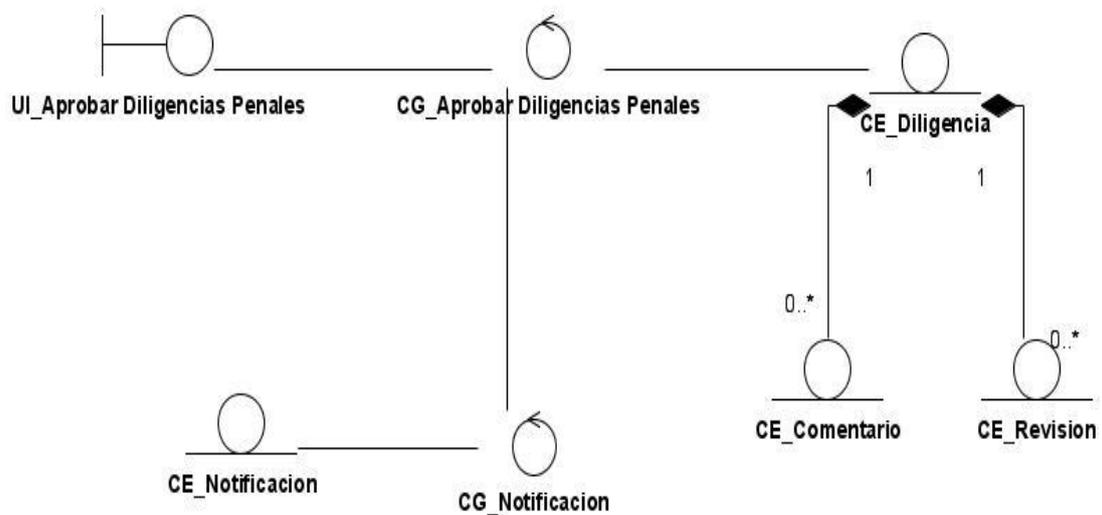
Anexo 11: CU Revisar Diligencias Penales. Descripción reducida.

<i>Caso de Usos</i> : Revisar Diligencias Penales.	
<i>Actores</i>	Revisor.
<i>Nivel</i>	Usuario.
<i>Objetivo</i>	Revisar una o más Diligencias Penales.
<i>Resumen</i>	El caso de uso inicia cuando el actor accede a la bandeja Revisiones desde la Agenda de Trabajo. El sistema muestra un listado de las diligencias que tiene registradas el Acta Procesal en estado "Emitido", brindando la posibilidad de aceptarla, rechazarla o ver su contenido. En caso de que el actor seleccione la opción de aceptar la diligencia, el sistema acepta la diligencia y notifica al responsable de la misma. En caso de que el actor seleccione la opción de rechazar la diligencia el sistema brinda la posibilidad de que se introduzca una justificación para el rechazo. En caso de que el actor seleccione la opción de ver el contenido de la diligencia, el sistema muestra el contenido de la misma, terminando así el caso de uso.

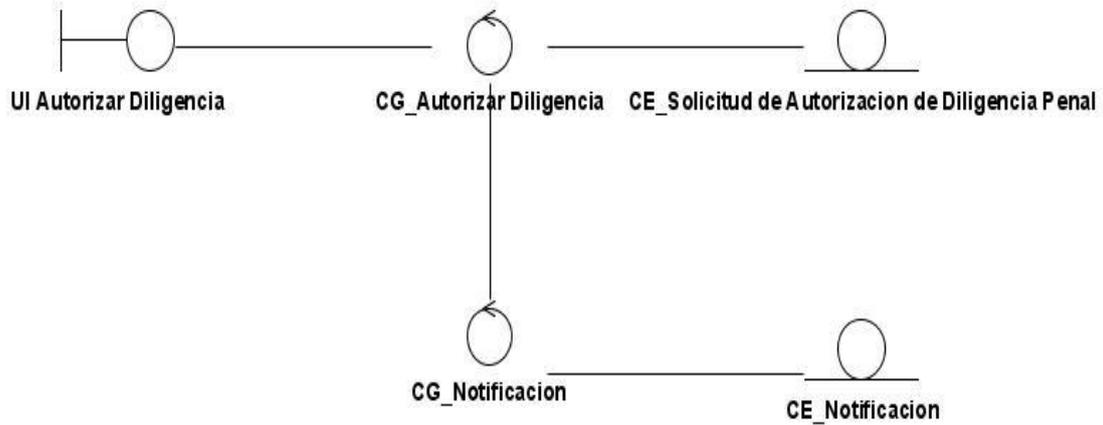
Anexo 12: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Análisis.



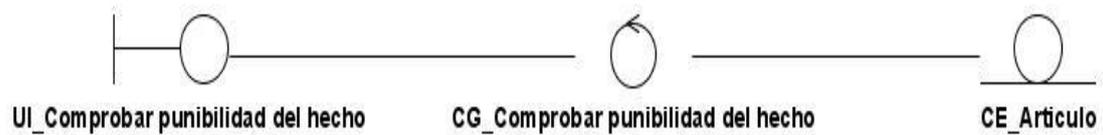
Anexo 13: CU Aprobar Diligencias Penales. Diagrama de Clases del Análisis.



Anexo 14: CU Autorizar Diligencias. Diagrama de Clases del Análisis.



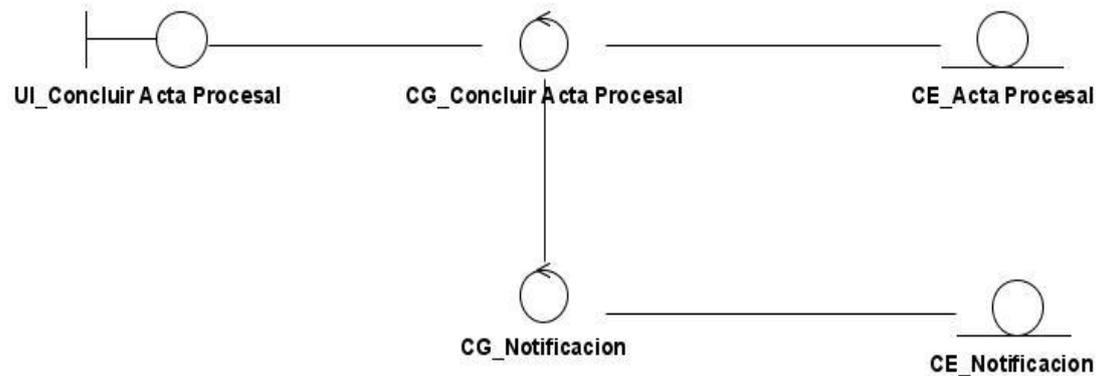
Anexo 15: CU Comprobar Punibilidad del Hecho. Diagrama de Clases del Análisis.



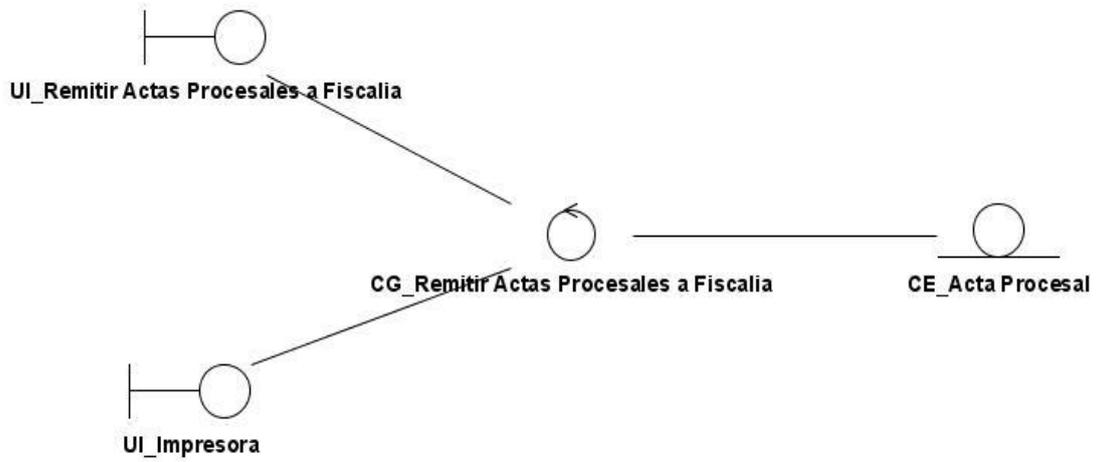
Anexo 16: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Análisis.



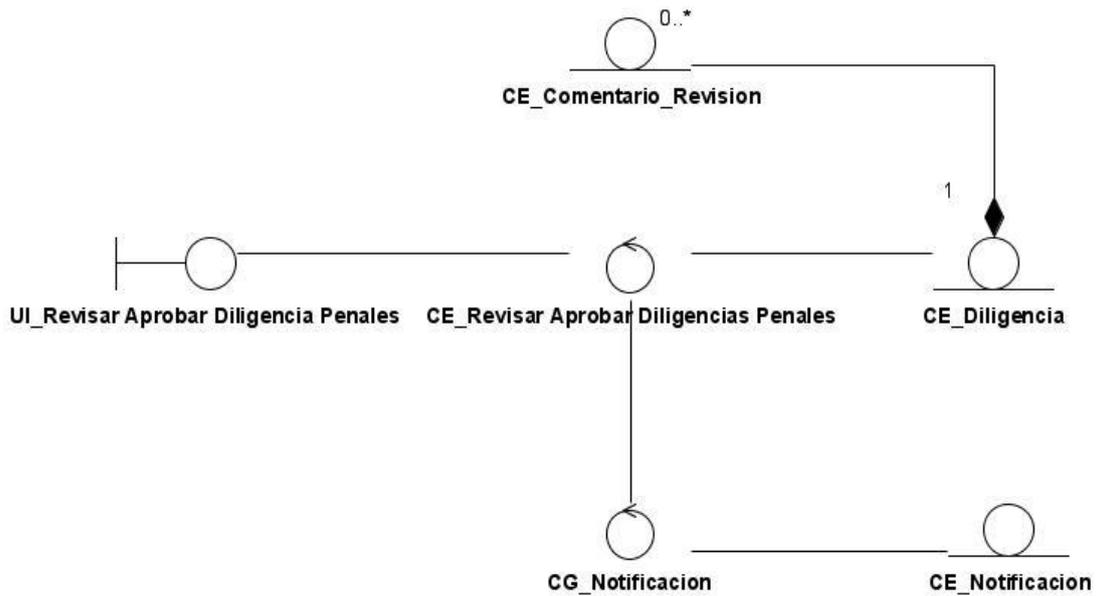
Anexo 17: CU Concluir Acta Procesal. Diagrama de Clases del Análisis.



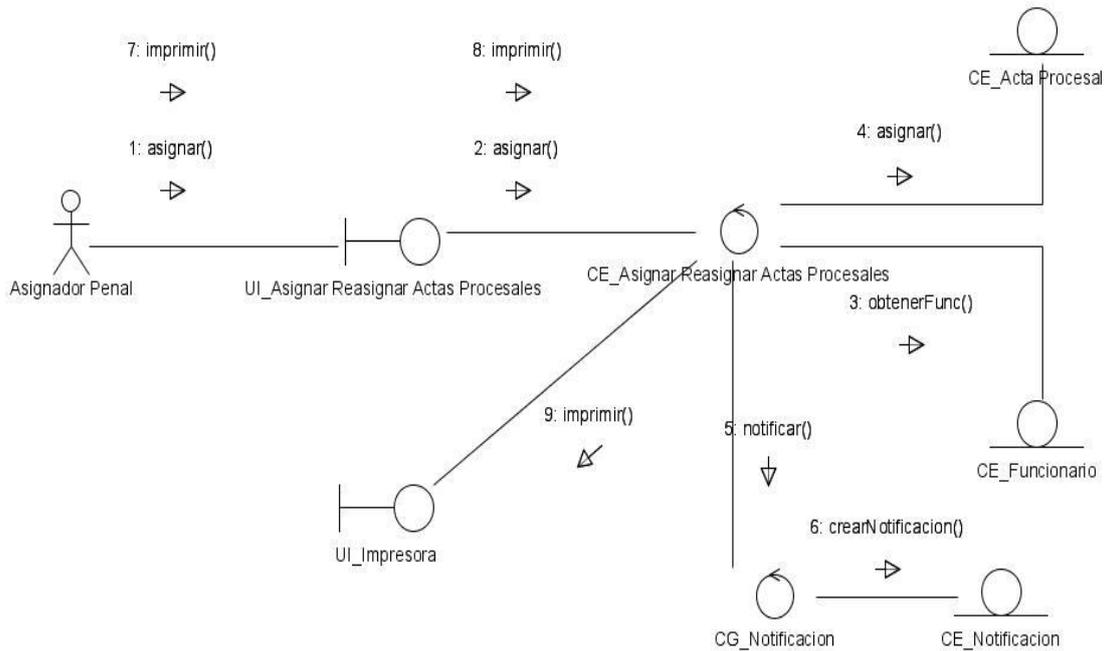
Anexo 18: CU Remitir Acta Procesales. Diagrama de Clases del Análisis.



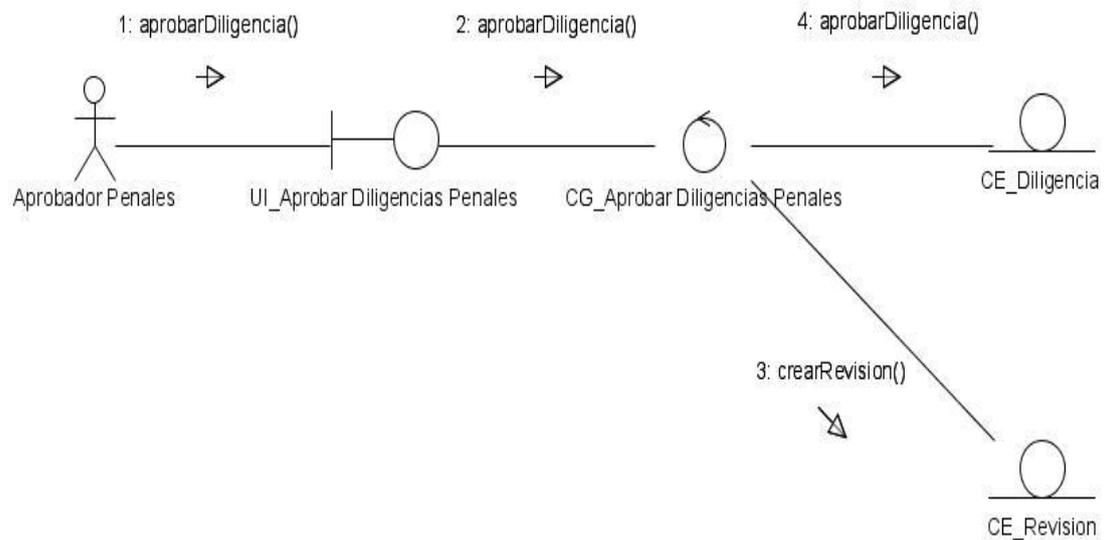
Anexo 19: CU Revisar Diligencias Penales. Diagrama de Clases del Análisis.



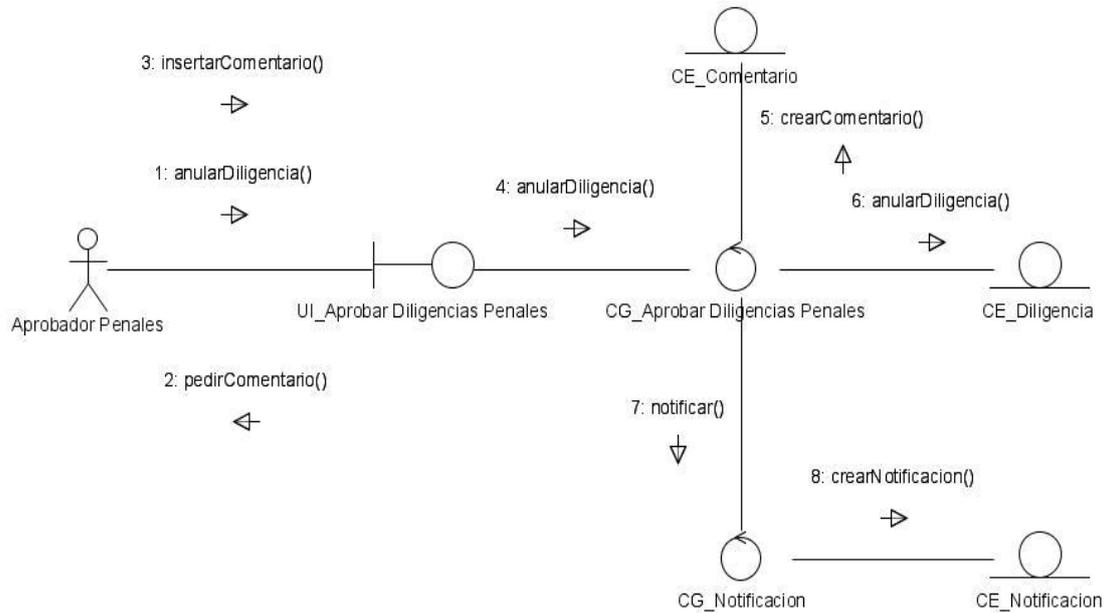
Anexo 20: CU Asignar / Reasignar Actas Procesales. Diagrama de Colaboración del Análisis.



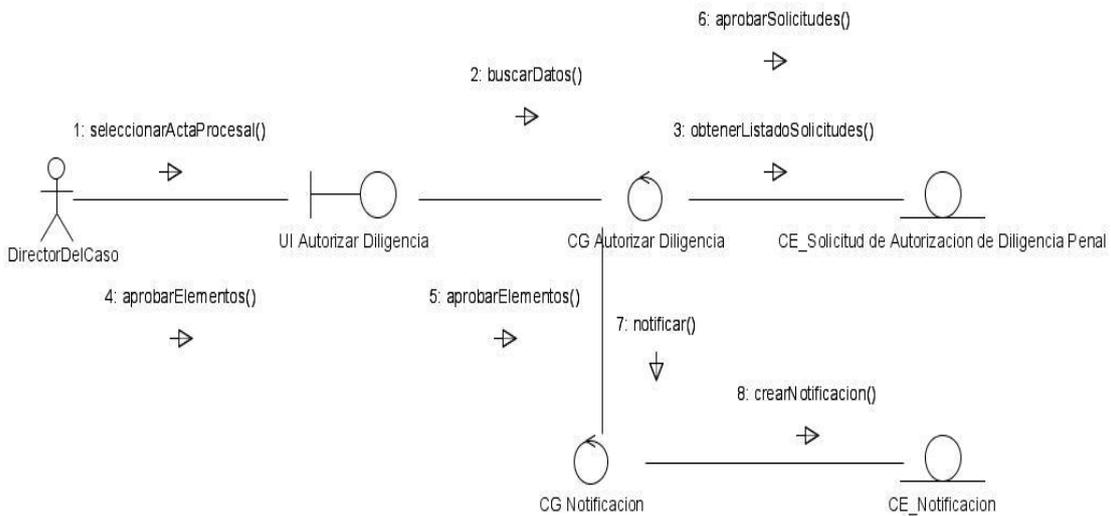
Anexo 21: CU Aprobar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Aprobar.



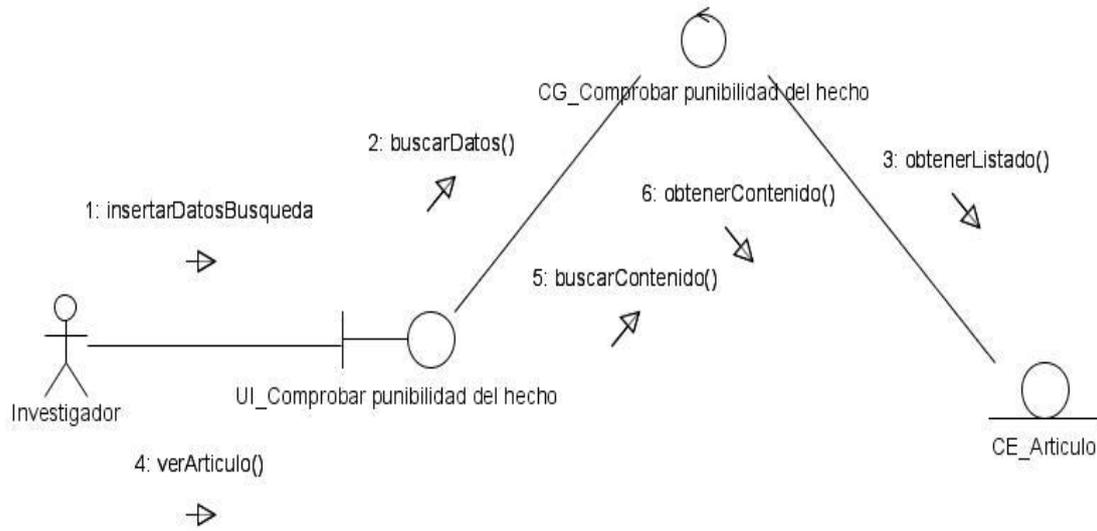
Anexo 22: CU Aprobar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Anular.



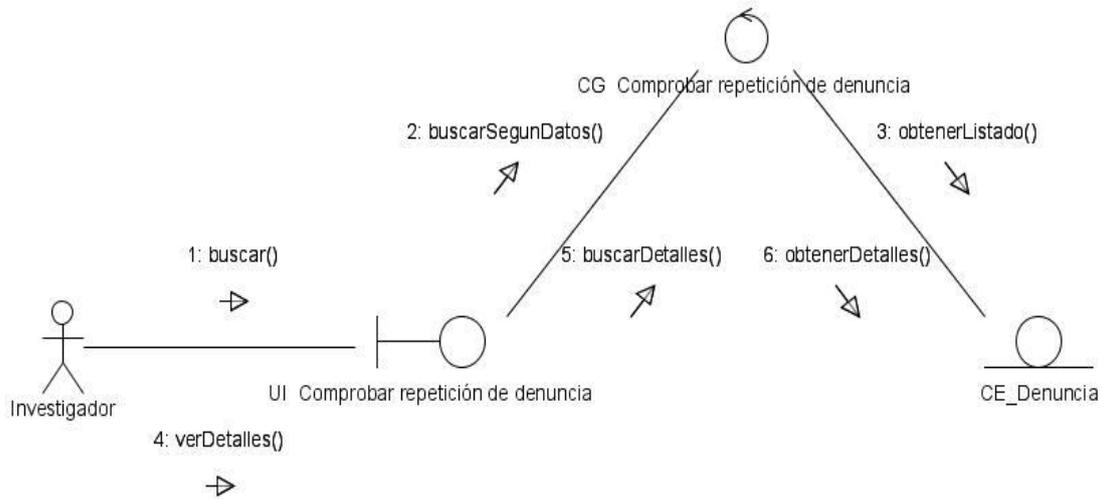
Anexo 23: CU Autorizar Diligencias. Diagrama de Colaboración del Análisis.



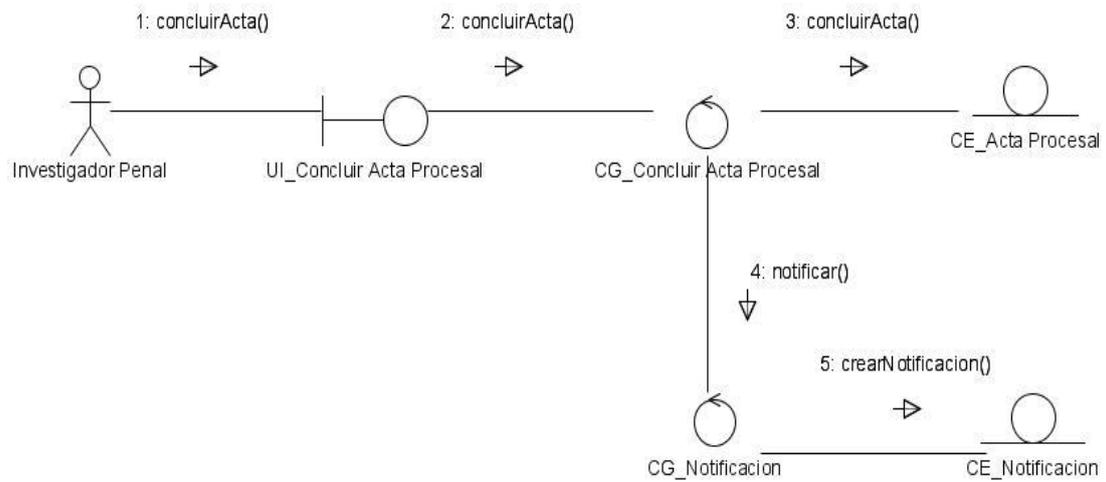
Anexo 24: CU Comprobar Punibilidad del Hecho. Diagrama de Colaboración del Análisis.



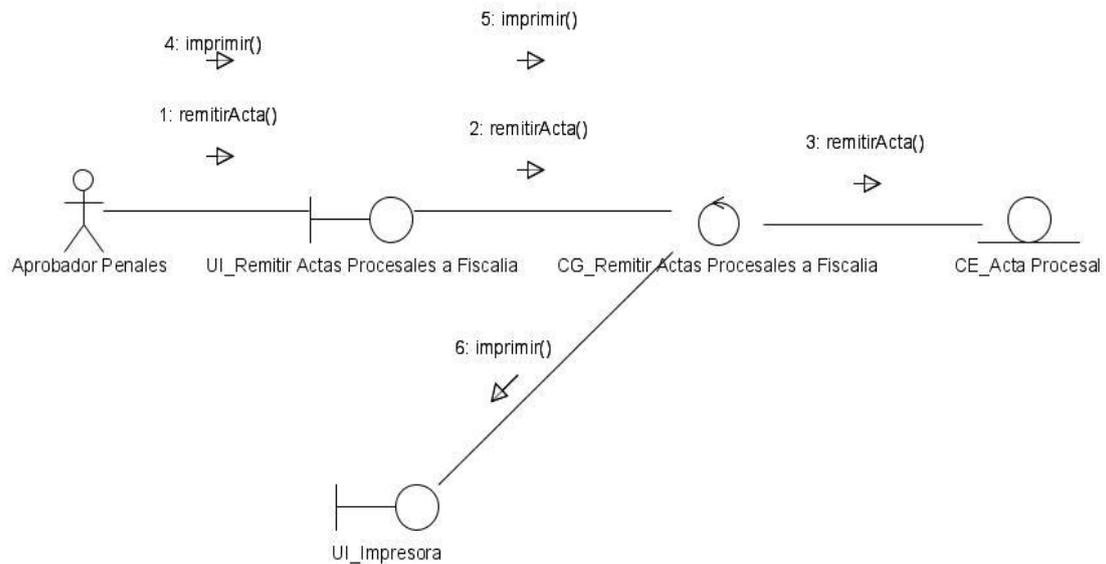
Anexo 25: CU Comprobar Repetición de Denuncia. Diagrama de Colaboración del Análisis.



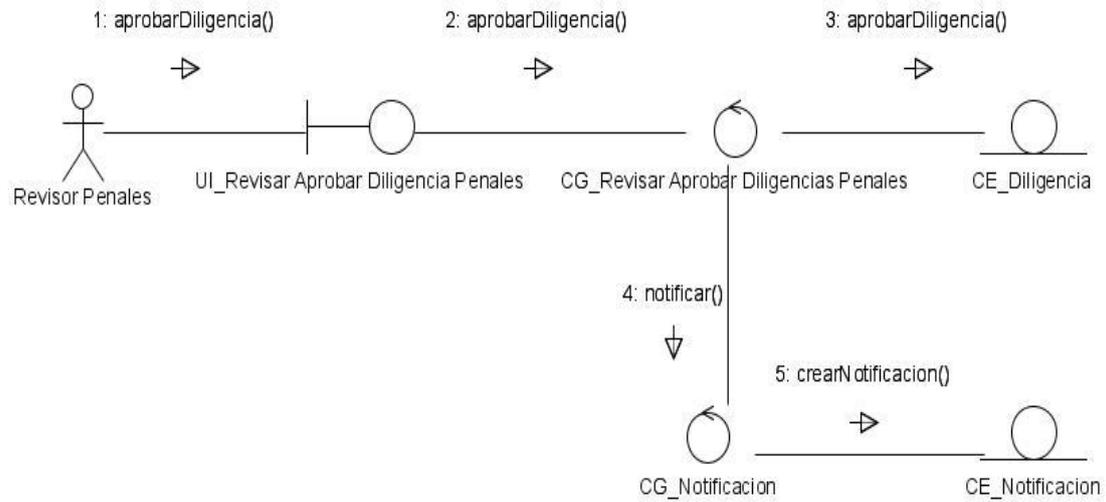
Anexo 26: CU Concluir Acta Procesal. Diagrama de Colaboración del Análisis.



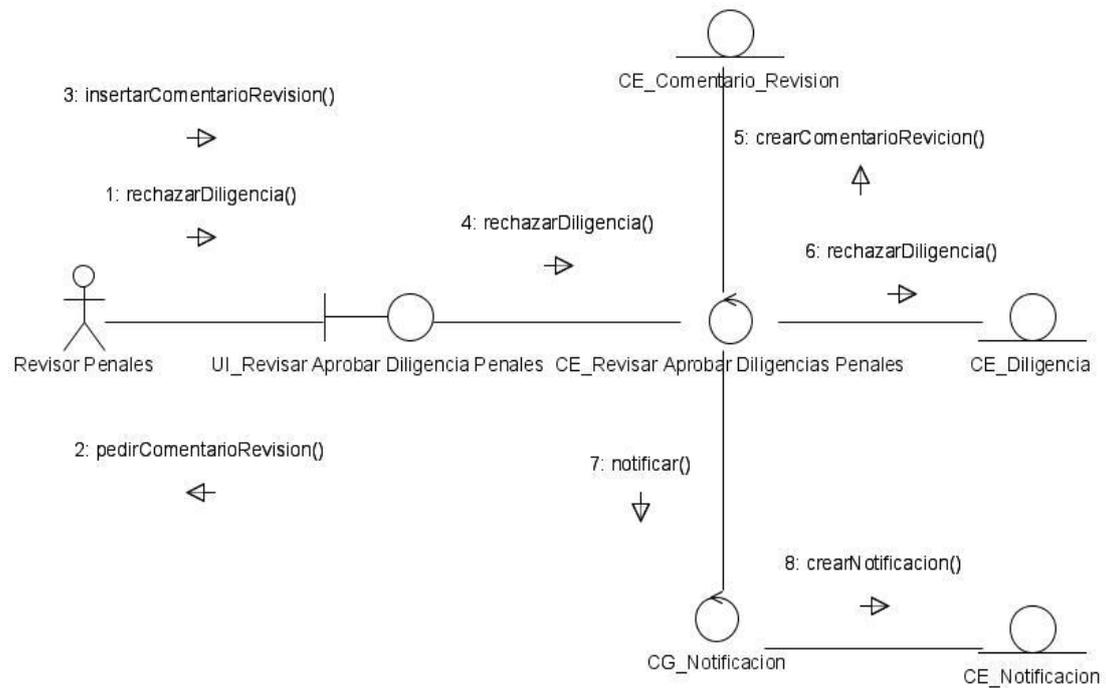
Anexo 27: CU Remitir Actas Procesales. Diagrama de Colaboración del Análisis.



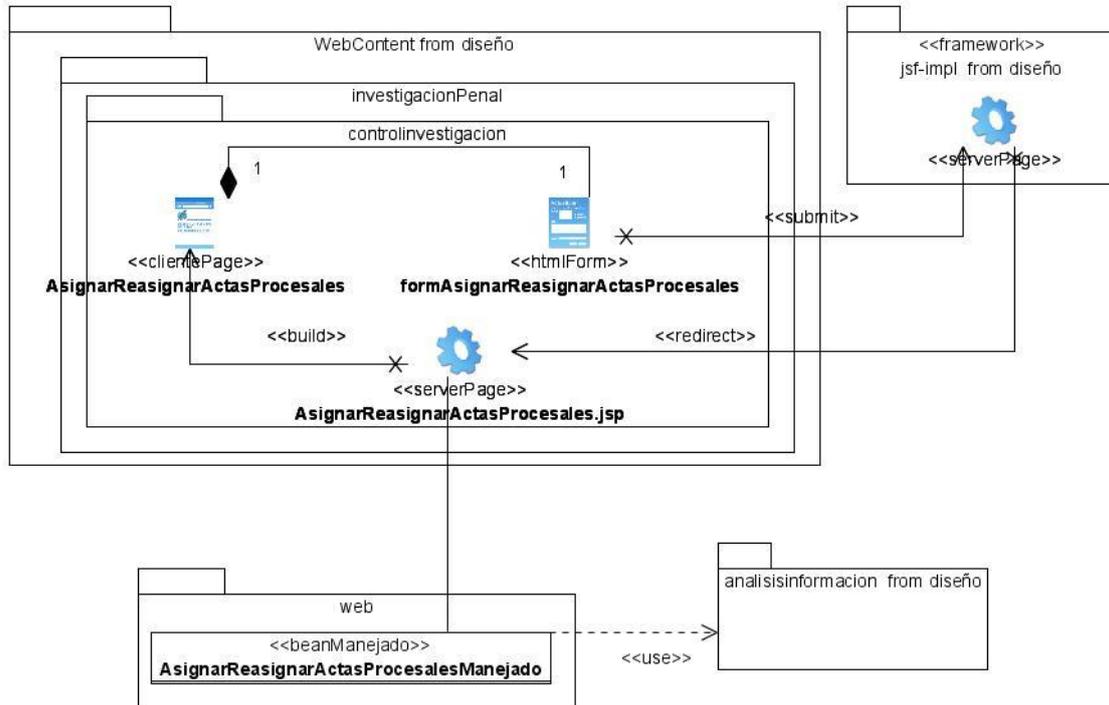
Anexo 28: CU Revisar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Aprobar.



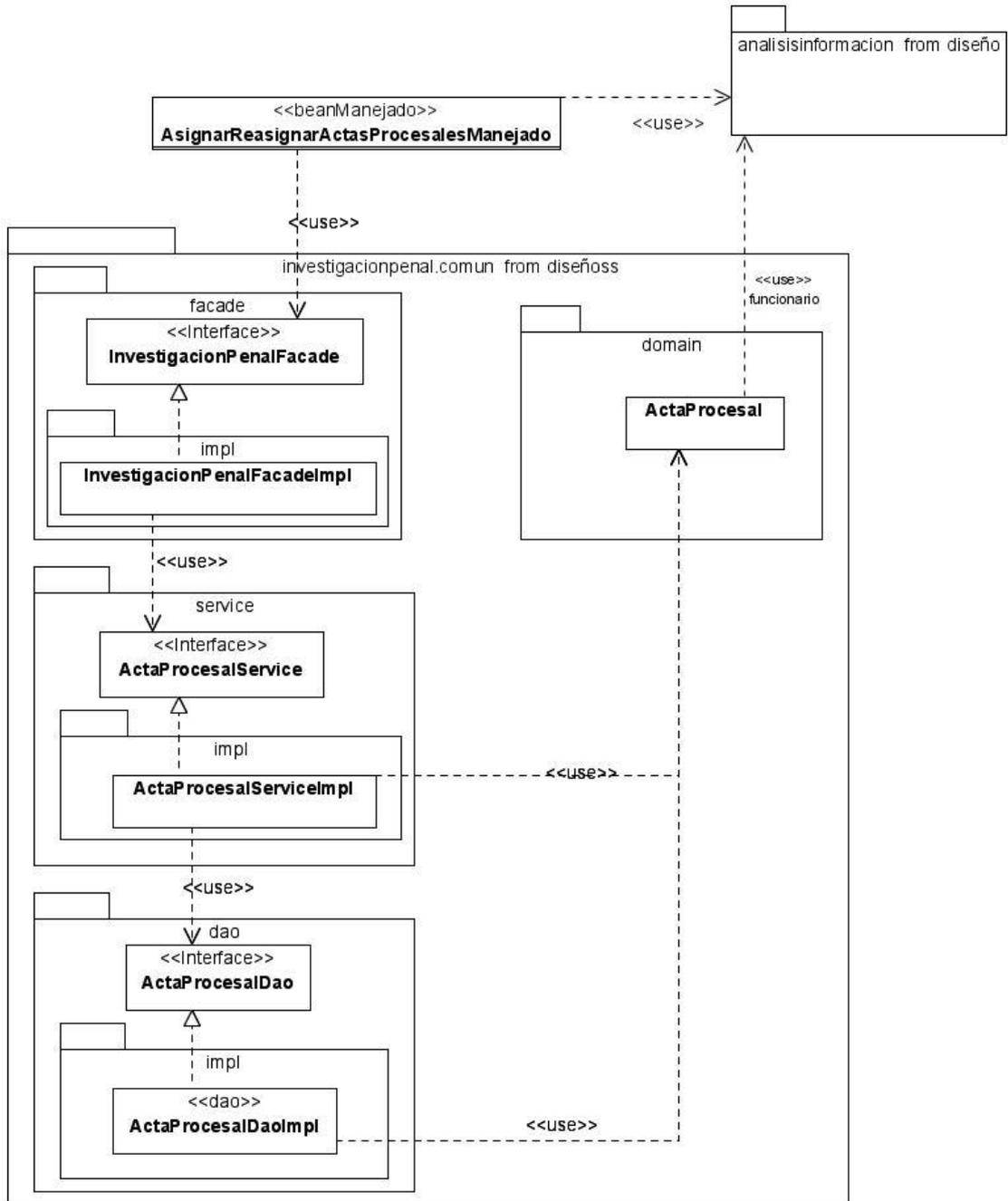
Anexo 29: CU Revisar Diligencias Penales. Diagrama de Colaboración del Análisis. Escenario Rechazar.



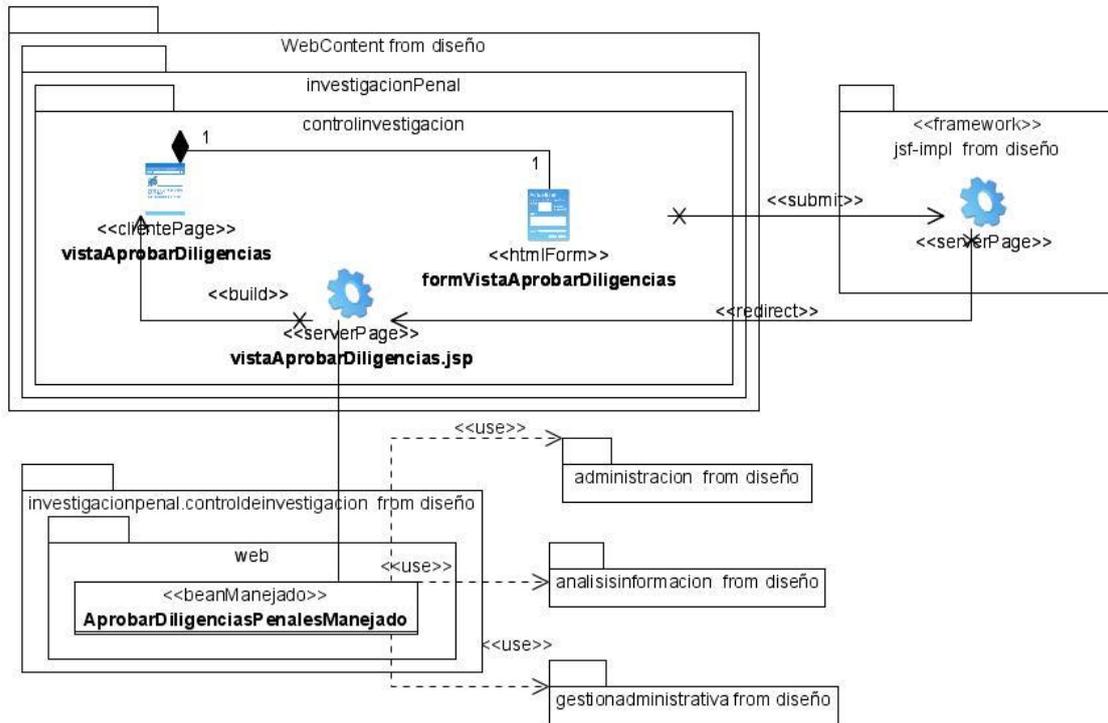
Anexo 30: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Diseño. Primera Parte.



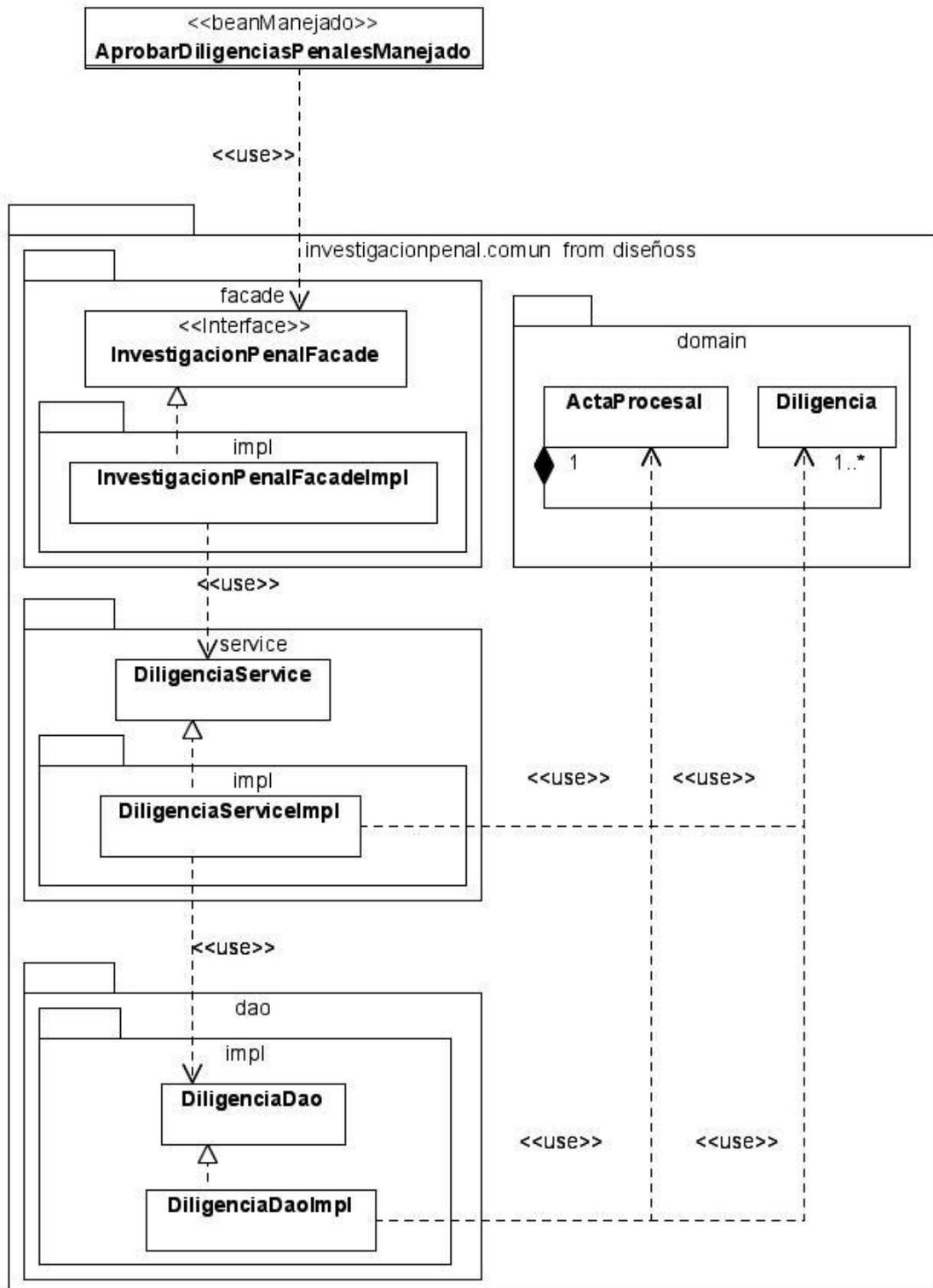
Anexo 31: CU Asignar / Reasignar Actas Procesales. Diagrama de Clases del Diseño. Segunda Parte.



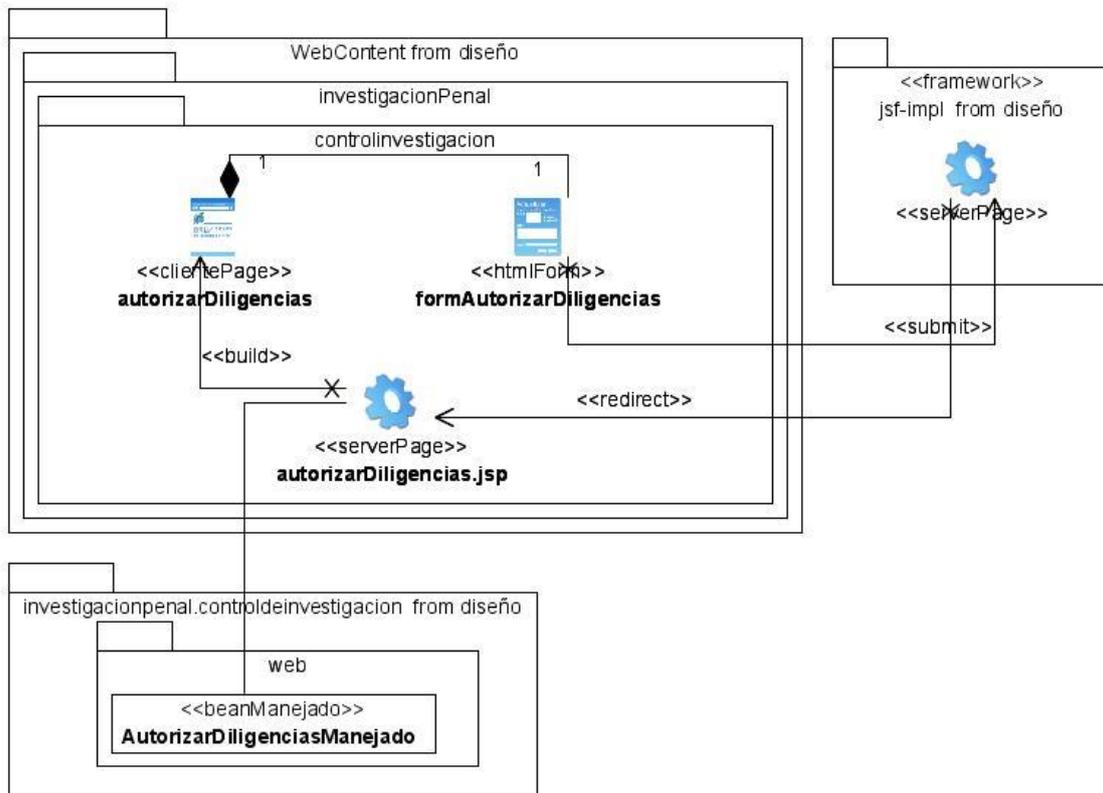
Anexo 32: CU Aprobar Diligencias Penales. Diagrama de Clases del Diseño. Primera Parte.



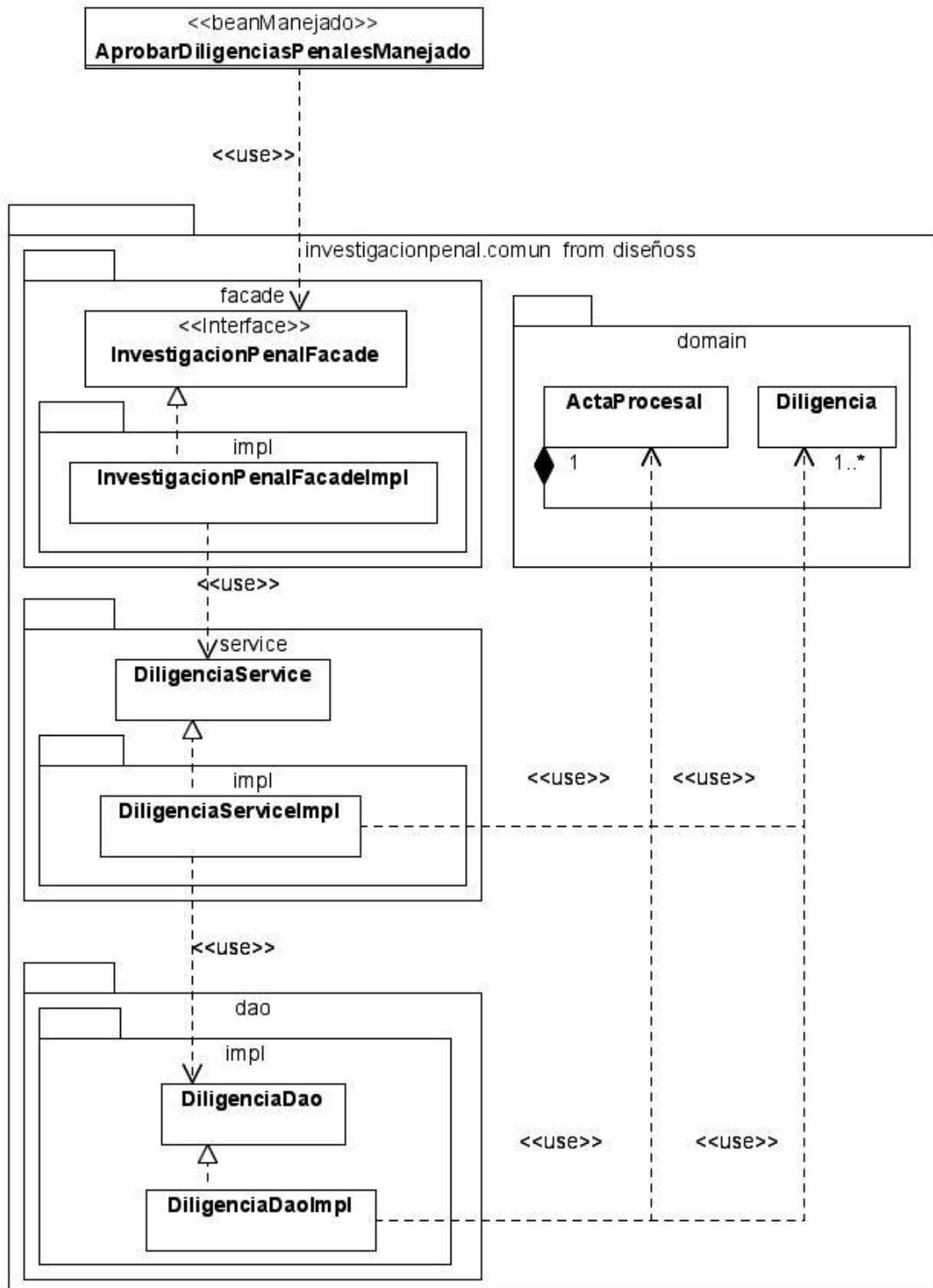
Anexo 33: CU Aprobar Diligencias Penales. Diagrama de Clases del Diseño. Segunda Parte.



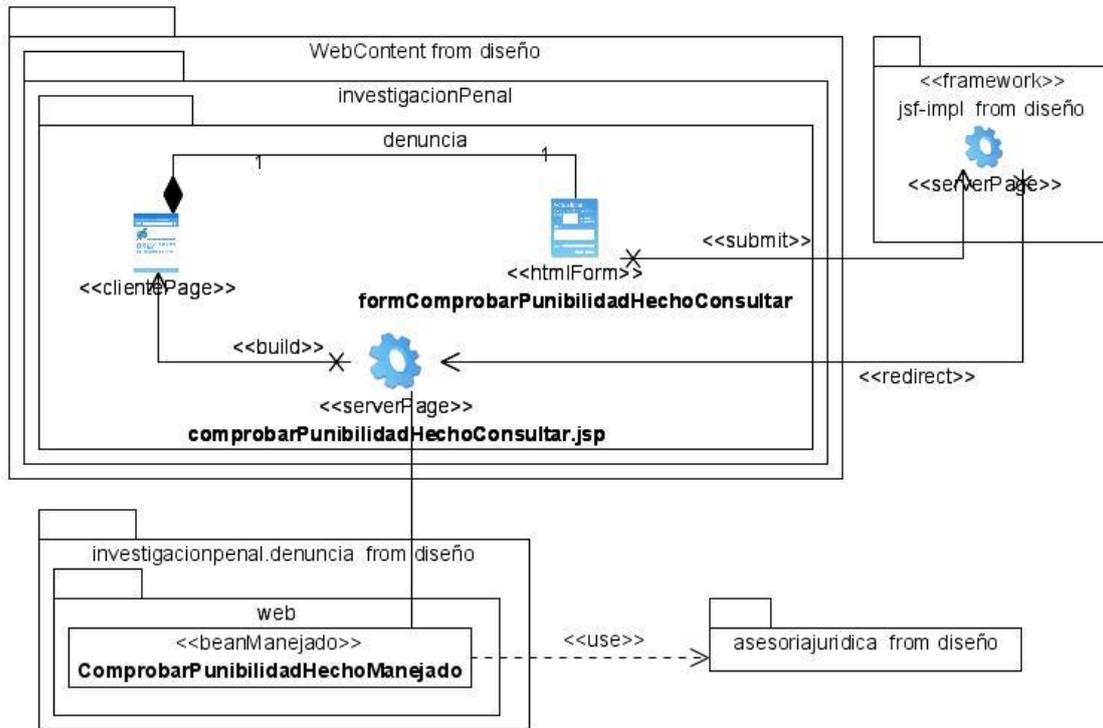
Anexo 34: CU Autorizar Diligencias. Diagrama de Clases del Diseño. Primera Parte.



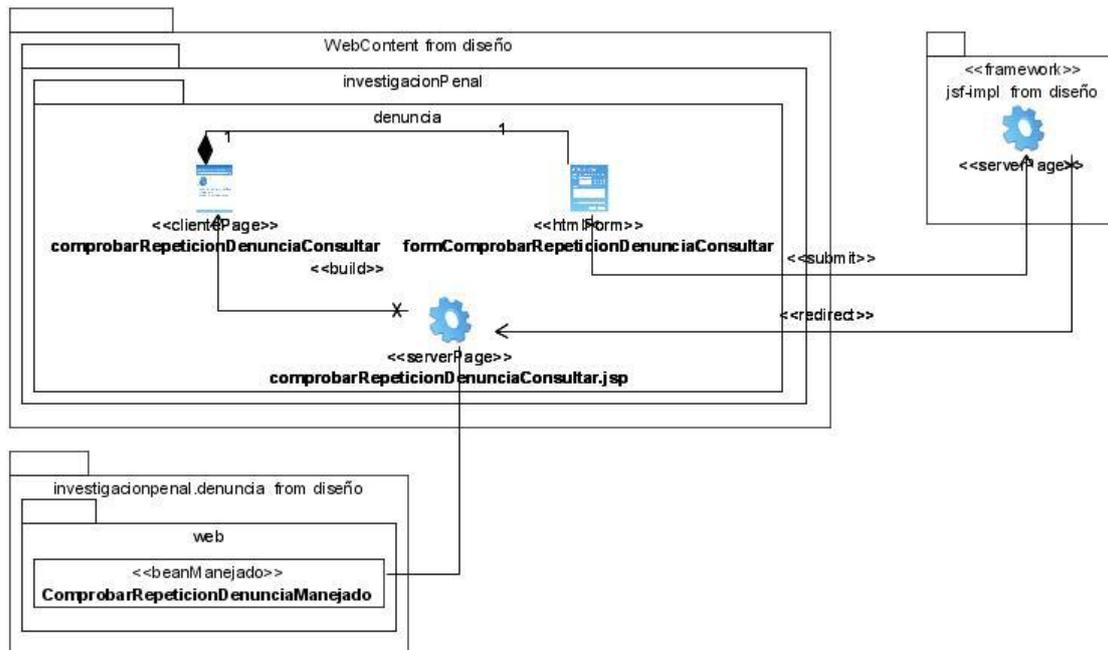
Anexo 35: CU Autorizar Diligencias. Diagrama de Clases del Diseño. Segunda Parte.



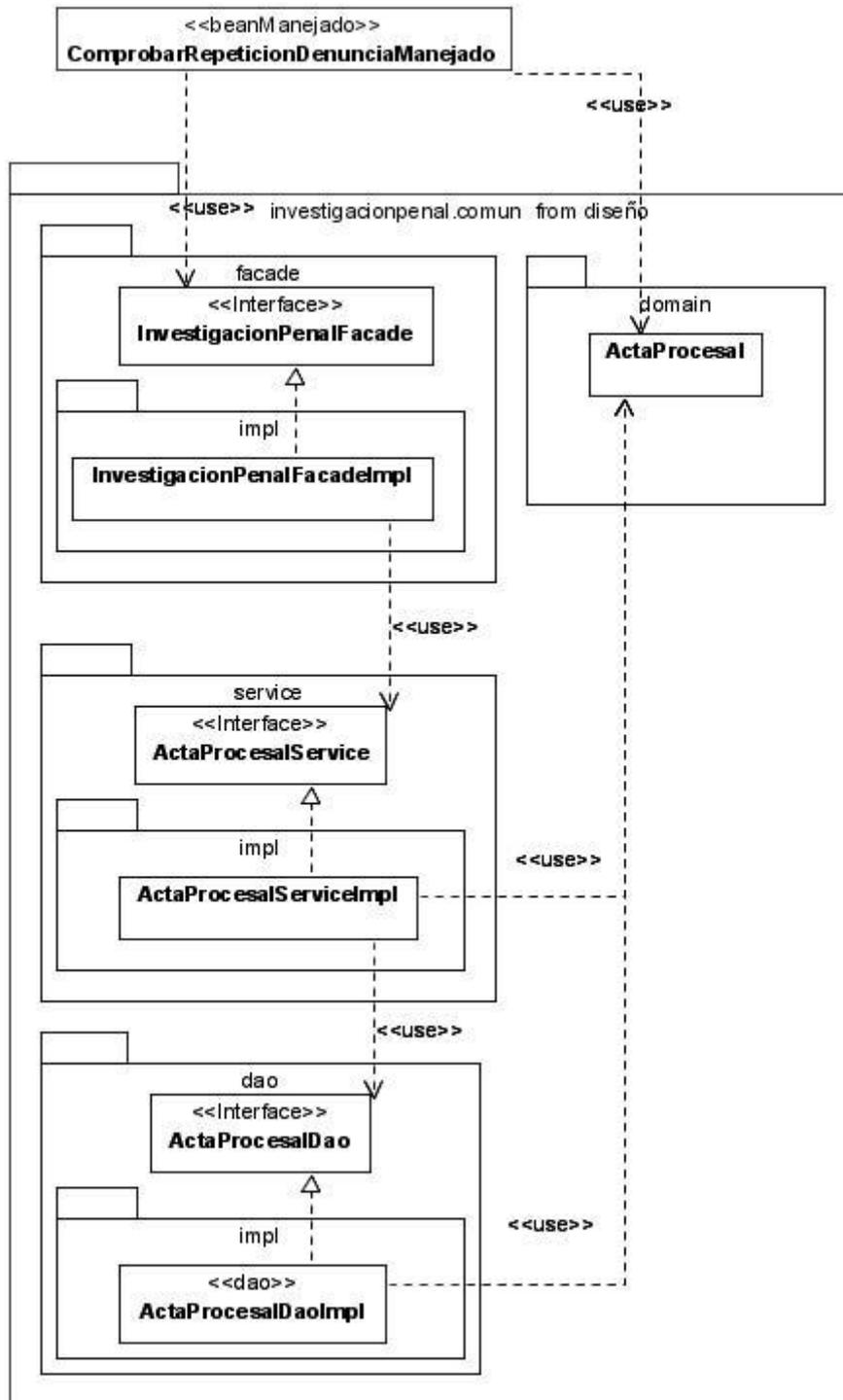
Anexo 36: CU Comprobar Punibilidad del Hecho. Diagrama de Clases del Diseño. Primera Parte.



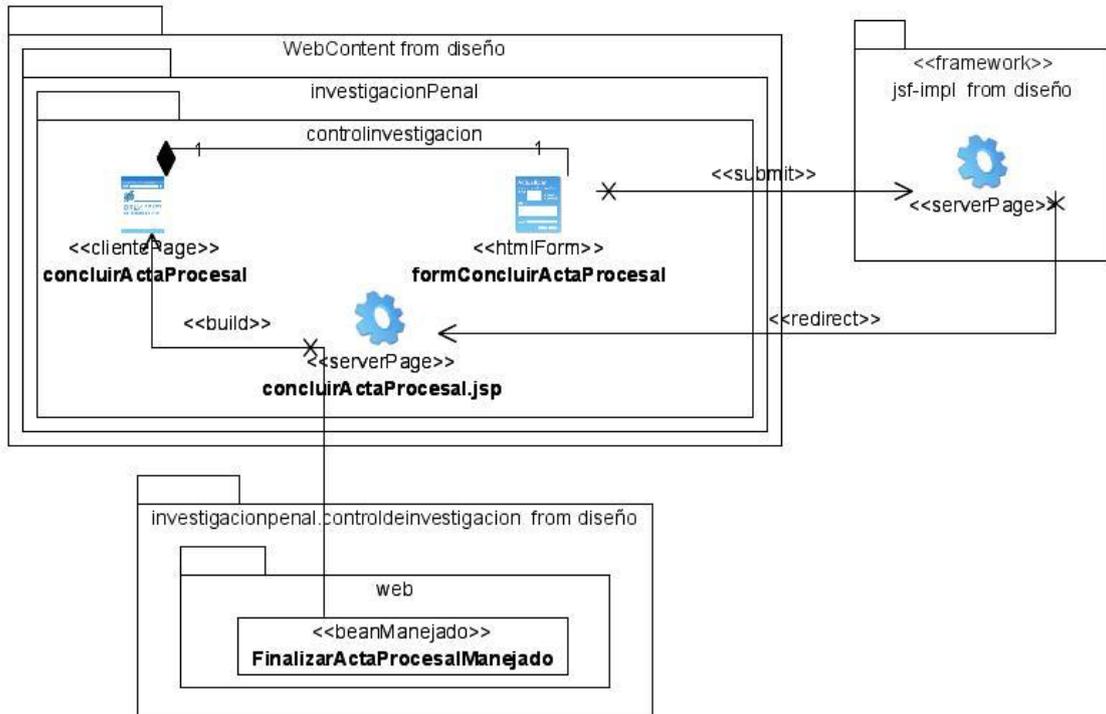
Anexo 37: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Diseño. Primera Parte.



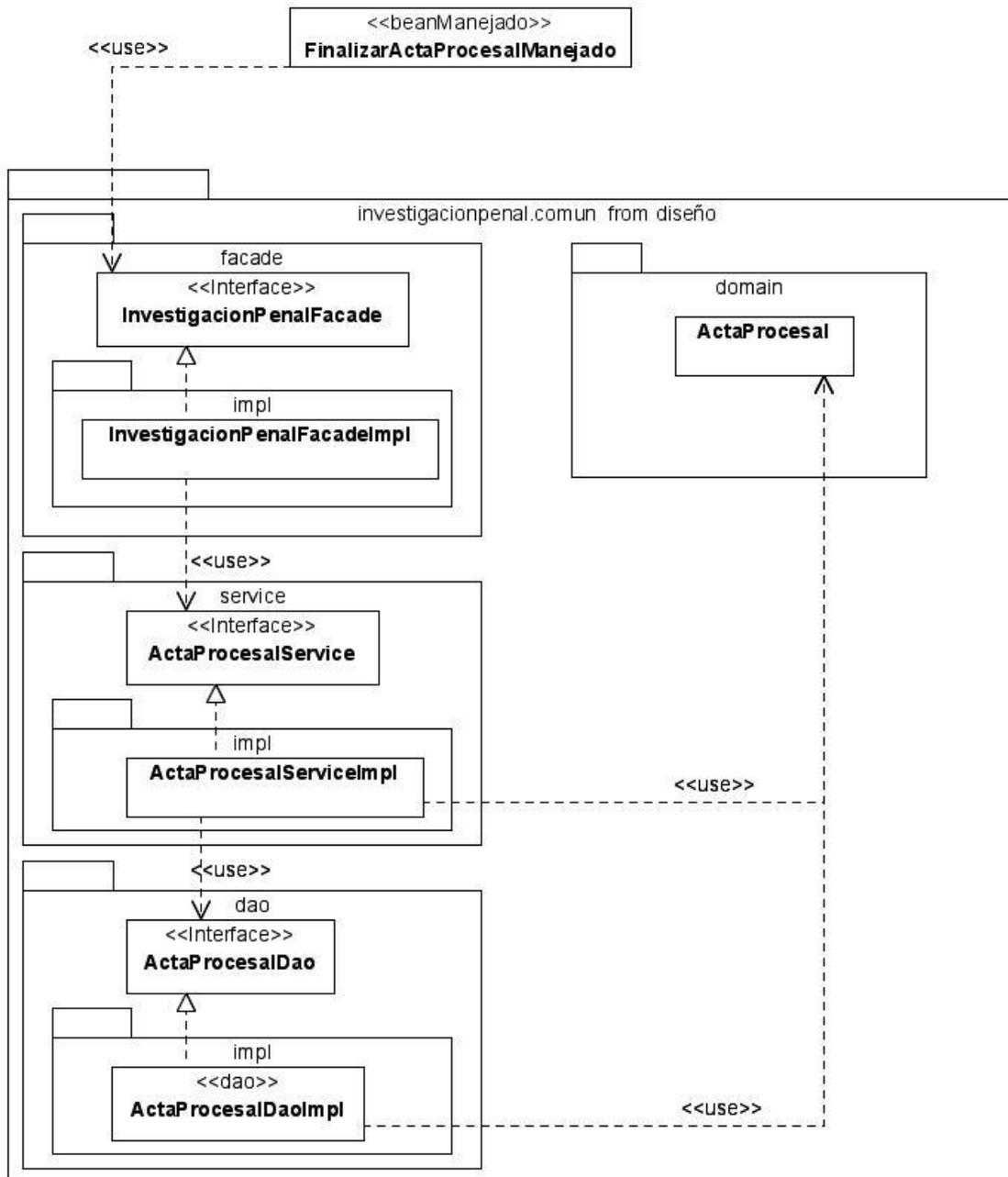
Anexo 38: CU Comprobar Repetición de Denuncia. Diagrama de Clases del Diseño. Segunda Parte.



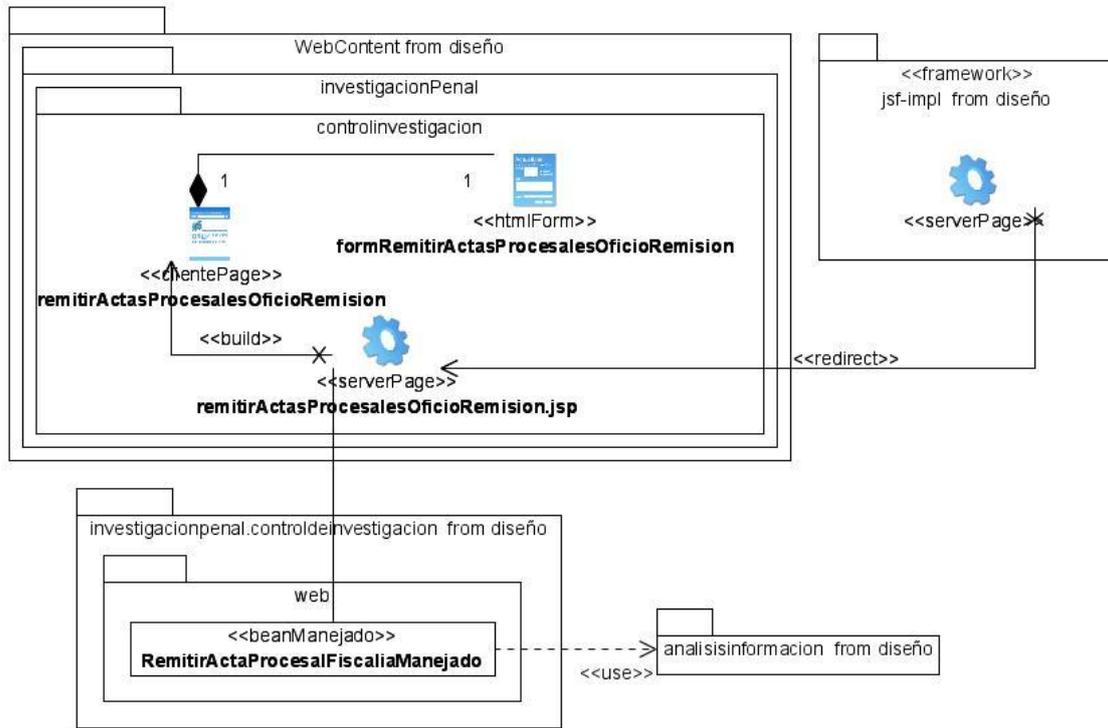
Anexo 39: CU Concluir Acta Procesal. Diagrama de Clases del Diseño. Primera Parte.



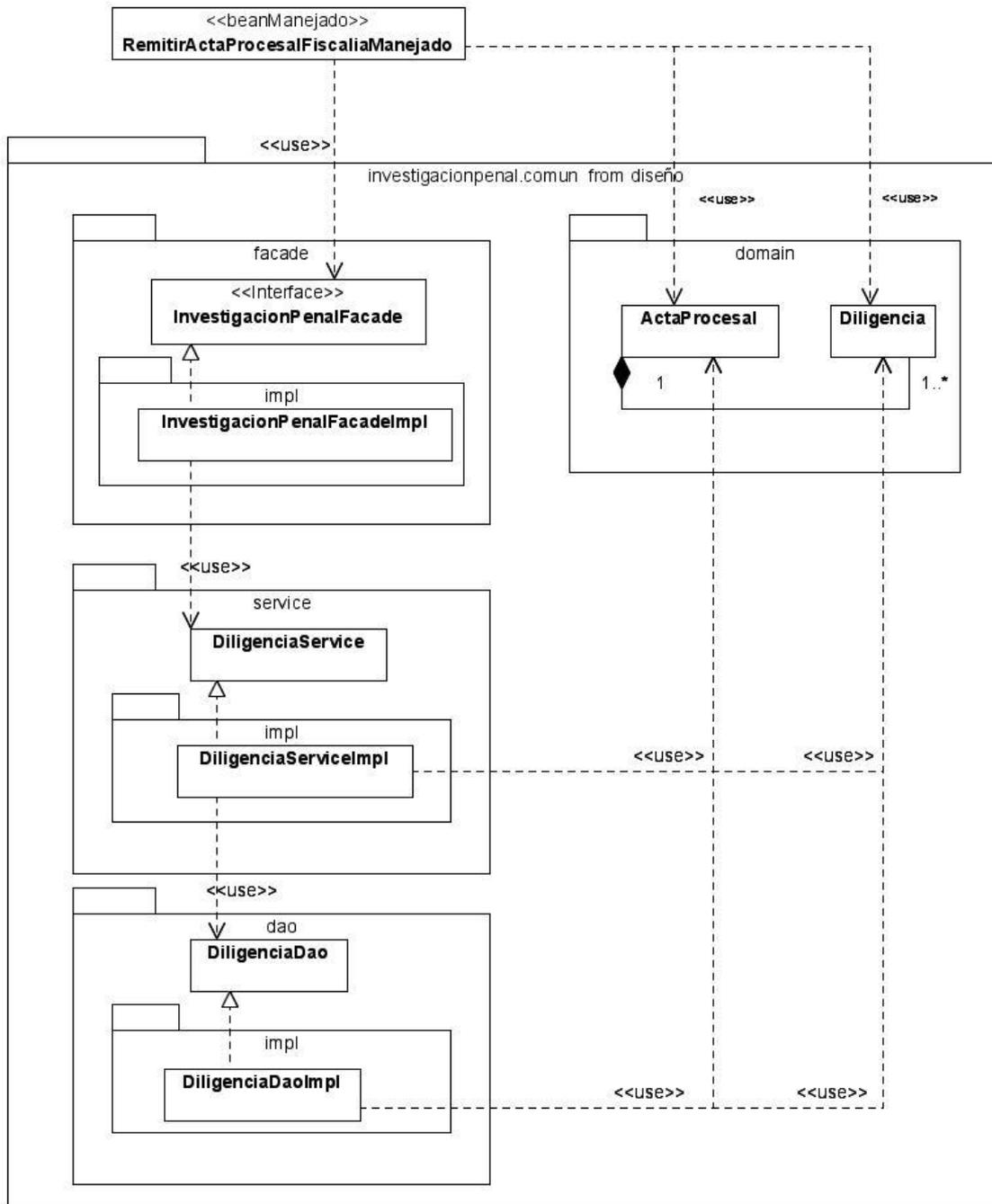
Anexo 40: CU Concluir Acta Procesal. Diagrama de Clases del Diseño. Segunda Parte.



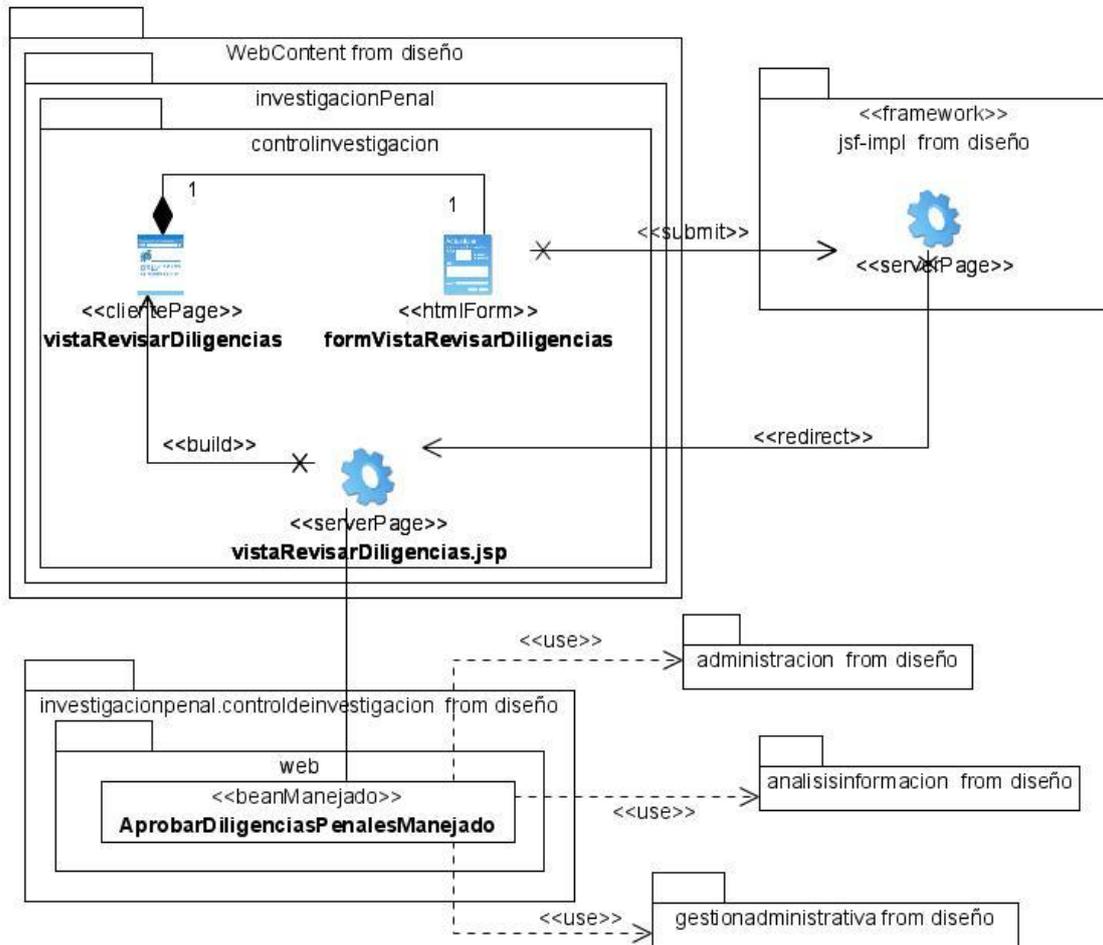
Anexo 41: CU Remitir Actas Procesales. Diagrama de Clases del Diseño. Primera Parte.



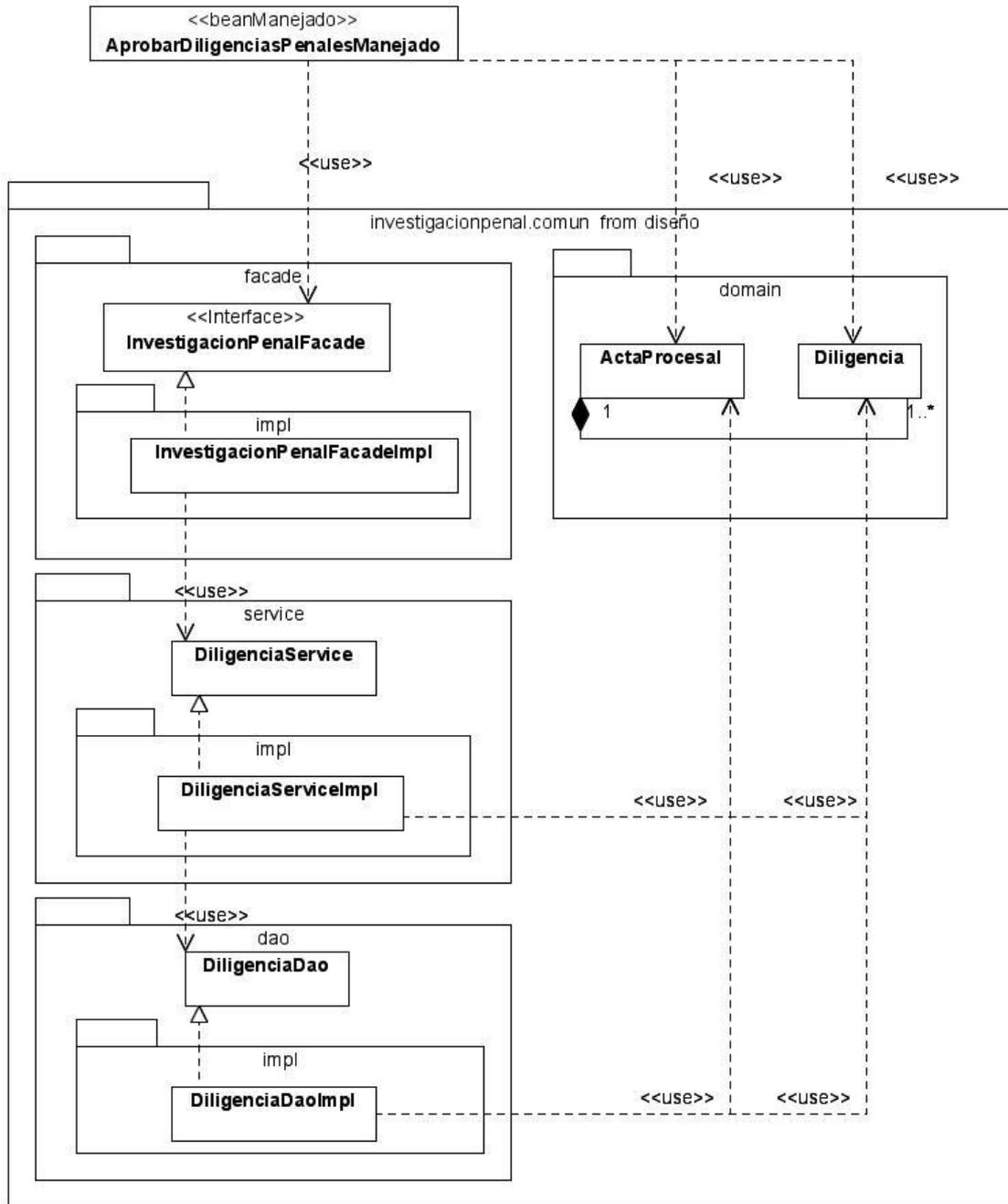
Anexo 42: CU Remitir Actas Procesales. Diagrama de Clases del Diseño. Segunda Parte.



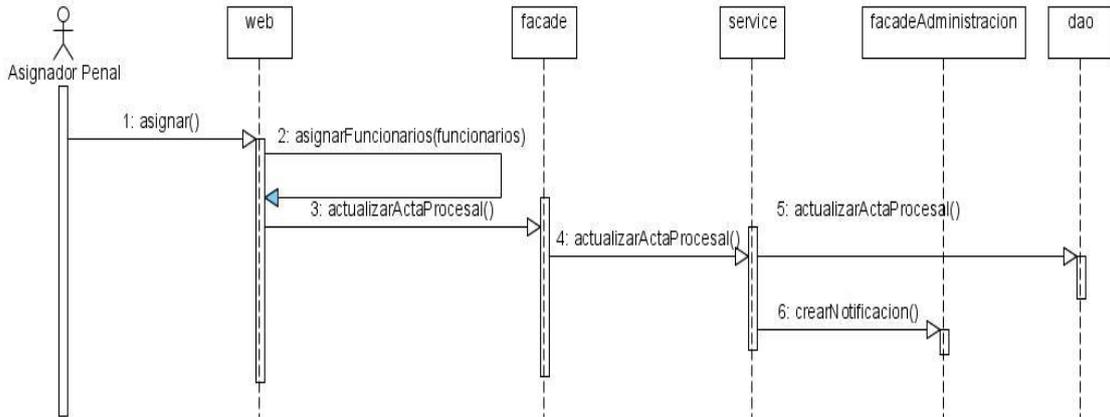
Anexo 43: CU Revisar Diligencias Penales. Diagrama de Clases del Diseño. Primera Parte.



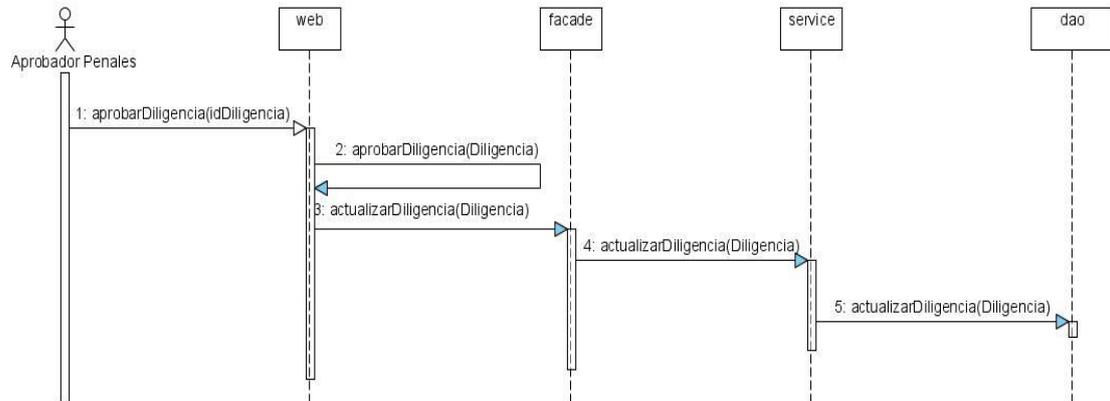
Anexo 44: CU Revisar Diligencias Penales. Diagrama de Clases del Diseño. Segunda Parte.



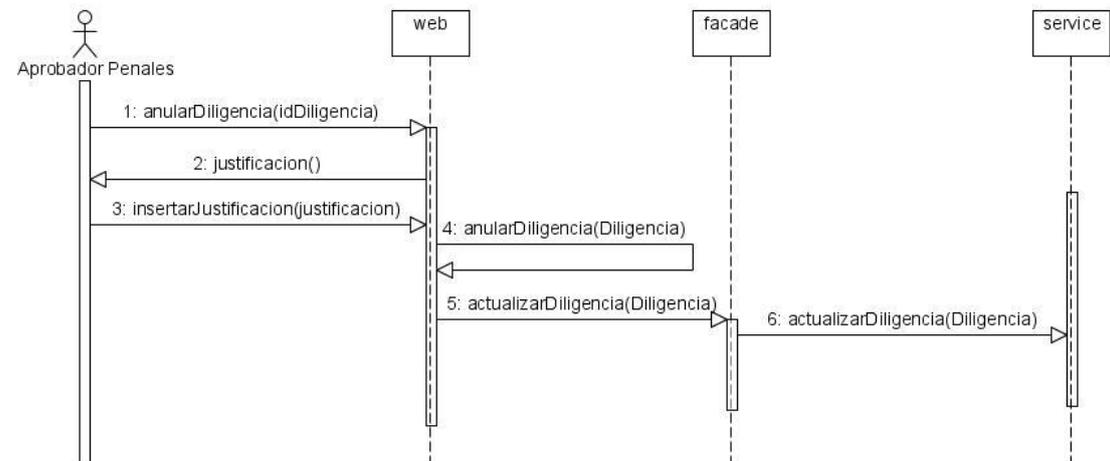
Anexo 45: CU Asignar / Reasignar Actas Procesales. Diagrama de Contratos entre Paquetes.

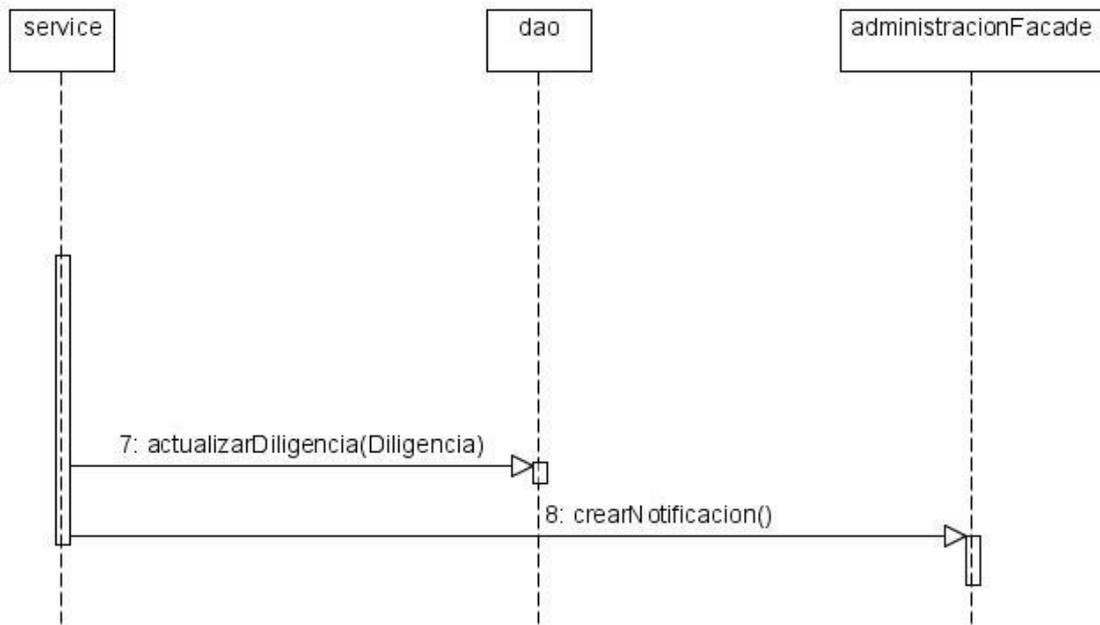


Anexo 46: CU Aprobar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Aprobar.

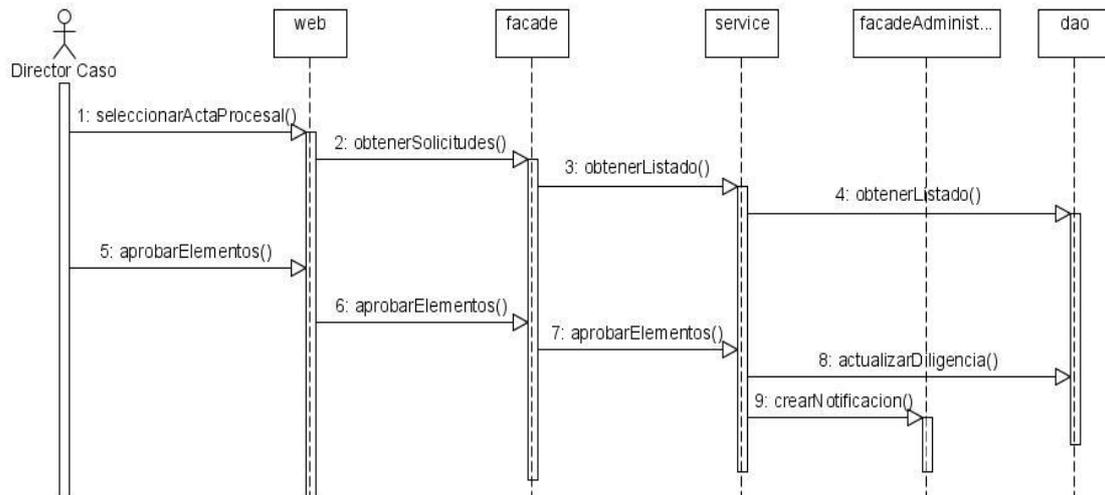


Anexo 47: CU Aprobar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Anular.

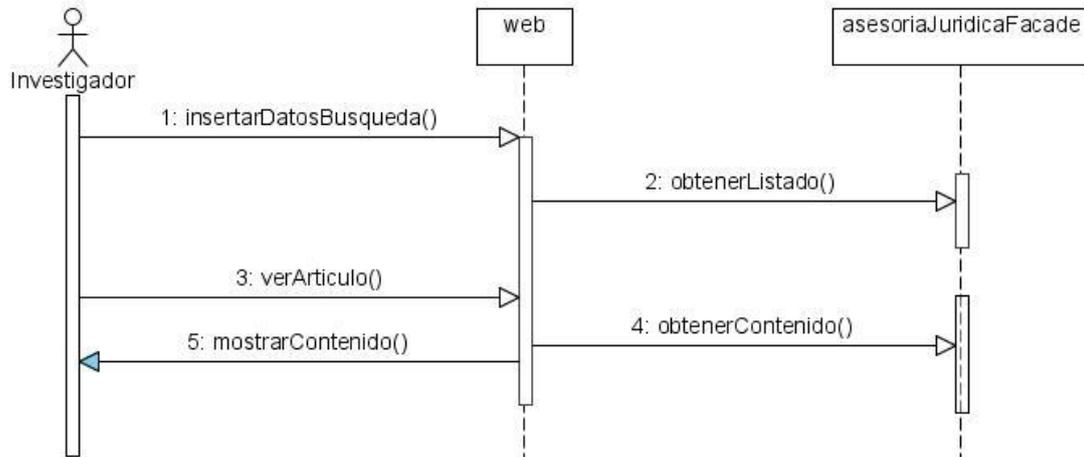




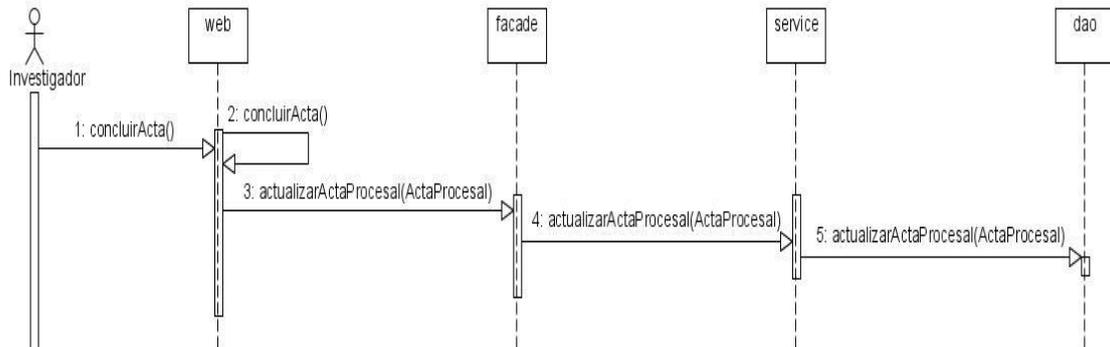
Anexo 48: CU Autorizar Diligencia. Diagrama de Contratos entre Paquetes.



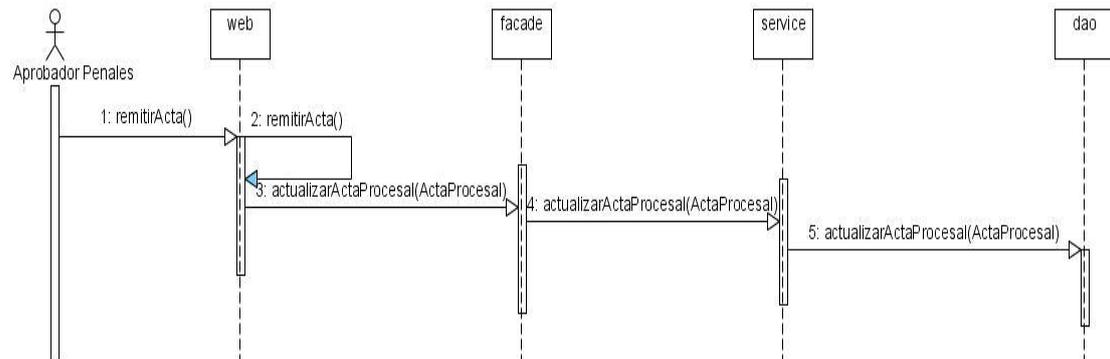
Anexo 49: CU Comprobar Punibilidad del Hecho. Diagrama de Contratos entre Paquetes.



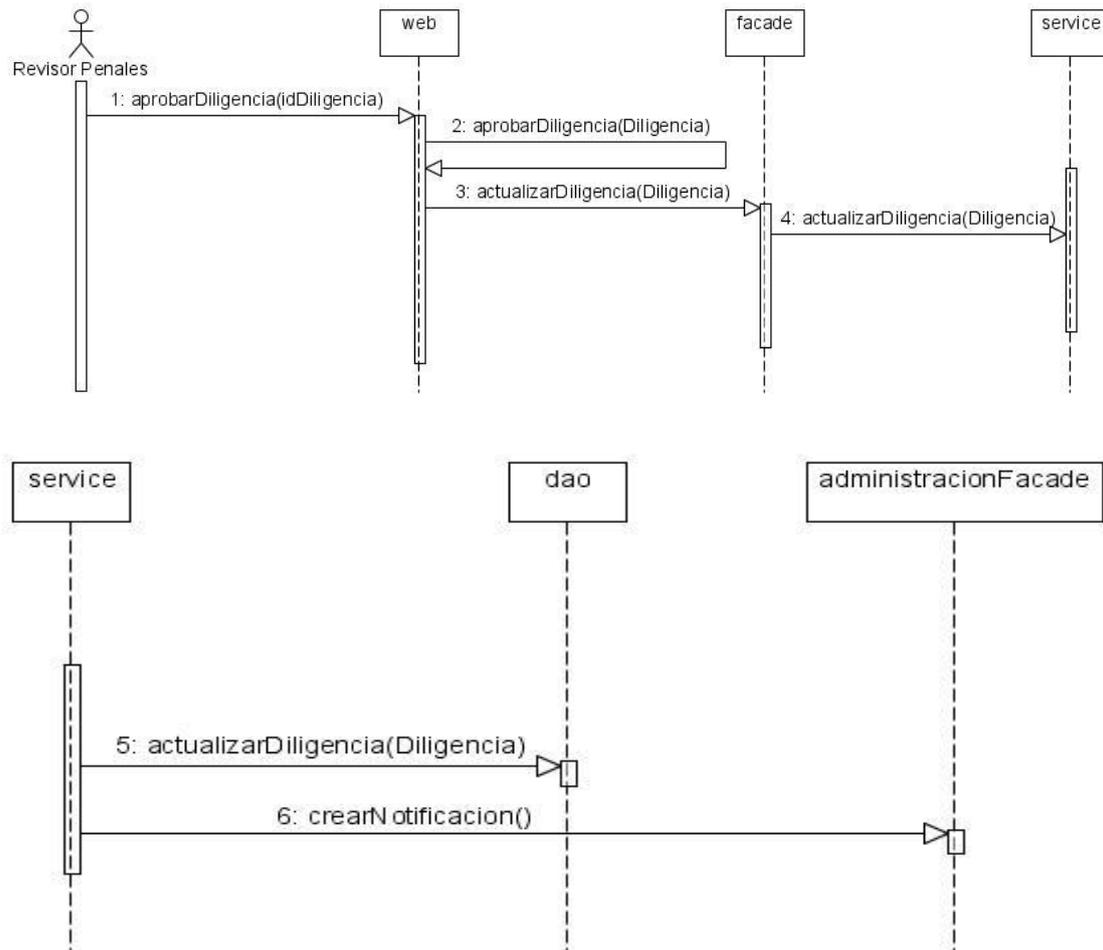
Anexo 50: CU Concluir Acta Procesal. Diagrama de Contratos entre Paquetes.



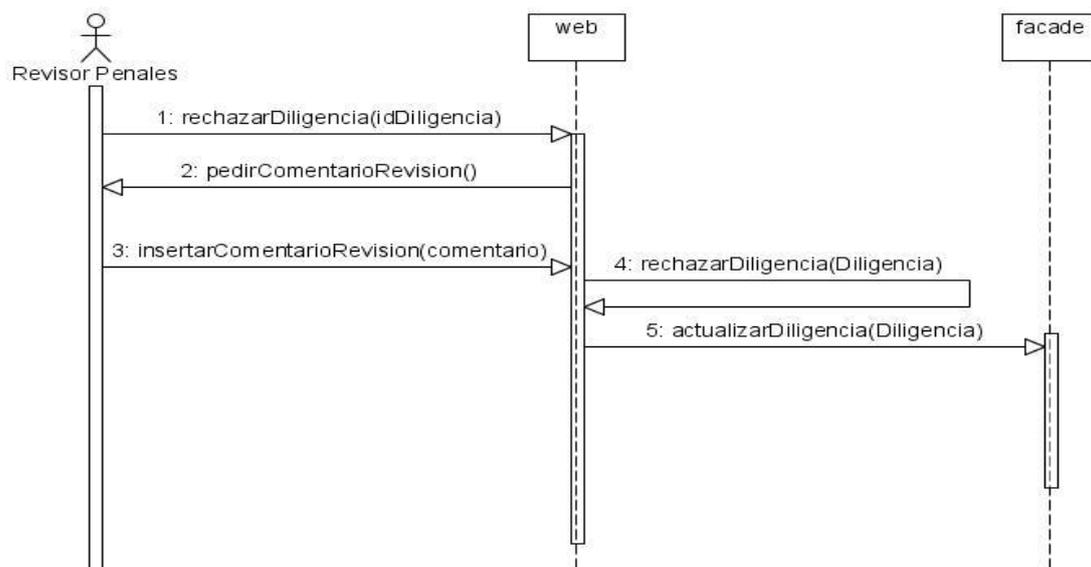
Anexo 51: CU Remitir Actas Procesales. Diagrama de Contratos entre Paquetes.

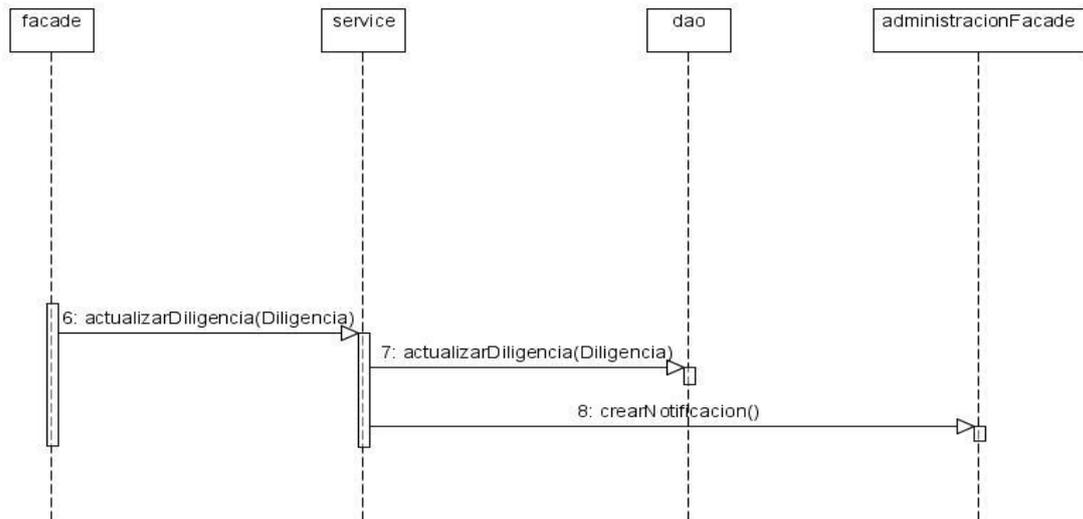


Anexo 52: CU Revisar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Aprobar.



Anexo 53: CU Revisar Diligencias Penales. Diagrama de Contratos entre Paquetes. Escenario Rechazar.





Bibliografía

1. **Alcaldía del Chacao. Venezuela.** Chacao. *Chacao*. [En línea] [Citado el: 1 de Febrero de 2009.] <http://www.chacao.gov.ve/violencia.asp>.
2. **CICPC.** CICPC. *CICPC*. [En línea] [Citado el: 4 de Febrero de 2009.] <http://www.cicpc.gov.ve/>.
3. **Ivar Jacobson, Grandy Booch, James Rumbaugh.** *El Proceso Unificado del Desarrollo del Software*. [trad.] Salvador Sánchez, y otros. s.l. : ADDISON-WESLEY, 1999. 84-7829-036-2.
4. **Rational Software Corporation.** Ayuda del RUP. *Ayuda del RUP*. [En línea] [Citado el: 20 de 4 de 2009.] Suite del Rational 2003..
5. **Núñez Camallea, Noel Luis y Coutin Abalo, Ronald.** *Diccionario de Informática*. Ciudad de la Habana : Cinetífico-Técnica, 2005. 959-05-0391-8.
6. **La Fecha, Diario de Ciencia y Tecnología.** [En línea] [Citado el: 8 de Noviembre de 2008.] <http://www.lafecha.net/canales/softlibre/red-hat-presenta-developer-studio-basado-en-eclipse/>.
7. **Guevara, Humberto Rivero.** Análisis, diseño e implementación del módulo Aprehensión del SIIPOL. *Análisis, diseño e implementación del módulo Aprehensión del SIIPOL*. La Lisa, Ciudad de la Habana, Cuba : s.n., junio de 2008. Trabajo de diploma para optar por el Título de Ing. Informático, correspondiente al rol Diseñador-Programador del proyecto CICPC.
8. **Pereira Ojeda, Maikel y Gago Martinez, Yudanis.** Análisis, Diseño e Implementación del módulo Experticias Criminalísticas del Sistema de Investigación e Información Policial. *Análisis, Diseño e Implementación del módulo Experticias Criminalísticas del Sistema de Investigación e Información Policial*. La Lisa, Ciudad de la Habana, Cuba : s.n., junio de 2008. Trabajo de diploma para optar por el Título de Ing. Informático, correspondiente al rol Diseñador-Programador del proyecto CICPC.
9. **JasperSoft.** JasperSoft. *JasperSoft*. [En línea] [Citado el: 1 de Octubre de 2008.] JasperSoft es un fabricante especializado en soluciones comerciales de código abierto. http://www.jaspersoft.com/JasperSoft_JasperReports.html.
10. **hibernate.org.** HIBERNATE. *HIBERNATE*. [En línea] [Citado el: 5 de Noviembre de 2008.] <http://www.hibernate.org/>.
11. **Exadel.** [En línea] [Citado el: 8 de Noviembre de 2008.] <http://exadel.com/web/portal/products/ExadelStudioPro>.
12. **Sun.** Sun. *Sun*. [En línea] [Citado el: 10 de Noviembre de 2008.] <http://java.sun.com/javaee/javaserverfaces/download.html>.
13. —. Sun. *Sun*. [En línea] [Citado el: 10 de Noviembre de 2008.] <http://java.sun.com/webservices/docs/1.2/tutorial/doc/index.html>.

14. **Visual Paradigm.** Visual Paradigm. [En línea] [Citado el: 14 de Noviembre de 2008.] <http://www.visual-paradigm.com/>.
15. **W3C.** W3C. *W3C*. [En línea] [Citado el: 28 de Enero de 2009.] <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>.
16. **XFire.** XFire. *XFire*. [En línea] [Citado el: 28 de Enero de 2009.] <http://xfire.codehaus.org/Stack+Comparison>.
17. **S. Pressman, Roger.** *Ingeniería del Software. Un enfoque práctico.* s.l. : Felix Varela, 2002.
18. **Gamma, Erich, y otros.** *Design Patterns. Elements of Reusable Object-Oriented Software.*
19. **Buschmann, Meunier, Rohnert, Sommerlad, Stal - Wiley.** *A System of Patterns.*
20. **Mark Grand, Wiley.** *Patterns in Java.*
21. **Floyd Marinescu, Wiley.** *EJB Design Patterns.*