

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 8

Análisis, Diseño e Implementación del módulo
EXPERTICIAS CRIMINALÍSTICAS v2.0
del Sistema de Investigación e Información Policial

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

Autores:

Yaimara Granados Hondares

Yeleyne Maure Díaz

Tutores:

Ing. Yadiel Ramos Rodríguez

Ing. Maikel Pereira Ojeda

Ciudad de La Habana, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

Declaración de autoría

Declaro 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 uso de este trabajo de la forma que se estime conveniente.

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

Yaimara Granados Hondares

Firma del Autor

Yeleyne Maure Díaz

Firma del Autor

Yadiel Ramos Rodríguez

Firma del Tutor

Maikel Pereira Ojeda

Firma del Tutor

Agradecimientos

A la Revolución, a la Universidad de Ciencias Informáticas y en especial a la Facultad 8 por hacer realidad nuestro sueño de formarnos como profesionales.

A nuestros tutores Yadiel y Maiquel por estar siempre dispuestos en los momentos en que necesitamos de su infinita sabiduría.

A nuestro equipo de desarrollo, nuestros amigos, por su consagración y compromiso: al equipo de desarrollo del proyecto CICPC.

Yeleyne Maure Díaz.

A mis padres Niurka y Amaury por su cariño, su dedicación, su apoyo incondicional, sus sacrificios y desvelos.

A mi hermanito Ernesto David, mi adoración, por ser mi mayor inspiración.

A mi familia inmensa por ofrecerme toda una vida llena de amor, por convertir mis debilidades en fortalezas, por ser la única luz al final del camino, mis alas cuando quise volar a conquistar mis sueños, por hacer de una pequeña niña la mujer que los ama a cada instante y más allá de los límites, la mujer que soy hoy.

A mis amigas Lumey, Mayrilis y Marelis, talentosas, divertidas, exitosas, por la dicha de conocerlas y lo importante que ha sido poder contar con ellas en los momentos más difíciles. A Abdany, emprendedor y constante, por ser nuestro D'Artagnan.

A Yaimara en especial, por estos 5 maravillosos años, por hacer más intensas mis alegrías y reducir a la mitad mis penas, por construir juntas esta extraordinaria amistad que sé, conservaremos por siempre.

Yaimara Granados Hondares.

A mis mejores amigos, mis padres Marú y Py, por convertir cada derrota en un triunfo, con confiar siempre en mí, por estar a mi lado cada momento de mi vida, por sus enseñanzas, sus consejos, por comprender y respetar mis decisiones, por su sacrificio y su infinito cariño.

A Jorge, luz de mi vida, por enseñarme que el amor vence a la distancia más grande, por todos nuestros sueños, por la dicha de tenerlo y por hacer de estos cinco años juntos lo mejor que ha podido pasarme.

A mis abuelos Cuca y Hondares por dedicarme toda su vida, por sus cuidados y su dedicación, por formarme junto a tía Julia, tío Dingue, tía Esther y tío Berto, con amor y dulzura, disciplina y respeto durante los primeros años de mi vida.

A tío Wilmar por su cariño incondicional y las largas carreras conmigo en su espalda.

A mi familia y a todos aquellos que se han sumado a esta durante los últimos años, por cada muestra de amor y confianza, por cada palabra de orgullo.

A mis maestros de primaria Mercedes y Margarita, por sus regaños merecidos, por toda su confianza y sobre todo por contribuir de manera significativa en mi formación como estudiante.

A todos mis amigos los viejos y los nuevos.

A mi amiga de hoy Maurín, por convertirse en mi amiga de siempre, por compartir cada susto, cada alegría y tristeza, cada sueño de estos cinco años en la UCI, por una amistad sin igual y la dicha de contar con alguien tan especial.

Dedicatoria

A nuestros padres.

Resumen

El Cuerpo de Investigaciones Científicas Penales y Criminalísticas de Venezuela, constituye una institución única en su tipo dentro del territorio bolivariano y posee como guía detectar, procesar y esclarecer hechos delictivos cometidos a diario. La presente investigación centra su objetivo general en la automatización de los procesos de una de las coordinaciones que integran dicha organización, la Coordinación Nacional de Criminalística (CNC), debido a que no cuenta con un sistema de gestión de información con tecnología de punta que agilice el desarrollo de las investigaciones llevadas a cabo. Esta idea se materializará con el diseño y desarrollo del módulo Experticias Criminalísticas, porción de software que permitirá una mejor gestión de la información, la reducción de los tiempos de respuesta de las peticiones, y abarcará los procesos que se desarrollan diariamente en CNC.

Abstract

The Scientific and Criminal Investigation Corps (CICPC), only institution of its kind within the territory of Venezuela, has the main objectives of identifying, processing and clarifying all criminal offenses committed. This research focuses on the automation of the processes taking place in the National Coordination of Criminalistics, because it has no system for managing information technology to expedite the investigations carried out. This idea was materialize with the design and development of the module Criminalistics Expertise, piece of software it will enable better management of information, reducing the response times of requests, and will cover the processes that take place daily in National Coordination of Criminalistics.

Contenido

Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Introducción	5
1.2 Software de Gestión Policial	5
1.3 El Cuerpo de Investigaciones Científicas Penales y Criminalísticas	7
1.3.1 Procesos de la Coordinación Nacional de Criminalística	9
1.4 El Sistema de Investigación e Información Policial	14
1.4.1 Módulo Experticias Criminalísticas. Requisitos	14
1.5 Metodologías, Lenguajes y Herramientas de Desarrollo	18
1.5.1 Proceso de desarrollo de software	19
1.5.2 Proceso Unificado de Desarrollo (RUP)	19
1.5.3 Proceso Ágil Unificado (AUP)	21
1.5.3 Selección del proceso de desarrollo	23
1.5.4 Lenguaje de Modelado	24
1.5.5 Lenguaje de programación	25
1.5.6 Plataforma de Desarrollo	26
1.5.7 Entorno de Desarrollo Integrado	27
1.5.8 Herramienta CASE	28
1.5.9 Frameworks utilizados en la solución	29
1.5.10 Sistema Gestor de Base de Datos	34
1.6 Arquitectura Técnica	35
1.7 Conclusiones	39
Capítulo 2. Análisis, Diseño e Implementación de la Propuesta de Solución	40
2.1 Introducción	40
2.2 Análisis y Diseño de la Solución Propuesta	40
2.2.1 Diagramas de Paquetes	41
2.2.2 Patrones de Diseño. Tarjetas CRC	42
2.2.3 Realización de los Casos de Uso	59
2.3 Modelo de Datos	65
2.3.1 Diagrama de Clases Persistentes	65
2.3.2 Diagrama de tablas del Modelo Relacional	67
2.4 Modelo de Implementación	68
2.4.1 Diagramas de subsistemas de implementación	68

2.4.2 Diagrama de componentes	69
2.4.3 Estándar de codificación	72
2.4.4 Interfaces de Usuario	72
2.5 Conclusiones.....	79
Capítulo 3 Validación de la Solución Propuesta.....	80
3.1 Introducción.....	80
3.2 Tipos de Prueba. Niveles de Prueba.	81
3.2.1 Niveles de Prueba.....	81
3.3 Conclusiones.....	90
Conclusiones	91
Recomendaciones	92
Bibliografía.....	93
Glosario de Términos.....	95
Anexos	97
Anexo 1: Tarjetas CRC para la capa de negocio del Módulo Experticias Criminalísticas.....	97
Anexo 2: Tarjetas CRC para la capa de negocio del Módulo Experticias Criminalísticas.....	98
Anexo 3: Interfaces de Usuario.	99
Anexo 3.1: IU Incluir Planilla de Retrato Hablado.....	99
Anexo 3.2: IU Incluir Informe de Inspección Técnica	100
Anexo 3.3: IU Incluir Informe Pericial de Experticia de Vehículo	101

Índice de Figuras.

Ilustración 1: Estructura Organizativa del CICPC.....	8
Ilustración 2: Modelo de Casos de Uso Módulo Experticias criminalísticas.....	15
Ilustración 3: Fases y Flujos de Trabajo definidos por AUP.	23
Ilustración 4: Estructura Cliente-Servidor de una aplicación Web	30
Ilustración 5: Flujo de petición de un componente Ajax4JSF.	31
Ilustración 6: Estructura de Spring.	33
Ilustración 7: Arquitectura de Hibernate.	34
Ilustración 8: Sistema Cliente-Servidor.	36
Ilustración 9: Modelo N-Capas.	37
Ilustración 10: Modelo Vista Controlador.....	38
Ilustración 11: Filtro AuthenticationProcessingFilter y sus dependencias.....	39
Ilustración 12: Diagrama de Paquetes.	42
Ilustración 13: Tarjetas CRC para la jerarquía de Diligencia.	45
Ilustración 14: Tarjetas CRC. CU Gestionar Trayectoria Intraorgánica.....	46
Ilustración 15: Tarjetas CRC. CU Gestionar Trayectoria Balística.....	47
Ilustración 16: Tarjetas CRC. CU Gestionar Resultado de Inspección Técnica de Robo a Banco.	48
Ilustración 17: Diagrama de Contrato entre Paquetes CU Intraorgánica: Incluir.....	59
Ilustración 18: Diagrama de Contrato entre Paquetes CU Intraorgánica: Ver.....	59
Ilustración 19: Diagrama de Contrato entre Paquetes CU Intraorgánica: Modificar.....	60
Ilustración 20: Diagrama de Contrato entre Paquetes CU Intraorgánica: Eliminar.....	60
Ilustración 21: Diagrama de Contrato entre Paquetes CU Balística: Incluir.....	61
Ilustración 22: Diagrama de Contrato entre Paquetes CU Balística: Ver.....	61
Ilustración 23: Diagrama de Contrato entre Paquetes CU Balística: Modificar.....	62
Ilustración 24: Diagrama de Contrato entre Paquetes CU Balística: Eliminar.....	62
Ilustración 25: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Incluir.....	63
Ilustración 26: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Ver.....	63
Ilustración 27: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Modificar.....	64
Ilustración 28: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Eliminar.....	64
Ilustración 29: Diagrama de clases persistentes. Vista I.	65
Ilustración 30: Diagrama de clases persistentes. Vista II.	66
Ilustración 31: Modelo Relacional.	67
Ilustración 32: Diagrama de subsistemas de Implementación.....	68
Ilustración 33: Diagrama de Componentes. Vista Acceso a Datos.....	69
Ilustración 34: Diagrama de Componentes. Vista Lógica de Negocio.	70
Ilustración 35: Diagrama de Componentes. Vista Presentación.....	71
Ilustración 36: IU Incluir Informe Pericial de Trayectoria Intraorgánica.....	73
Ilustración 37: IU Incluir Informe Pericial de Trayectoria Intraorgánica. Montaje Fotográfico.....	74

Ilustración 38: IU Incluir Informe Pericial de Trayectoria Intraorgánica. Editar Imágenes.	75
Ilustración 39: IU Incluir Informe Pericial de Trayectoria Balística.	76
Ilustración 40: Incluir Informe Pericial de Trayectoria Balística. Orificios e/o Impactos.	77
Ilustración 41: IU Incluir Informe Pericial de Trayectoria Balística. Montaje Fotográfico.	78
Ilustración 42: NC detectadas durante la Pruebas Internas	83
Ilustración 43: NC detectadas durante las Pruebas Cruzadas	84
Ilustración 44: NC detectadas durante las Pruebas de Liberación	84
Ilustración 45: NC detectadas durante las Pruebas de Liberación	85
Ilustración 46: NC detectadas durante las Pruebas de Aceptación	85
Ilustración 47: NC detectadas durante las Pruebas Piloto.....	86
Ilustración 48: Resumen Gráfico de la Pruebas realizadas	88
Ilustración 49: Experticias Criminalísticas respecto al SIIPOL.....	89
Ilustración 50: Experticias Criminalísticas respecto al SIIPOL.....	89

Introducción

Venezuela ha sido declarada en varias ocasiones el país más violento del mundo con una tasa de crímenes muy por encima a la media de la región. La tensa situación en las calles y la corrupción en el aparato policial contribuyen a la inseguridad ciudadana y al estado de paranoia generalizados.

En consecuencia el país suramericano, como parte del proyecto de Seguridad Ciudadana llevado a cabo por el Ejecutivo Nacional Hugo Rafael Chávez Frías al inicio de su mandato, cuenta con el Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC), institución única en su tipo dentro del territorio bolivariano que posee como objetivos generales detectar, procesar y esclarecer hechos delictivos cometidos a diario (1). Para ello cuenta con el Sistema Integrado de Información Policial como aplicación Informática, cuya ineficiencia, -debido a la tecnología obsoleta que presenta, el alto costo en tiempo de respuesta a las peticiones , así como su interfaz poco amigable para los usuarios que interactúan con ella- dificulta de manera considerable los procesos que se llevan a cabo dentro de la institución.

El CICPC consta dentro de su estructura organizativa con la Coordinación Nacional de Criminalística (CNC), que posee entre sus tareas principales planificar, coordinar y dirigir los procesos técnico-científicos de las investigaciones. Esta Coordinación es la encargada de definir las políticas, normas, procedimientos y planes estratégicos que se implementan para el correcto trabajo investigativo, además de diseñar estrategias y definir métodos de estandarización en la rama criminalística.

Hoy en día los funcionarios que laboran en la CNC, alegan que el sistema vigente no cubre la totalidad de sus necesidades, la consulta y procesamiento de la información, se tornan cada vez más complicados por su volumen creciente. Esto trae consigo la falta de información actualizada, oportuna y fiable a las entidades de la dirección del CICPC. Para que los entes solicitantes de las experticias conozcan información sobre los análisis o sus resultados, tienen que esperar por la materialización de la investigación o el informe, para conocer las conclusiones y poder valorar su importancia. En casos específicos existe falta de seguridad y control en los datos que se registran. Otra problemática importante por la que atraviesa la Coordinación es que existen limitaciones en la diversidad de información que se requiere para la investigación de los hechos y en la calidad de uso de la que se

almacena. Por otra parte, está presente la imposibilidad de acceder y utilizar información de interés de otras organizaciones. Si se desea conocer alguna información relacionada con algún caso, esta tiene que ser canalizada a través del despacho que lleva la averiguación de dicho caso. En algunas áreas que componen esta Coordinación existe falta de seguridad y control en los datos que se manejan, debido a que los registros son llevados en un formato Microsoft Excel, lo que da cabida a que los apuntes puedan ser modificados o alterados; no existe un supervisor o persona encargada del control y cuidado del registro de experticias, lo que origina grandes posibilidades de pérdida.

El CICPC y sus coordinaciones se ven en la necesidad de contar con un sistema de gestión competitivo, con tecnología de punta, que almacene y maneje la información generada.

Con el propósito de desarrollar un nuevo sistema, más completo, que permita una mejor gestión de la información, disminuya los tiempos de respuesta de las peticiones, y cubra los procesos que se desarrollan diariamente en la CNC, se firmó un contrato en el marco de las relaciones Cuba – Venezuela por la colaboración de los países del ALBA, con el fin de modernizar y automatizar el CICPC a través de la creación del Sistema de Investigación e Información Policial (SIIPOL).

A partir de la aplicación de la ingeniería de requerimientos basada en un análisis exhaustivo del funcionamiento de la institución, así como sus procesos de negocio, se generó un conjunto de requisitos funcionales y no funcionales, que resumen las capacidades y condiciones que debe cumplir el nuevo sistema y que dan al traste con el antiguo Sistema Integrado de Información Policial.

Para llevar a cabo la implementación de esta solución informática se ha modelado un sistema dividido en subsistemas, y estos en módulos que encapsulan funcionalidades afines, uno de los cuales es el módulo Experticias Criminalísticas, el cual constituye el objeto de la presente investigación.

Planteada la situación problemática se puede enunciar el **problema científico** a partir de la siguiente interrogante. ¿Cómo garantizar el cumplimiento de los requisitos funcionales y no funcionales asociados al módulo Experticias Criminalísticas del Sistema de Investigación e Información Policial (SIIPOL)?

El **objeto de estudio** determinado para el presente trabajo de diploma se basa en el proceso de desarrollo de software en Sistemas de Gestión de Información Policial y el **campo de acción** regido por la idea anterior se enmarca en el análisis, diseño e implementación del módulo Experticias Criminalísticas, perteneciente al sistema de gestión de información SIIPOL.

La **idea a defender** se plantea de la siguiente forma: “Con el análisis, diseño e implementación del módulo Experticias Criminalísticas se garantizará el cumplimiento de los requisitos funcionales y no funcionales obtenidos en la Ingeniería de Requerimientos”.

El **objetivo general** de este trabajo es analizar, diseñar e implementar el módulo Experticias Criminalísticas perteneciente al subsistema Investigación Criminalística del Sistema de Investigación e Información Policial (SIIPOL).

Los **objetivos específicos** son:

- Valorar los resultados obtenidos en la aplicación de la ingeniería de requerimientos.
- Analizar, definir, estructurar y detallar mecanismos de diseño que garanticen el cumplimiento de las necesidades del módulo Experticias Criminalísticas.
- Implementar el módulo diseñado respetando la arquitectura definida para el nuevo SIIPOL.
- Lograr la integración satisfactoria al subsistema Investigación Criminalística y al SIIPOL del módulo implementado.

Para dar cumplimiento a los objetivos planteados se planificaron varias **tareas investigativas**, tales como:

1. Indagar acerca de otras soluciones informáticas o aplicaciones similares, fundamentalmente en aquellas que desarrollen sistemas de Gestión Policial.
2. Investigar acerca de la metodología, herramientas y lenguaje a utilizar en el análisis, diseño e implementación del módulo Experticias Criminalísticas perteneciente al subsistema Investigación Criminalística del SIIPOL.
3. Estudiar la propuesta de Arquitectura para el SIIPOL y el modelo de Casos de Uso que da cumplimiento a los requisitos funcionales y no funcionales

asociados al módulo Experticias Criminalísticas perteneciente al subsistema Investigación Criminalística del SIIPOL.

4. Investigar la aplicación de Patrones de Diseño en soluciones similares.
5. Realizar diagramas de clases de diseño y los diagramas de interacción para los Casos de Uso de mayor complejidad.
6. Aplicar Patrones de Diseño estudiados que contribuyan a optimizar el modelo de diseño.
7. Implementar los componentes necesarios de forma tal que se dé cumplimiento a los objetivos propuestos.
8. Probar los componentes desarrollados, así como integrar la solución al Sistema de Investigación e Información Policial a medida que se obtienen resultados satisfactorios.
9. Validar la solución desarrollada.

Con el desarrollo exitoso de las tareas investigativas anteriormente expuestas se espera lograr una porción de software funcional que satisfaga los estándares de calidad requeridos y que cumpla con la totalidad de requisitos funcionales y no funcionales arrojados a partir de la aplicación de la ingeniería de requerimientos a los procesos de la Coordinación Nacional de Criminalística.

El contenido a desarrollar en el presente trabajo está estructurado en tres capítulos: **Fundamentación Teórica**, incluye un estado del arte del tema tratado; **Análisis, diseño e implementación de la propuesta de solución**, constituye el grueso de la investigación, muestra resultados tangibles en un software funcional; y **Validación de la solución propuesta**, el cual resumirá el conjunto de pruebas realizadas al software que patentizarán su correcto funcionamiento.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se definirán un conjunto de ideas que enmarcarán al lector, cualquiera que este sea, en el contexto en el cual se desarrolla esta investigación. Asimismo se espera lograr un basamento teórico sólido a partir de conceptos y definiciones que protagonizarán esta parte del documento. La exposición de tendencias actuales utilizadas en el mundo para dar respuesta y solución a problemáticas y necesidades similares a las dadas a conocer en este trabajo, será de gran utilidad en su futuro desarrollo práctico.

Para lograr este último con la calidad requerida se hace necesario el estudio previo, al que va dedicado este primer capítulo.

1.2 Software de Gestión Policial

En el mundo actual los índices de criminalidad en numerosas partes del mundo resultan alarmantes y cobran a diario miles de vidas en cada rincón del planeta. Ningún país está exento y todos los gobernantes se plantean estrategias a lo largo de su mandato, para disminuir en gran medida estos hechos delictivos.

Hoy día la gestión policial encuentra en la informática una alternativa al verdadero desafío que representa enfrentar las nuevas tendencias de la criminalidad como el crimen organizado, el narcotráfico, el lavado de dinero, el delito medio-ambiental y el cibercrimen, si bien la informatización no es siempre sinónimo de modernización, es indudable que ella constituye una herramienta necesaria y fundamental. En tal caso los Sistemas de Gestión Policial han venido a revolucionar la manera de procesar la información. Entre los más comunes se encuentran los sistemas AFIS (Automated Fingerprint Identification System). (1) Los Servicios policiales fueron los primeros en utilizar estos sistemas de identificación de huellas dactilares, con el objetivo de analizar y comparar rastros latentes. Las principales policías del mundo están equipadas de sistemas AFIS para la investigación criminal. Asimismo IBIS (Integrated Ballistics Identification System) es frecuentemente utilizado en los organismos policiales ya que reducen los tiempos y aumentan las certezas en la identificación de evidencias o personas involucradas en delitos. (2)

El Sistema Territorial de Emergencias y Gestión Policial (STEGPOL), identifica en forma veraz y efectiva dónde geográficamente se están cometiendo o se han

cometido actos delictuales a nivel territorial, apoyado con sistemas de información en línea desde el lugar de los hechos, que permitan la acción rápida y coordinada entre los diferentes actores encargados de la seguridad ciudadana a nivel país. (3)

El Sistema de Gestión Penitenciaria SIGEP constituye un proyecto que está actualmente en etapa de desarrollo y que dará respuesta a las necesidades de gestión, información y apoyo a la toma de decisiones de la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR). Este sistema tiene como objetivos fundamentales aumentar la eficacia, profesionalismo y equidad en el sistema penitenciario venezolano, de manera que se logre un incremento de la confianza en el sistema penitenciario en general; concebir y diseminar información vital para el funcionamiento de los establecimientos penitenciarios. (4)

Helios es una aplicación informática integral que permite administrar, controlar y registrar diferentes actividades relacionadas directamente con estaciones policiales, dígame gestión de agentes, denuncias de tráficos, control de vehículos y detenidos, búsquedas y análisis estadísticos. Posee una interfaz perfectamente configurable.

Con respecto a los creadores de algunos de Sistemas de Gestión Policial, se destaca el grupo TECNI-SOFT, empresa que lleva más de 15 años desarrollando aplicaciones para la Policía Local en varias regiones de España, entre las aplicaciones desarrolladas se encuentra POL dedicado a la Gestión Integral de Policía Local , sistema que cuenta con más de 100 instalaciones en toda España. El sistema se ha desarrollado por informáticos que son miembros de la Policía Local, está enlazado con PDAS, sistema de SMS, E-mail, Cajeros de Cobros de multas, entre otros; CPOL, cuya finalidad es la Gestión de Croquis para Atestados por Accidentes de Tráfico, permite realizar croquis de accidentes de forma sencilla además podrá utilizar como base para importar cruces y zonas del callejero del plano de su ciudad, si lo tiene en formato de AUTOCAD, con extensión DXF o DWG; POLMobile centrado en la Gestión de recogida de infracciones en materia de tráfico y Anomalías y desperfectos en la vía pública con PDAS integrado en POL; MULTAPOL es el programa ideal para la Gestión de Multas de Tráfico, contiene entre otras estas características la gestión y configuración del fichero de agentes denunciadores, libro de registro de entradas y salidas de documentos, ordenanza de tráfico en la ciudad, así como sistema SMS y fax integrado.

Numerosas son las naciones que se han beneficiado con la instalación de alguno de estos softwares de gestión policial, tal es el caso de Colombia en su incesante lucha contra el narcotráfico, Chile, Venezuela, España, Estados Unidos, entre otros, señalando así el aporte significativo que brindan estos sistemas en procesamiento ágil de la información y la disminución de los índices de criminalidad.

1.3 El Cuerpo de Investigaciones Científicas Penales y Criminalísticas

El Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC) es una institución que garantiza la eficiencia en la investigación del delito en la República Bolivariana de Venezuela mediante su determinación científica, asegurando que el ejercicio de la acción penal conduzca a una sana administración de la justicia. Su misión se concentra, dada su reconocida capacidad científica y máxima excelencia de los recursos que posee, en alcanzar el más alto nivel de credibilidad nacional e internacional en la investigación del fenómeno delictivo organizado y criminalidad violenta.

El CICPC ostenta entre sus objetivos fundamentales el de optimizar las acciones de investigación criminal con el fin de lograr el esclarecimiento de los hechos delictivos; capacitar el personal integrado en la institución, garantizando de esta forma la eficacia y la eficiencia; elevar el sentido de pertenencia a partir de la práctica de valores indispensables para el CICPC; garantizar las acciones y medios tendentes a mejorar la calidad de vida de sus miembros, en el aspecto educativo, cultural, deportivo, social y económico; consolidar la imagen de profesionalismo de la institución ante la comunidad en general, fundamentada en una gerencia de alta capacidad de respuesta; lograr insertarse en la comunidad internacional como organismo de investigación penal de vanguardia; dotar al capital humano del CICPC de herramientas, mecanismos logísticos y de infraestructura que garanticen el óptimo desempeño de sus funciones; apoyar las políticas de Estado a través de estrategias dirigidas a la reducción de los delitos en todas sus modalidades; fortalecer organizacionalmente la institución y su sinergia con otros organismos de la Administración Pública Nacional y con instituciones privadas. (5)

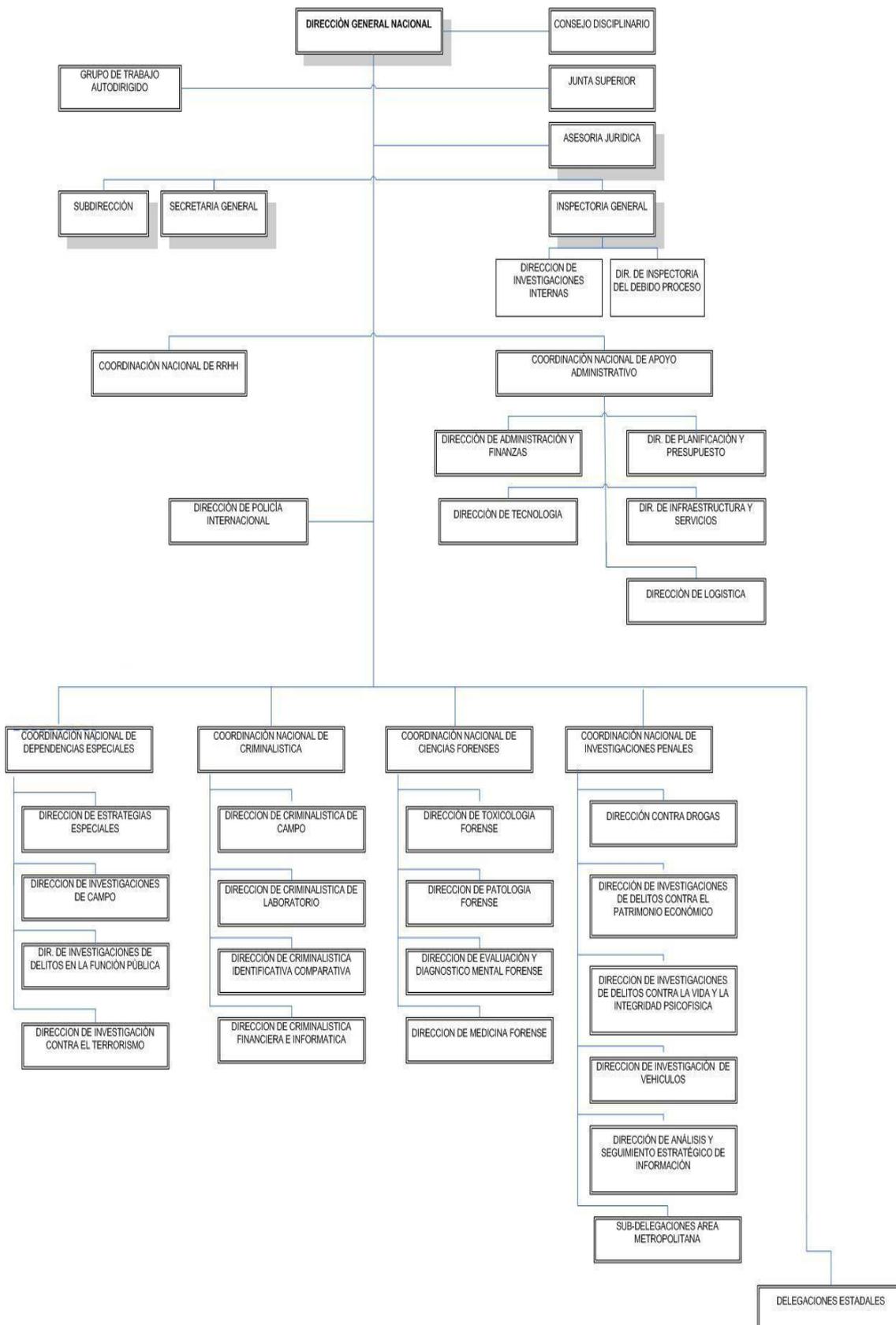


Ilustración 1: Estructura Organizativa del CICPC

El CICPC cuenta con el Sistema Integrado de Información Policial como aplicación Informática, el cual se considera en la actualidad ineficiente, ya que no cubre todos los procesos que se realizan a diario en la institución, muchos de ellos no se encuentran automatizados. La tecnología obsoleta que presenta, el alto costo en tiempo de respuesta a las peticiones, así como su interfaz poco amigable para los usuarios que interactúan con ella, constituyen aspectos que obstaculizan en lugar de agilizar las diversas operaciones que se efectúan dentro del Cuerpo de Investigaciones Científicas Penales y Criminalísticas. A partir de la información antes mencionada se patentiza la necesidad de contar con sistema con tecnología de punta y que brinde solución instantánea a los diferentes problemas por los que atraviesa actualmente la institución.

1.3.1 Procesos de la Coordinación Nacional de Criminalística

El Cuerpo de Investigaciones Científicas Penales y Criminalísticas cuenta, dentro de su estructura organizativa, con la Coordinación Nacional de Criminalística, la cual es la encargada de planificar, coordinar y dirigir los procesos técnico-científicos de las investigaciones. Entre las principales tareas que se desarrollan en esta área del CICPC se encuentra la de diseñar las estrategias que garanticen la cadena de custodia de evidencias físicas, definir métodos de estandarización criminalística, definir las políticas, normas, procedimientos y planes estratégicos; así como promover la actualización técnico-científica en materia de criminalística. A partir de un estudio realizado de esta coordinación, se identificaron procesos de negocio que la distinguen.

Proceso: Realizar Inspección Técnica Especializada

Mediante este proceso los entes investigativos del CICPC y otras entidades de la misma índole obtienen información completa de todo lo referente a un lugar donde se detectó un hecho delictivo.

Proceso: Reconstrucción de Hechos

En este proceso se obtiene como resultado un Informe Pericial sobre la reconstrucción en el sitio del suceso de un hecho punible relacionado con trayectorias de balas de las arma de fuego relacionadas en el hecho.

Proceso: Levantamiento Planimétrico

Este proceso se efectúa con el objetivo de elaborar el plano de un sitio determinado a partir de las observaciones obtenidas en el lugar.

Proceso: Retrato Hablado

En este proceso se elabora el retrato hablado en función de la información que suministra el agraviado o testigo y se emite el oficio o memorando en conjunto con el Retrato Hablado.

Proceso: Trayectoria Balística

En este proceso se obtiene como resultado un Informe Pericial sobre lo ocurrido en el sitio del suceso de un hecho punible relacionado con trayectorias balísticas de un arma de fuego.

Proceso: Trayectoria Intraorgánica

Este proceso se efectúa con el objetivo de obtener un informe pericial con el estudio efectuado a un protocolo de autopsia de un cadáver.

Proceso: Siniestros

Distribuir, coordinar y vigilar las solicitudes de las evidencias colectadas en los siniestros ocurridos, los expertos realizan el análisis criminalístico pertinente y verifican las causas que dieron origen al siniestro a través de la elaboración del informe pericial.

Proceso: Revelado o análisis de fotografías

Proceso mediante el cual se garantiza la obtención de fotografías a las entidades solicitantes y el análisis de las mismas cuando sea necesario.

Proceso: Elaborar reseñas

Realizar reseñas fotográficas y dactilares a todas aquellas personas implicadas en algún tipo de delito, las cuales son trasladadas a este despacho por funcionarios adscritos a las diferentes divisiones operativas del CICPC.

Proceso: Experticias biológicas

Proceso que permite identificar una evidencia basándose en su estudio biológico, este proceso comienza cuando llega una solicitud que puede estar

acompañada de la evidencia o puede requerir que los expertos acudan a un área determinada. Una vez concluida la experticia se remite el informe pericial a la entidad solicitante.

Proceso: Experticias Física comparativa o audio visual

Proceso a través del cual se obtiene un informe pericial como resultado del análisis criminalístico desde el punto de vista físico Comparativo o audio visual.

Proceso: Análisis Físico Químico

En este proceso se efectúan experticias para emitir Informe Pericial, acerca de los resultados del análisis de muestras para identificar sustancias.

Proceso: Análisis de Microscopia Electrónica

Elaborar experticia para emitir Informe Pericial acerca de los resultados del análisis de muestras para identificar sustancias que estén presentes en una persona o evidencia determinada.

Proceso: Experticias de Avalúos

En este proceso se Obtiene un Informe Pericial de los avalúos reales de bienes muebles e inmuebles o regulaciones prudenciales.

Proceso: Experticias Contables

Este proceso se efectúa con el objetivo de obtener un Informe Pericial acerca del análisis de la documentación de una institución o persona que esté sujeta de una investigación determinada.

Proceso: Experticias Informáticas

Este proceso se inicia una vez recibida la solicitud de experticias informáticas, esta solicitud es asignada a un experto en experticias informáticas, que realiza un informe pericial el cual será remitido a la entidad solicitante.

Proceso: Análisis Balístico

Este proceso se inicia una vez recibida una solicitud con la evidencia correspondiente, los expertos elaboran un dictamen pericial que luego es entregado por un mensajero a correspondencia terminando así el proceso.

Proceso: Análisis de documentología

Este proceso se inicia una vez recibida una solicitud para realizar análisis de documentología a la evidencia entregada. Una vez creado el informe pericial se le entrega al despacho solicitante.

Proceso: Experticias de Rastros

El objetivo es procesar todos los rastros de una persona, que se recibe a fin de verificar si la misma posee algún registro que la logre identificar.

Proceso: Experticias Dactilares

El objetivo es procesar todas las evidencias dactilares que reciben a fin de verificar si las mismas poseen algún registro que las logre identificar.

Proceso: Activaciones Especiales

El objetivo de este proceso es efectuar las activaciones especiales en un sitio de un suceso determinado, en un vehículo que se encuentre en un estacionamiento o a una evidencia determinada, a fin de encontrar rastros que puedan identificar a personas involucradas con la evidencia o elemento donde se realice la activación.

Proceso: Servicios en la Oficina de Flagrancia

El objetivo de este proceso es tomar las reseñas de un detenido por delito flagrante y verificar si el mismo posee algún registro policial.

Proceso: Peritación Civil

El objetivo de este proceso es lograr esclarecer la legitimidad de una madre mediante el análisis de las reseñas dactilares de la misma.

Proceso: Gestión del AFIS

El objetivo de este proceso es dotar al Sistema AFIS con las nuevas evidencias dactilares, así como eliminarlas en caso de que sea necesario.

Proceso: Experticias de Vehículos

En este proceso se obtiene un Informe Pericial a cerca de las experticias realizadas a un vehículo para determinar si sus seriales han sido o no alterados.

Proceso: Notificación de Extravío de Matrícula

En este proceso se lleva a cabo la denuncia de la pérdida o extravío de la matrícula de un vehículo por parte de un agraviado.

Proceso: Resguardo y custodia de la evidencia

Este proceso se inicia cuando llega al despacho un funcionario solicitando el resguardo y custodia de determinada evidencia o solicita la retirada de una evidencia a través de una autorización elaborada por un Fiscal del Ministerio Público o juez con ese privilegio.

La Coordinación Nacional de Criminalística posee una estructura organizativa que permite efectuar las experticias de forma organizada, esta se divide en:

Oficina de Asesoría y Desarrollo Criminalístico: Su misión es impulsar, asesorar, promover y desarrollar conocimientos técnico-científicos, así como contribuir al fortalecimiento científico y tecnológico, garantizando el acceso directo y la difusión del conocimiento en el área de criminalística.

División de Resguardo y Custodia de Evidencias Físicas: Esta división se encarga de recibir, clasificar y controlar las evidencias físicas provenientes de la perpetración de un hecho punible, así como garantizar su resguardo y disponibilidad. Además vela por la creación de las estrategias que garanticen la cadena de custodia de dichas evidencias.

Dirección de Criminalística de Campo: Se encarga de coordinar, implantar y supervisar las actuaciones periciales de campo.

Dirección de Criminalística de Laboratorio: Coordina y supervisa el análisis de laboratorio, contribuyendo a la validación de pruebas.

Dirección de Criminalística de Identificativa Comparativa: Se especializa en llevar a cabo análisis físicos, identificativos–comparativos, balísticos, documentológicos, lofoscópicos y físicos – comparativos.

Dirección de Criminalística Financiera Informática: Es el área de trabajo dentro de la Coordinación que se encarga de controlar, supervisar, coordinar e implantar avalúos, sistemas de computación, investigaciones financieras y contables.

1.4 El Sistema de Investigación e Información Policial

El Sistema de Investigación e Información Policial (SIIPOL) como sistema de gestión competitivo, con tecnología de punta, que almacene y maneje la información generada, suplantará al viejo SIIPOL, ofreciendo numerosas ventajas.

Como propuesta general del nuevo SIIPOL se define una aplicación cliente-servidor, publicada en un servidor web, que contará además con un servidor de Base de Datos y uno de aplicaciones.

La aplicación estará dividida en subsistemas y estos en módulos para mayor organización y facilidad a la hora de desarrollar.

1.4.1 Módulo Experticias Criminalísticas. Requisitos

Experticias Criminalísticas constituye un módulo, dentro del subsistema Investigación Criminalística, que brinda funcionalidades que permiten registrar los resultados correspondientes a las solicitudes de experticias criminalísticas y acceder a las acciones que sean necesarias para complementar dichos resultados. Este módulo cubre el 63% del subsistema, el cual se encarga de la manipulación automatizada de las respuestas de los especialistas, en determinadas áreas de las ciencias criminalísticas, a las solicitudes que son recibidas a través de una cadena de comunicación provista por el software o físicamente.

1.4.1.1 Modelo de Casos de Uso: Módulo Experticias Criminalísticas.

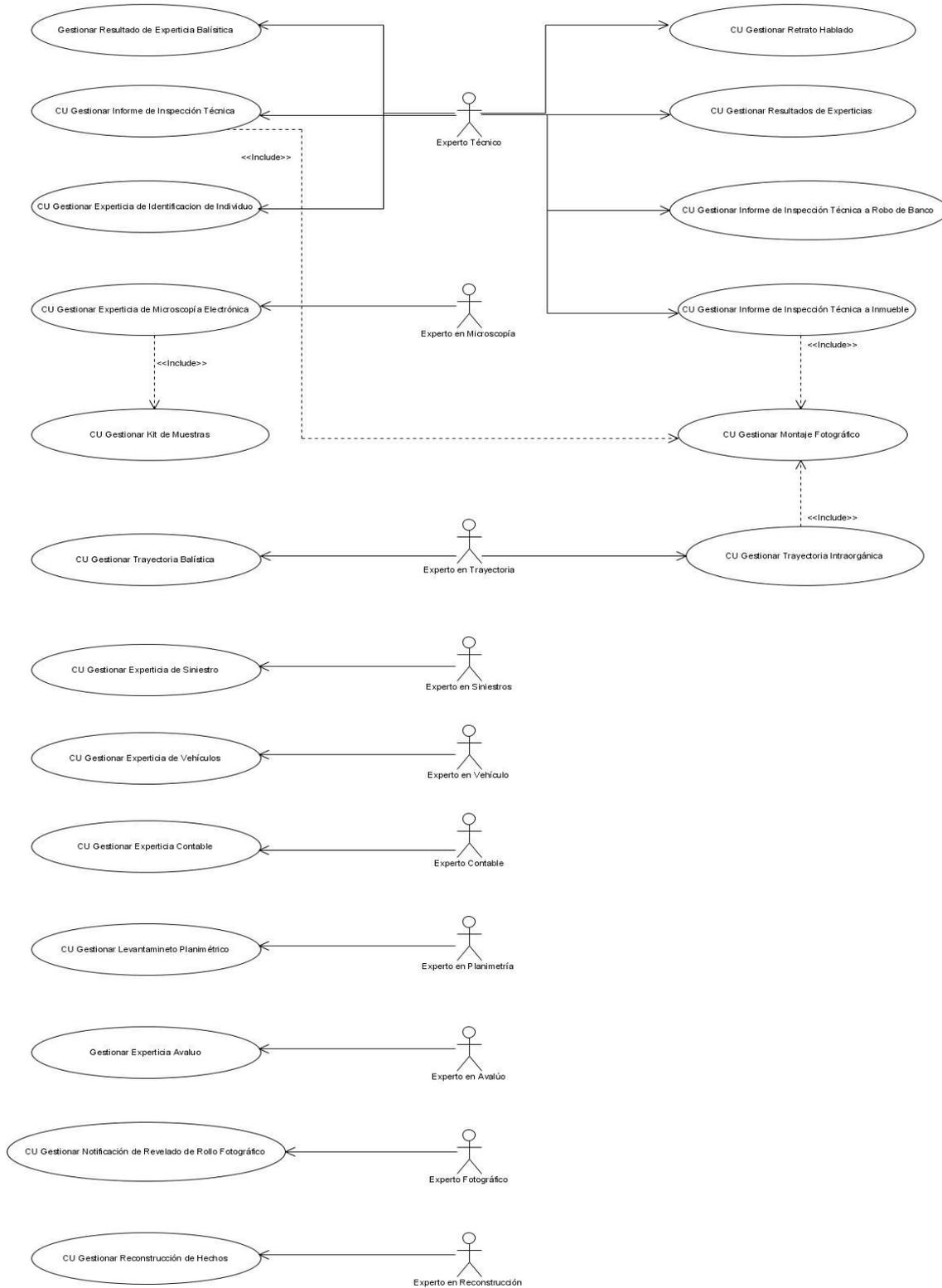


Ilustración 2: Modelo de Casos de Uso Módulo Experticias criminalísticas

1.4.1.2 Descripción Casos de Uso.

Entre los Casos de Uso pertenecientes al Módulo Experticias Criminalísticas, se han seleccionado los siguientes para ilustrar el desarrollo de la presente investigación:

- Gestionar Trayectoria Intraorgánica

Este Caso de Uso es iniciado por el Experto en Trayectoria y posee como objetivo esencial el de permitir la realización de cualquier acción sobre un Informe de Trayectoria Intraorgánica, dígase incluirlo, modificar su información, eliminarlo o simplemente mostrar sus datos.

Para que este Caso de Uso se lleve a cabo es necesario que se haya generado el escritorio de trabajo del usuario autenticado. El informe puede incluirse como respuesta a una Solicitud de Experticia de tipo Trayectoria Intraorgánica seleccionada previamente o incluirse en el sistema de forma temporal bajo el estado *En Curso*, hasta que el experto encargado de gestionar la investigación lo asocie definitivamente a la solicitud que le dio origen dando por concluida la experticia.

- Gestionar Trayectoria Balística

Este Caso de Uso es iniciado por el Experto en Trayectoria y posee como objetivo esencial el de permitir la realización de cualquier acción sobre un Informe de Trayectoria Balística, dígase incluirlo, modificar su información, eliminarlo o simplemente mostrar sus datos.

Para que este Caso de Uso se lleve a cabo es necesario que se haya generado el escritorio de trabajo del usuario autenticado. El informe puede incluirse como respuesta a una Solicitud de Experticia de tipo Trayectoria Balística seleccionada previamente o incluirse en el sistema de forma temporal bajo el estado *En Curso*, hasta que el experto encargado de gestionar la investigación lo asocie definitivamente a la solicitud que le dio origen dando por concluida la experticia.

- Gestionar Resultado de Inspección Técnica de Robo a Banco

Este Caso de Uso es iniciado por el Experto Técnico y posee como objetivo esencial el de permitir la realización de cualquier acción sobre un Informe de

Inspección Técnica de Robo a Banco, dígame incluirlo, modificar su información, eliminarlo o simplemente mostrar sus datos.

Para que este Caso de Uso se lleve a cabo es necesario que se haya generado el escritorio de trabajo del usuario autenticado. Es importante además, en el momento de incluir un informe, que una Solicitud de Inspección Técnica a Entidades Financieras haya sido seleccionada previamente.

1.4.4.3 Requisitos Suplementarios Fundamentales

A partir de un análisis realizado a los Casos de Uso pertenecientes al módulo Experticias Criminalísticas, se derivó un conjunto de requisitos suplementarios de vital importancia para la correcta construcción del software. A continuación se darán a conocer algunos de estos.

RNF 1 Formularios.

- Los formularios serán accesibles y estarán estandarizados.
- Los campos de texto deben tener un tamaño estándar de acuerdo al espacio con que se cuente en el área de la página y en la medida que se llene esa área primaria agregar la barra de desplazamiento vertical.
- No se recomienda usar barras de desplazamiento horizontal excepto en casos muy justificados (Ejemplo: visualización de imágenes).
- Las opciones textuales que encabezan los Botones de opción han de ser redactadas de tal manera que forzosamente alguna de ellas satisfaga al usuario. Han de ser claramente excluyentes entre ellas.
- Mostrar al inicio de la etiqueta (en caso de tener muchos términos) la palabra más significativa.
- En el sistema se deberán usar las casillas de verificación siempre al lado izquierdo del texto asociado.
- La interfaz señalará en color rojo los campos obligatorios.
- El sistema presentará los términos capitalizados (primera letra en mayúsculas) y situar dos puntos al final de cada campo.
- El sistema será consistente en el uso de abreviaturas, usar la misma abreviación siempre para la misma palabra.
- En caso de revisar los campos antes de almacenarlos el sistema mostrará de manera distinguible los campos con errores.

RNF 2 Ubicación, guías y ayudas al usuario.

- Se deben acceder desde todas las vistas a la ayuda general.
- Se debe mostrar las rutas de acceso según la navegación que tenga el usuario.
- El sistema debe mostrar el nombre del usuario que se autentique.
- El sistema debe tener acceso al menú general desde cualquiera de sus páginas.
- Se deberá definir un color único para los enlaces visitados en una misma sección de trabajo.
- Subrayar los enlaces textuales y dado el caso colocarlos en azul.
- Se deberán usar etiquetas alternativas para todas las imágenes descriptivas (fotos, documentos escaneados, etc.)
- Ubicar en cada página el Pie de Página.

RNF 3 Acceso

- El sistema mostrará un texto de bienvenida al acceder a su página de entrada.
- El sistema mostrará de una forma distintiva mensajes para cada eventualidad que ocurra con la entrada al sistema.
- El sistema mostrará textos personalizados para los mensajes de error.
- El sistema informará quien es el usuario correspondiente con la sección que este activa.

RNF 4 Navegadores

- El sistema funcionará de manera óptima en los navegadores Web Mozilla, y Firefox.
- El sistema funcionará con versiones viejas de cada uno de los Navegadores Web antes mencionados.
- El sistema funcionará correctamente en todos los tamaños de ventanas de los navegadores Web.

1.5 Metodologías, Lenguajes y Herramientas de Desarrollo

El entorno de desarrollo, - metodología, lenguaje de programación y modelado, herramienta CASE, etc.-, definido para la construcción del software, fue producto de un estudio realizado por el equipo de Arquitectura, y establecido como políticas del

proyecto, por lo que la selección del mismo queda fuera del alcance del presente trabajo. Solo se brindará una breve descripción de cada herramienta a utilizar.

Resulta muy importante destacar el caso específico de la metodología de desarrollo, ya que el Proceso Unificado de Desarrollo (RUP) fue seleccionado de forma general, pero en la construcción del módulo Experticias Criminalísticas, por sus características y las del equipo de desarrollo, se pretende utilizar Proceso Ágil Unificado (AUP).

1.5.1 Proceso de desarrollo de software

El proceso de desarrollo de software es aquel en que las necesidades del usuario son traducidas en requisitos de software, los mismos son transformados en diseño y el diseño implementado en código; el código es probado, documentado y certificado para su uso operativo. Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". (6)

Con el objetivo de producir eficiente y eficazmente un software que cumpla los requisitos del cliente, un proceso de desarrollo de software tiene como objetivo primordial, elevar la calidad del software en todas las fases por las que este atraviesa, a través de una mayor transparencia y control sobre el proceso.

1.5.2 Proceso Unificado de Desarrollo (RUP)

El Proceso Unificado de Desarrollo (Rational Unified Process RUP), por sus siglas en inglés, se ha convertido en un estándar de facto en no pocas ocasiones para líderes y gerentes en la medida en que estos se interesan por mejorar los procesos de desarrollo de software que tienen lugar en sus organizaciones. Los autores del Proceso Unificado de Desarrollo destacan que el proceso de software propuesto por RUP tiene tres características esenciales: (7)

- **Dirigido por Casos de Uso:** los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. En RUP los Casos de Uso constituyen la forma de especificar los requisitos del sistema, guiar su diseño, implementación y prueba. Constituyen además un elemento integrador y una guía del trabajo, pues inician el proceso de desarrollo y proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

- **Centrado en la arquitectura:** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, el cual recibe este nombre porque lo forman: la Vista Lógica, Vista de Implementación, Vista de Proceso y Vista de Despliegue, más la Vista de Casos de Uso que ofrece cohesión a todas.

- **Iterativo e incremental:** el trabajo se divide en partes más pequeñas o mini proyectos, permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto y así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Desafortunadamente, la puesta en práctica de RUP es un arduo y arriesgado proceso para los equipos de desarrollo. RUP propone una forma de trabajo que favorece el incremento excesivo de artefactos de documentación y aunque puede ser ajustado, no incluye formas para ajustarlo debidamente a procesos ágiles, lo que usualmente conlleva a reveses en proyectos ante las presiones constantes que son a su vez resultado de una planificación inadecuada, condicionada por los plazos de entrega de los proyectos. Ciertamente RUP, no tiene concebidas prácticas de desarrollo ágil a la hora del diseño y la codificación, por tanto no debe ser culpado por los contratiempos de los equipos, sin embargo ha favorecido en los últimos años a incrementar la tendencia de tomar cada vez con más seriedad la necesidad de considerar el desarrollo ágil como alternativa de éxito en la producción de un software en tiempo y dentro del presupuesto, un software al que se le pueda dar mantenimiento y que pueda ser reutilizado, en pocas palabras: un software completamente funcional.

1.5.3 Proceso Ágil Unificado (AUP)

El Proceso Ágil Unificado (Agil Unified Process AUP), es una versión simplificada de RUP que describe de forma simple y fácil de comprender el uso de técnicas ágiles para el desarrollo de aplicaciones que permanezcan dentro de los conceptos de RUP. AUP promueve principios ágiles de los cuales acentuamos particularmente los siguientes:

- **Integración continua:** mantener una sola revisión para todo el equipo de programación. Se recomiendan lapsos de integraciones de pocas horas a no más de un día.
- **Refactorización:** mantener una depuración y simplificación constante del sistema. Una vez que se ha añadido alguna funcionalidad es necesario ser críticos para encontrar puntos de simplificación de código.
- **Propiedad colectiva del código:** el código puede ser modificado por cualquier miembro del equipo de desarrollo. Se asume que los programadores se rigen por los estándares de codificación definidos, por tanto luego de un tiempo razonable, cualquier programador tiene la habilidad de conocer y dominar todo el código.
- **Bienestar del programador:** se deben lograr condiciones de trabajo óptimas para programar “a máxima velocidad”. Las semanas laborales deben constar de 40 horas divididos en turnos de 8 horas máximo. Las horas de trabajo extras o fines de semana trabajando solo contribuyen al desgaste del equipo de programación, afectan el rendimiento y generan un ambiente propicio para cometer errores, ser displicentes en el apego a las normas y producir software de mediocre calidad. Ha de desarrollarse un ritmo sustentable para el equipo de trabajo.

AUP mantiene las cuatro fases propuestas por RUP (inicio, elaboración, construcción y transición), no así para los flujos de trabajo donde el Modelo, primer flujo establecido por AUP, abarca las disciplinas Modelo de Negocio, Requerimientos y Análisis y Diseño. (8)

Fases:

- **Inicio:** Se determina el alcance inicial del proyecto, se identifica una arquitectura candidata para el sistema y se obtiene la financiación inicial del proyecto así como la aceptación de los interesados.
- **Elaboración:** Se prueba la arquitectura del sistema.

- **Construcción:** Se construye un software de manera regular y gradual partiendo por aquellas funcionalidades que poseen la prioridad más alta en términos de necesidad del cliente.

- **Transición:** Se valida y despliega el sistema en el entorno de producción.

Flujos de Trabajo:

- **Modelo:** Este flujo de trabajo centra su atención en comprender los procesos de negocio que tienen lugar dentro de la organización, el problema inicial que dio origen al proyecto así como identificar una solución viable para el entorno de la empresa.

- **Implementación:** El objetivo de este flujo de trabajo es transformar los modelos en código ejecutable y desarrollar un nivel básico de pruebas, particularmente pruebas de unidad.

- **Prueba:** Se persigue desarrollar un proceso de evaluación de software que garantice la calidad del producto a través de la detección de errores, comprobar que el sistema funcione de acuerdo a como fue diseñado y que se hayan cumplido los requisitos funcionales y no funcionales del mismo.

- **Despliegue:** Esta disciplina establece un plan para la entrega del sistema de manera que los usuarios finales reciban un software completamente funcional.

- **Administración de Configuración:** Se administra el acceso a los artefactos producidos como resultado del proceso de desarrollo del software, así como la actualización y el control de versiones.

- **Administración de Proyecto:** La finalidad de este flujo consiste en dirigir las actividades que tienen lugar dentro del proyecto, esto incluye administración de riesgos y dirección del personal, así como la coordinación entre personas y sistemas fuera del marco del proyecto con el objetivo de garantizar la entrega del sistema en tiempo y sin exceder el presupuesto.

- **Ambiente:** Dispone actividades que describen los procesos y herramientas (hardware, software, etc.) que soportarán el equipo de trabajo del proyecto.

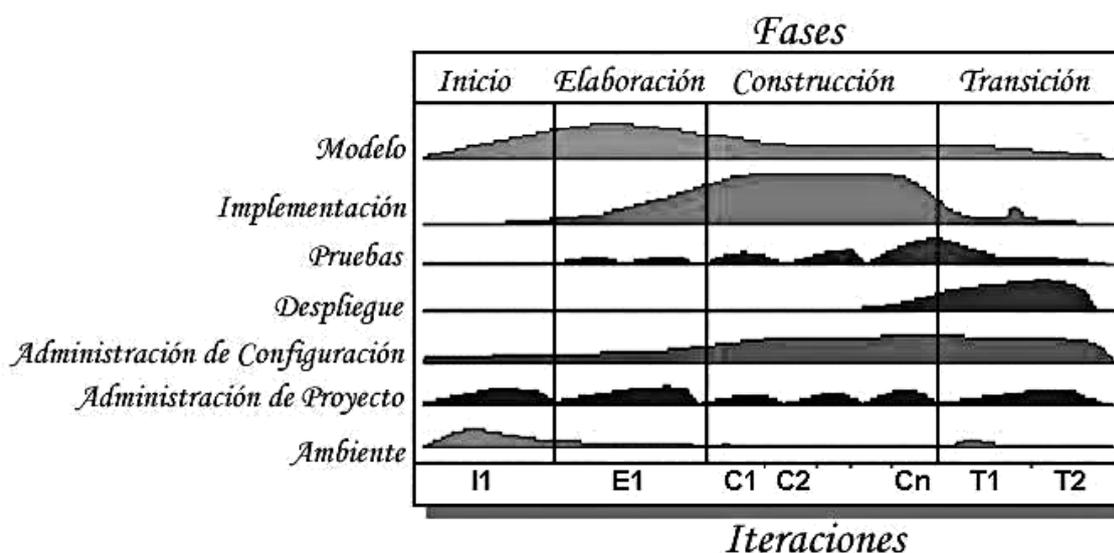


Ilustración 3: Fases y Flujos de Trabajo definidos por AUP.

1.5.3 Selección del proceso de desarrollo

Durante el proceso de desarrollo del módulo Experticias Criminalísticas en su versión 2.0 del SIIPOL, teniendo en cuenta las experiencias acumuladas en la iteración anterior, el equipo de trabajo optó por la puesta en práctica de AUP, que como se abordó anteriormente, incorpora elementos ágiles para el desarrollo teniendo en cuenta los conceptos prefijados por RUP.

Se hace necesario destacar un elemento esencial que distingue al Proceso Ágil Unificado del Proceso Unificado de Desarrollo. De acuerdo con las metodologías tradicionales, el cliente no está obligado a mantener una interacción constante con el equipo de desarrollo, puede seguir el avance del proyecto a través de reuniones planificadas y con un largo período de ocurrencia, siendo necesario por ello una mayor documentación para que exista un control de todo el proceso. AUP establece que debe determinarse una persona que se encargue de velar por el cumplimiento de los requisitos y las prioridades del sistema, esta persona puede ser o bien el cliente en cuestión o un equipo de analistas. En el caso particular, descrito en el documento, este importante rol fue desempeñado por el equipo de analistas, quienes fueron los encargados de hacer cumplir por parte del equipo de desarrollo, las exigencias del cliente de manera eficiente.

Los resultados obtenidos estimulan la puesta en práctica de metodologías ágiles que ofrecen entre sus más connotados beneficios, la capacidad de respuesta a cambios de requisitos a lo largo del desarrollo; entrega continua y en plazos breves de un software funcional; marcada importancia a la simplicidad, eliminando el trabajo innecesario; atención continua a la excelencia técnica y al buen diseño contribuyendo de esta forma a mejoras frecuentes de los procesos y del equipo de desarrollo de manera general.

1.5.4 Lenguaje de Modelado

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. (9)

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que es un lenguaje, cuenta con reglas para combinar tales elementos. Es importante destacar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

En UML 2.0, se encuentran diferentes tipos de diagramas, que debidamente categorizados jerárquicamente, facilitan en gran medida su comprensión.

Dicha jerarquía, podría ilustrarse de la siguiente manera:

Los Diagramas de Estructura estática: Describen las propiedades estructurales del sistema.

- Diagrama de clases: Conjunto de clases, interfaces y colaboraciones; así como sus colaboraciones.
- Diagrama de objetos: Conjunto de objetos y sus relaciones.
- Diagrama de casos de uso: Conjunto de casos de uso y actores y sus relaciones.

Los Diagramas de Comportamiento: Acentúan en lo que debe suceder en el sistema modelado.

- Diagramas de interacción (secuencia y colaboración): Objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.
- Diagrama de estados: Muestra una máquina de estado que consta de estados, transiciones, eventos y actividades.
- Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Los Diagramas de Implementación:

- Diagrama de componentes: Organización y las dependencias entre un conjunto de componentes.
- Diagrama de despliegue: Configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

UML, es considerado un potente lenguaje de modelado que garantiza la especificación de los procesos en el desarrollo del software así como una excelente vía de comunicación y documentación.

1.5.5 Lenguaje de programación

Un lenguaje de programación es un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

1.5.5.1 Java

El lenguaje de programación Java, fue diseñado por la compañía Sun Microsystems Inc., con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadoras formadas por más de un tipo de computadora, ya sean PC, MAC's, estaciones de trabajo, etc.), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa de Java puede ejecutarse en cualquier máquina o plataforma. El lenguaje fue diseñado teniendo en cuenta las siguientes características: (10)

- **Simple:** elimina la complejidad de los lenguajes como "C" y da paso al contexto de los lenguajes modernos orientados a objetos. La filosofía de programación orientada a objetos es diferente a la programación convencional.
- **Robusto:** maneja la memoria de la computadora. No hay necesidad de preocuparse por apuntadores, memoria que no se esté utilizando, etc. Java realiza todo esto sin necesidad de que el programador se lo indique.
- **Seguro:** tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de una computadora.
- **Portable:** como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computadora que tenga implementado el intérprete de Java.
- **Independiente a la arquitectura:** al compilar un programa en Java, el código resultante es un tipo de código binario conocido como byte code. Este código es interpretado por diferentes computadoras de igual manera, solamente hay que implementar un intérprete para cada plataforma. De esa manera Java logra ser un lenguaje que no depende de una arquitectura computacional definida.
- **Dinámico:** no requiere que se compilen todas las clases de un programa para que este funcione. Al realizar una modificación a una clase, Java se encarga de realizar un Dynamic Bynding o un Dynamic Loading para encontrar las clases.

Por las características planteadas anteriormente y por distinguirse Java como lenguaje de programación adecuado para aplicaciones de gran envergadura, es seleccionado para el desarrollo del SIIPOL, con el objetivo de garantizar un sistema robusto y con elevados índices de seguridad y confiabilidad.

1.5.6 Plataforma de Desarrollo

En un proyecto de software, para seleccionar la plataforma en la que se desarrollará el mismo, se toma en cuenta el entorno de ejecución. Cuando se necesita una solución informática multiplataforma, de alto rendimiento, escalabilidad y seguridad, la decisión más provechosa es sin lugar a dudas, la plataforma Java 2 Platform Enterprise Edition (J2EE); por esta razón, se toma como plataforma de desarrollo para el SIIPOL.

J2EE, es una plataforma de programación -parte de la Plataforma Java- para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java, con arquitectura de N niveles, distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones. J2EE define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems. J2EE simplifica además, las aplicaciones empresariales basándolas en componentes modulares y estandarizados, proveyendo un completo conjunto de servicios a estos componentes, y manejando varias de las funciones de la aplicación de forma automática, sin necesidad de una programación compleja. (11)

Con esta elección se garantiza que el SIIPOL sea una solución informática potente, robusta, eficiente y capaz de integrarse a otras aplicaciones dentro de la amplia gama de sistemas en el territorio venezolano.

1.5.7 Entorno de Desarrollo Integrado

Una vez seleccionada la plataforma J2EE solo restaría determinar el Entorno de Desarrollo Integrado (IDE). Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.

1.5.7.1 Red Hat Developer Studio

Red Hat Developer Studio combina productos aportados a Red Hat por Exadel en marzo de 2007 -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 más relevantes se encuentran: (12)

- Un modelo de programación unificado: Developer Studio aumenta y proporciona nuevas herramientas alrededor de JBoss Seam para construir aplicaciones de manera sencilla y consistente. Hoy, el tipo de aplicación típicamente dicta el modelo de programación que el desarrollador utiliza, lo que significa que deben aprender a utilizar diferentes modelos. JBoss, que actualmente está siendo estandarizado como Web Beans en el Java Community Process, ofrece un modelo unificado y sencillo para desarrollar cualquier clase de aplicación, eliminando la necesidad de múltiples modelos de programación.

- 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.
- Utilidades de la Plataforma Java para Aplicaciones Empresariales: Developer Studio hace más sencilla la construcción de aplicaciones Java 2 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 2 EE.
- Un tiempo de ejecución integrado con las herramientas de desarrollo: Red Hat Developer Studio es el primer entorno de desarrollo open source basado en Eclipse que une el runtime con las herramientas. Esto elimina la necesidad de que el desarrollador improvise estructuras y componentes open source antes de que empiece a desarrollar.

1.5.8 Herramienta CASE

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 una aplicación informática destinada a aumentar la productividad en el desarrollo de software reduciendo el coste de las misma en términos de tiempo y de dinero.

1.5.8.1 Visual Paradigm for UML

Visual Paradigm es una herramienta CASE profesional que de igual forma, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Utiliza UML como lenguaje de modelado y se integra con herramientas Java tales como: Eclipse, JBuilder, NetBeans, Oracle JDeveloper, entre otras. Es una herramienta que genera la documentación del proyecto automáticamente en varios formatos como Web o .Pdf, soporta la realización de ingeniería tanto directa como inversa y permite control de versiones. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos. Cabe destacar igualmente su robustez, usabilidad y portabilidad. (13)

Por sus cuantiosas y relevantes prestaciones fue seleccionado Visual Paradigm como herramienta CASE para el desarrollo del nuevo SIIPOL.

1.5.9 Frameworks utilizados en la solución

Un framework, constituye una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Para el desarrollo del sistema SIIPOL se llevó a cabo una selección de frameworks que brindan soporte a cada una de las capas componentes de la arquitectura, de las cuales se expondrá una breve descripción.

1.5.9.1 Presentación

JavaServer Faces (JSF) es un excelente framework para construir interfaces de usuario que ha alcanzado un merecido prestigio entre desarrolladores al punto de convertirse en un estándar para aplicaciones web basadas en Java siguiendo el patrón Modelo Vista Controlador (MVC). La construcción de interfaces de usuario suelen ser la parte más costosa del esfuerzo de desarrollo, especialmente por la dificultad de mantenimiento, sin embargo JSF nos ofrece un marco de trabajo que facilita el desarrollo de aplicaciones, separando las diferentes capas de una arquitectura: presentación, reglas y entidades de negocio.

Como uno de sus más connotados beneficios, es necesario destacar que la tecnología JavaServer Faces ofrece una amplia gama de componentes para el procesamiento de datos, validación de la entrada del usuario, y manejo de eventos.

De manera general JSF como framework de presentación ofrece:

- Un modelo de trabajo basado en componentes UI (interfaz de usuario), definidos por medio de etiquetas y XML.
- Una arquitectura basada en el patrón MVC.
- Asociación (de forma modular) de cada componente gráfico con los datos (beans de respaldo).

- Capa de control, definida de forma declarativa en archivos XML que facilita el control de eventos y errores.
- Validación en cliente y en servidor.
- Control de mensajes y roles.

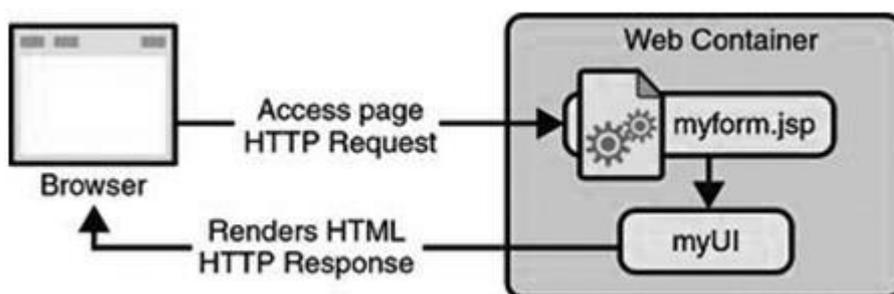


Ilustración 4: Estructura Cliente-Servidor de una aplicación Web.

Ajax4JSF es una extensión Open Source del estándar JavaServer Faces que añade las capacidades Ajax a las aplicaciones JSF sin necesidad de escribir código JavaScript. Presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. También permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el árbol de componentes JSF después de que la petición Ajax cambie los datos en el servidor. (14)

Los tags de Ajax4jsf permiten indicar una lista de identificadores de componentes que serán re-renderizados cuando la llamada Ajax sea procesada, actualizando automáticamente regiones independientes de la página que se hayan indicado previamente.

Dentro de las características más significativas de Ajax4jsf se encuentran:

- Soporte Ajax para páginas con componentes JSF que tienen comunicación asíncrona y actualizaciones de porciones de página.
- Event listeners, validaciones, transformaciones y mensajes.
- La capacidad de habilitar Ajax a componentes JSF sin cambios en el propio componente.
- Características de Ajax pero sin código JavaScript.

- Permite crear componentes propios con soporte built-in para Ajax.
- Trabaja en el lado del servidor.
- Soporta Facelets

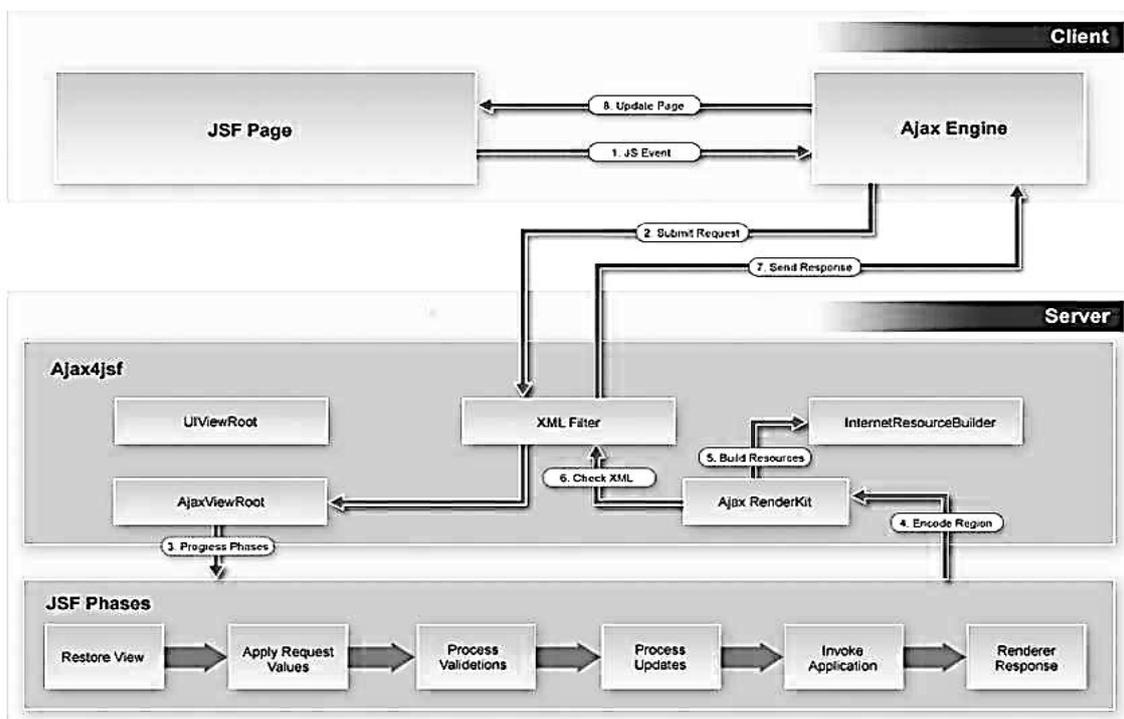


Ilustración 5: Flujo de petición de un componente Ajax4JSF.

RichFaces es una librería de componentes JSF avanzados con capacidades Ajax, proporciona componentes con funcionalidades ya desarrolladas, listos para usar en aplicaciones JSF que facilitan la programación de aplicaciones JavaServer Faces.

Surge como resultado de una asociación estratégica entre Red Hat y Exadel, otorgando a la comunidad del Open Source: JBoss RichFaces y JBoss Ajax4jsf, bajo la licencia LGPL en JBoss.org, por esta razón la distribución de la librería Ajax4JSF ya no se hace de manera independiente, sino junto con la librería RichFaces.

RichFaces facilita el desarrollo y permite la reusabilidad de componentes de una manera estándar y bastante bien documentada, pues los desarrolladores podrían crear sus propios componentes JSF artesanales con funcionalidad Ajax implementada con scripts propios, sin embargo RichFaces posibilita la programación con un extenso número de componentes listos para utilizarse.

Cuando la aplicación que se debe desarrollar tiene como requisito que el tiempo de respuesta de interacción con los componentes visuales sea corto, es entonces cuando se piensa en un mecanismo de petición/respuesta asíncrono, es decir, la petición lanzada desde la aplicación va al servidor e internamente manipula información y da una respuesta de regreso mientras el usuario está haciendo otras cosas en esa misma página. Facilidades como estas quedan totalmente cubiertas con el uso de RichFaces.

1.5.9.2 Lógica de Negocio

Spring es un framework que ha venido a revolucionar la manera de programar aplicaciones Java debido a la facilidad de crear componentes reutilizables, además de integrarse fácilmente con otros frameworks como lo son Hibernate, iBatis, Struts, entre otros; formando una poderosa herramienta para el desarrollo de aplicaciones empresariales. Spring constituye además un framework de código abierto que interviene en todas las capas arquitectónicas de una aplicación J2EE, y brinda soporte a Java Server Faces. (15)

A pesar de que Spring no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y un sustituto del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Dentro de las ventajas que ofrece Spring, se encuentran:

- Facilita la manipulación de los objetos.
- Reduce la proliferación de *Singletons*.
- Elimina la necesidad de usar distintos y variados tipos de ficheros de configuración.
- Permite el uso de la programación orientada a aspectos.

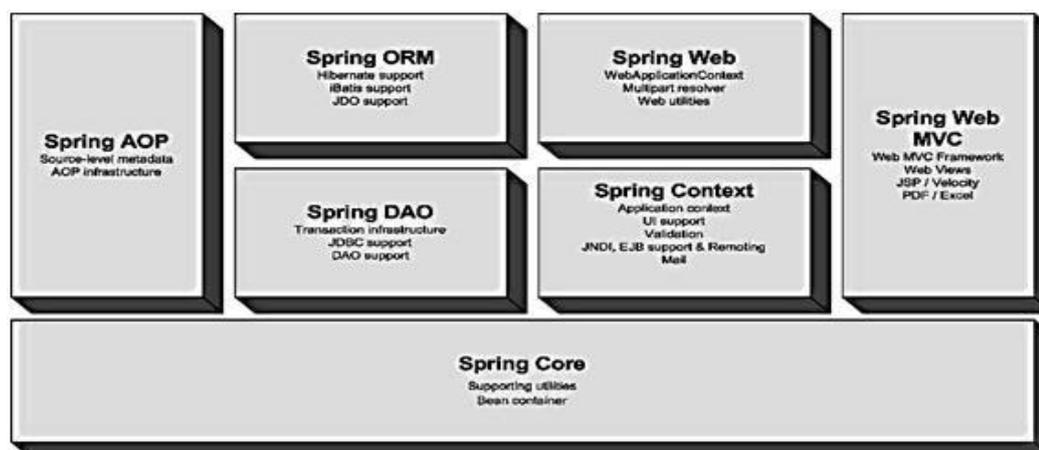


Ilustración 6: Estructura de Spring.

1.5.9.3 Acceso a Datos

Hibernate: solución ORM (Object-Relational Mapping) para Java, constituye el framework utilizado en el desarrollo de la capa de acceso a datos del SIIPOL, se selecciona a partir de sus características y ventajas. Aunque esta selección no se encuentra en el marco de este trabajo, resulta de vital importancia para la fundamentación teórica del mismo una breve explicación del framework Hibernate. (16)

Hibernate es una capa de persistencia objeto/relacional y un generador de sentencias sql. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada puede generar bases de datos en cualquiera de los entornos soportados: Oracle, DB2, MySql, etc.

Hibernate tiene como objetivo fundamental solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el orientado a objetos y el relacional. Para lograr esto le permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate hace posible a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la programación orientada a objetos. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos. Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a

partir de la información disponible. Asimismo ofrece un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate además es Open Source y la licencia del producto está eximida de costo.

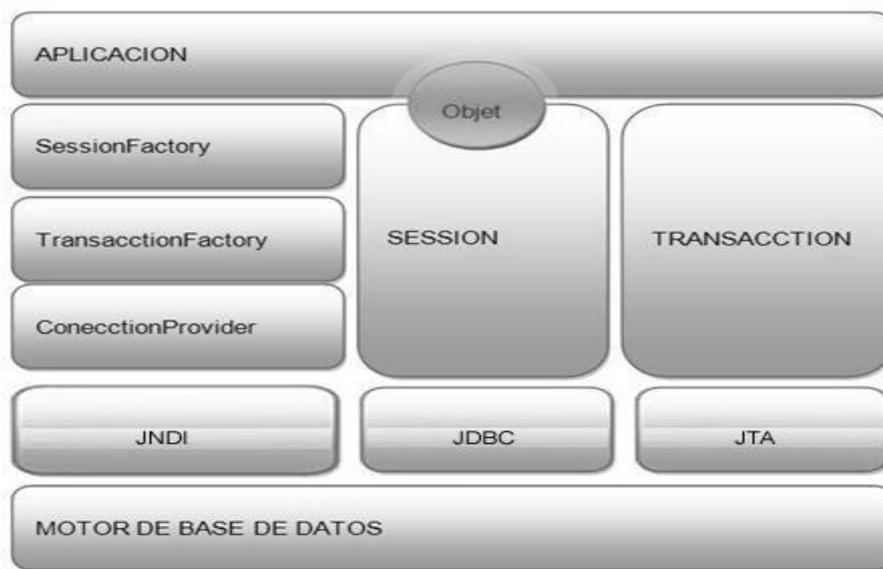


Ilustración 7: Arquitectura de Hibernate.

1.5.10 Sistema Gestor de Base de Datos

Oracle es básicamente una herramienta cliente/servidor para la gestión de base de datos la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general.

Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos (Oracle) hasta sistemas de gestión. Cuenta además, con herramientas propias de desarrollo para realizar potentes aplicaciones, como Oracle Designer. (17)

Aunque la utilización de esta potente herramienta es excesivamente costosa debido al precio de su licencia, fue seleccionado a raíz de ser considerado el motor de base de datos relacional más potente a nivel mundial; puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador. Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa muy fuerte. Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso

ciertas versiones admiten la administración de bases de datos distribuidas. El software del servidor puede ejecutarse en multitud de sistemas operativos.

Oracle ha sido diseñada para que las organizaciones puedan controlar y gestionar grandes volúmenes de contenidos no estructurados en un único repositorio con el objetivo de reducir los costes y los riesgos asociados a la pérdida de información.

1.6 Arquitectura Técnica

Como propuesta de arquitectura para el Sistema de Investigación e Información Policial se define una propuesta de solución cliente-servidor de un alto nivel que especifica una aplicación que hace uso del protocolo HTTP, con cuatro servidores HP Integrity que facilitan la gestión de múltiples necesidades empresariales en un único entorno optimizando la potencia del centro de datos donde será instalado y reduciendo los costos drásticamente. En los dos primeros se montará el servidor de aplicaciones, utilizando un failover y en los restantes un servidor de base de datos Oracle conectado en clúster los que estarán corriendo de manera continua, no obstante, de existir problemas en los mismos, existe un centro de contingencia el cual tiene dos servidores HP Integrity, uno para la aplicación web y otro para la base de datos.

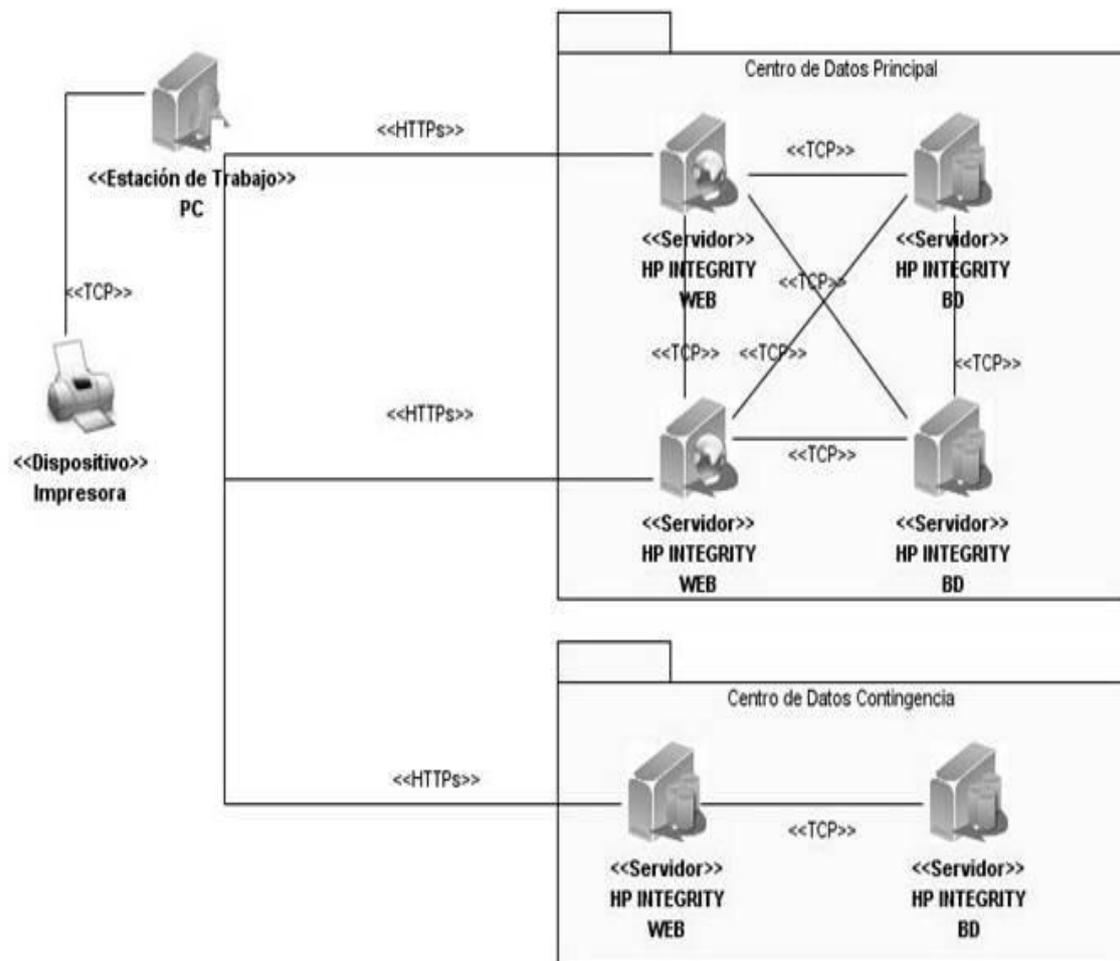


Ilustración 8: Sistema Cliente-Servidor.

El estilo arquitectónico en capas tomará parte en la solución proporcionándole ventajas sustanciales como lo son: la centralización de los aspectos de seguridad y transaccionalidad, no replicación de lógica de negocio en los clientes posibilitando así, que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios reduciendo los costes de mantenimiento y garantizando a su vez una mayor sencillez de los clientes.

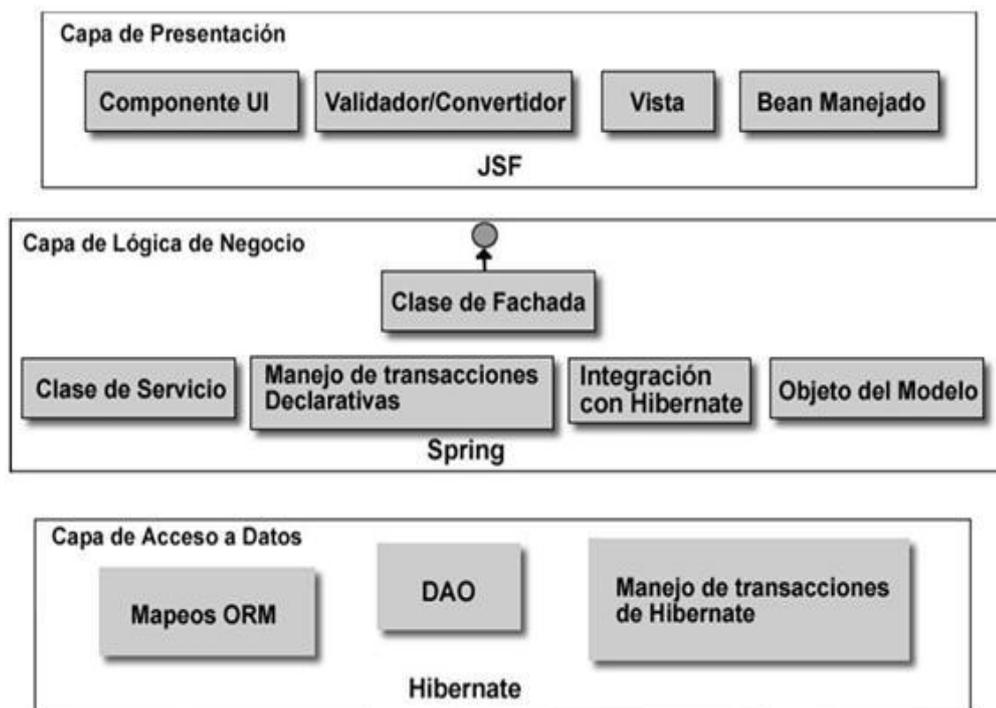


Ilustración 9: Modelo N-Capas.

Como patrón arquitectónico se elige MVC (Modelo Vista Controlador), el cual mediante la división de la aplicación en tres partes fundamentales: **modelo**, que contiene la lógica de negocio de la aplicación; **vista**, que muestra al usuario la información que éste necesita y **controlador**, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización, facilitará a la solución mayor sencillez para crear distintas representaciones de los mismos datos así como la realización de pruebas unitarias de los componentes, la posibilidad de aplicar desarrollo guiado por pruebas, reutilización de los componentes simplicidad en el mantenimiento de los sistemas capacidad para desarrollar prototipos rápidos y escalabilidad en el sistema.

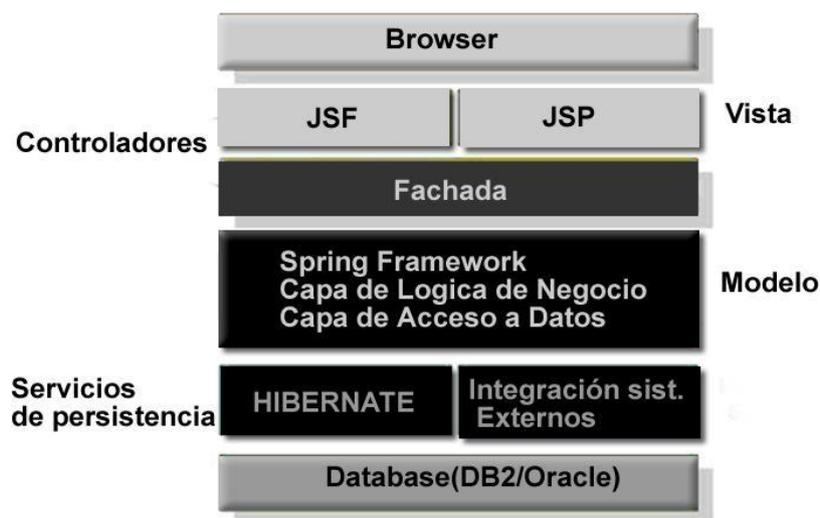


Ilustración 10: Modelo Vista Controlador.

En términos informáticos la palabra seguridad es asociada comúnmente al acceso restringido según roles específico. El manejo de la seguridad a un alto nivel no forma parte del alcance del presente trabajo de diploma, sin embargo por su importancia en el sistema se hace mención de algunas cuestiones de interés. En el SIIPOL la seguridad se garantiza con Acegi Security que constituye un framework ampliamente difundido para modelar los requisitos de seguridad en aplicaciones web. Acegi usa una cadena de al menos 3 filtros para manejar la seguridad de la aplicación dentro de los cuales se encuentra el filtro `AuthenticationProcessingFilter`, el filtro `HttpSessionContextIntegrationFilter` y el filtro `ExceptionHandlerFilter`. Dentro de sus ventajas más reconocidas se encuentra: la configuración no intrusiva pues la totalidad del sistema de seguridad puede funcionar en una aplicación web utilizando los filtros que proporciona.

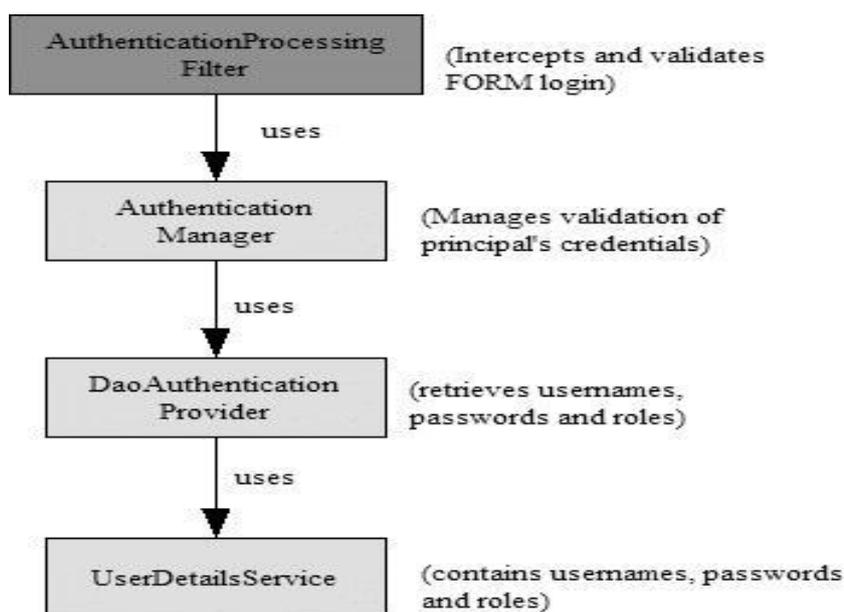


Ilustración 11: Filtro AuthenticationProcessingFilter y sus dependencias.

1.7 Conclusiones

Con la culminación de este capítulo se dio a conocer un conjunto de elementos que marcan el punto de partida del cual se iniciará la construcción de un software que cumpla con los objetivos propuestos.

El estudio del estado actual en el que se encuentra inmersa la Gestión Policial y la exposición de las tendencias mayormente utilizadas en el mundo contribuyeron de manera importante en la creación de la propuesta general para el nuevo SIIPOL.

Varias de las aplicaciones expuestas, poseen características que serán adoptadas en la construcción del Software que se desea desarrollar.

La utilización de la metodología seleccionada facilitará el seguimiento y monitoreo de un proceso de desarrollo que agilice el trabajo de los implicados en el mismo, de ahí que la obtención de un producto con la calidad requerida, sea un hecho seguro.

La utilización de las herramientas y las definiciones arquitectónicas mencionadas permitirá el desarrollo claro y fluido de un sistema construido sobre bases sólidas y un entorno de desarrollo bien definido.

Capítulo 2. Análisis, Diseño e Implementación de la Propuesta de Solución

2.1 Introducción

En este capítulo se desarrollará la propuesta de solución dada a conocer con anterioridad. En la solución, el subsistema Investigación Criminalística se dividió en tres módulos, los cuales son Experticias Criminalísticas, Control de Investigaciones Criminalísticas y Solicitudes. Experticias Criminalísticas se encarga de la manipulación automatizada de las respuestas de los especialistas en determinadas áreas de las ciencias criminalísticas a las solicitudes que son recibidas a través de una cadena de comunicación provista por el software o físicamente.

El módulo Experticias Criminalísticas cuenta con diecinueve casos de uso, de los cuales en este capítulo se abordará el desarrollo de tres de ellos, a partir de un estudio realizado y por su representatividad de acuerdo a los componentes utilizados, fueron seleccionados los siguientes Casos de Uso: Gestionar Trayectoria Intraorgánica, Gestionar Trayectoria Balística y Gestionar Resultado de Inspección Técnica de Robo a Banco.

2.2 Análisis y Diseño de la Solución Propuesta

AUP, metodología seleccionada para el desarrollo del módulo Experticias Criminalísticas en su versión 2.0, no concibe el Análisis y Diseño como Flujo de Trabajo donde tiene lugar el refinamiento de los requisitos, la identificación de las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Como fue abordado en el capítulo anterior, AUP define el Modelo, como primer flujo de trabajo que abarca las disciplinas Modelo de Negocio, Requerimientos y Análisis y Diseño prefijadas por RUP.

Se hace necesario destacar que no se obvia el análisis y el diseño, evidentemente es necesario a la hora de desarrollar un sistema tener en cuenta cómo será implementado el software a partir de las funcionalidades previstas y las restricciones impuestas, sin embargo en el Proceso Ágil Unificado estos procesos tienen lugar a través de la realización de tarjetas CRC (Class, Responsibilities, Collaborators), técnica para la representación de sistemas orientado a objetos que brinda la posibilidad de modelar las responsabilidades de las clases y su interacción.

Como metodología que promueve el desarrollo ágil, AUP usa un enfoque basado en el valor para construir software, colaborando con el cliente e incorporando los cambios continuamente. Los documentos no pueden sustituir, ni pueden ofrecer la riqueza y generación de valor que se logra con la comunicación directa entre las personas y a través de la interacción con los prototipos. Por eso, siempre que sea posible debe preferirse, y reducir al mínimo indispensable el uso de documentación que genera trabajo y no aporta un valor directo al producto.

La utilización de la técnica CRC contribuyó en gran medida a la comunicación entre los miembros del equipo de desarrollo, modelar el sistema de forma tal que los desarrolladores hablen el mismo idioma con respecto al estilo de trabajo y prioridad de funcionalidades a tener en cuenta para la implementación, cuestiones que sin lugar a dudas influyeron de manera determinante en la fluidez durante el proceso.

2.2.1 Diagramas de Paquetes

El módulo Experticias Criminalísticas a partir de la complejidad de sus procesos, presenta un estrecho vínculo con diversos componentes del sistema SIIPOL, los cuales brindan interfaces que permiten el acceso a los servicios necesarios para la gestión de los mismos. Dentro de los componentes de mayor dependencia se encuentra Análisis de Información, subsistema que proporciona las funcionalidades concretas para el trabajo con Funcionarios, Armas, Vehículos, entre otros elementos necesarios para la gestión de experticias. Asimismo se da el caso del subsistema de Administración, Investigación Forense, Investigación Penal. El diagrama de paquetes ilustra de forma gráfica la integración del Módulo Experticias Criminalísticas con otros componentes o partes del sistema.

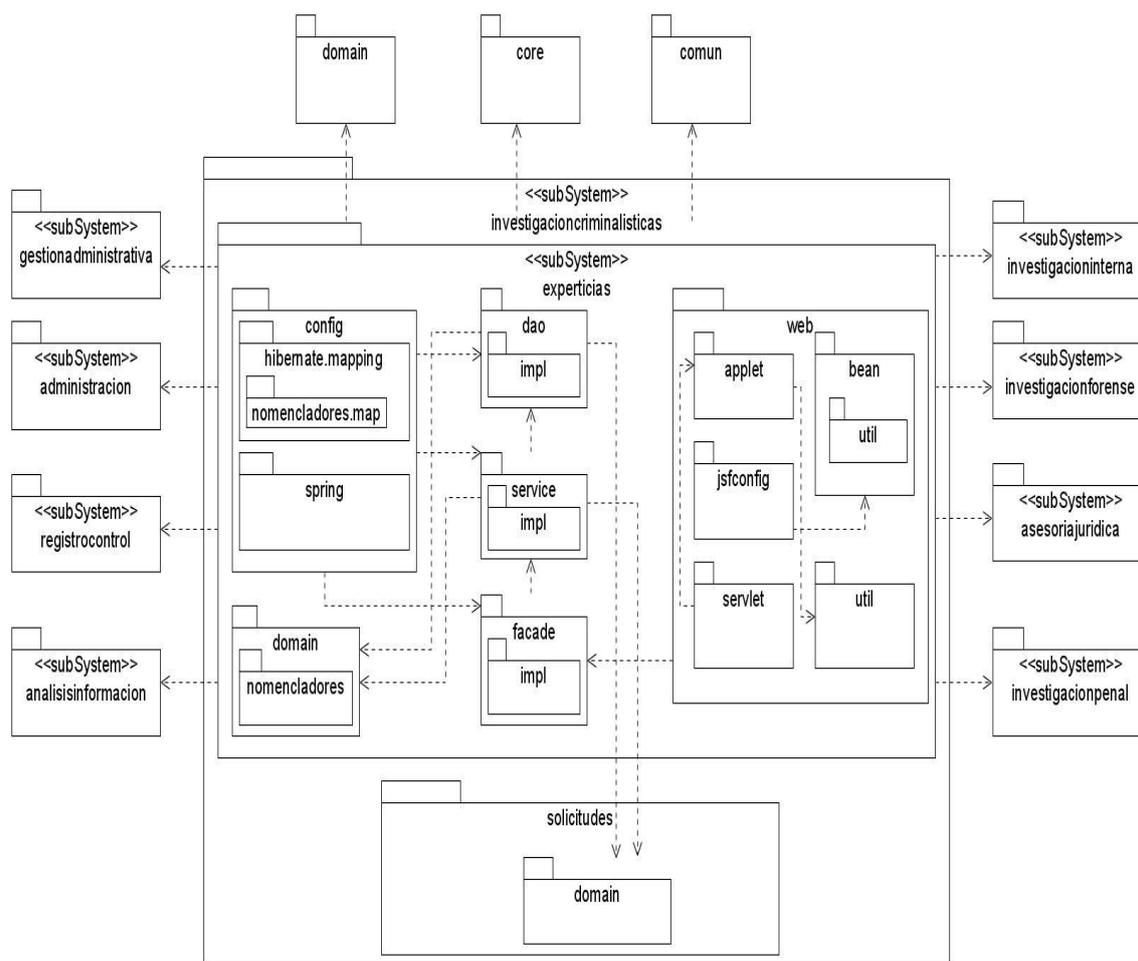


Ilustración 12: Diagrama de Paquetes.

2.2.2 Patrones de Diseño. Tarjetas CRC

2.2.2.1 Patrones de Diseño.

Un patrón de diseño es una abstracción de una solución en un alto nivel. Constituye la respuesta a un problema de diseño no trivial que es efectiva (resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (puede aplicarse a diferentes problemas de diseño en distintas circunstancias). Cuando se tienen conocimientos sobre el comportamiento y los beneficios de los patrones de diseño, uno de los errores más comunes que suele cometerse es querer aplicarlos en todo momento en el sistema que se esté desarrollando, provocando así su uso excesivo muchas veces sin realizar un previo análisis para determinar si sería efectivo realmente y aportaría la reusabilidad, extensibilidad y mantenimiento que se necesita en la solución desarrollada.

Con el propósito de lograr la simplicidad y eficiencia en el sistema, luego de un estudio exhaustivo para no incurrir en el error antes mencionado, se seleccionaron un conjunto de patrones de diseño a utilizar en la solución, cuyas principales características serán enunciadas seguidamente.

Capa Presentación.

- Front – Controller

Es recomendable en aplicaciones web hacer uso del patrón Front-Controller (Controlador Frontal) pues proporciona un punto de entrada único que controla y gestiona las peticiones web realizadas por los clientes, evitando así tener que repetir la misma lógica de control en todos los .jsp. Normalmente se utiliza junto con un Dispatcher que es el responsable de redirigir el flujo de ejecución hacia el jsp adecuado. Este Dispatcher puede ser realizado por el propio Controlador Frontal ó estar modelado en una clase diferente.

- Composite View

El patrón Composite View (Vista Compuesta) es utilizado cuando varias vistas compuestas utilizan sub-vistas similares así como de igual forma cuando las porciones atómicas del contenido de una vista cambian con frecuencia. Composite View permite crear vistas compuestas de varias sub-vistas de forma modular, flexible y extensible para construir vistas de páginas JSP para aplicaciones J2EE. Por ejemplo, una página JSP que incluye otras páginas JSP y HTML usando la directiva include o el action include es un patrón Composite View.

- Composite

Composite (Compuesto) como patrón de diseño permite componer objetos dentro de estructuras de árbol para representar jerarquías parte-todo. Se utiliza cuando se requieran clientes capaces de ignorar la diferencia entre composiciones de objetos y objetos individuales, clientes que puedan tratar todos los objetos de la estructura compuesta de forma uniforme. Una de sus ventajas más notables radica en la simplicidad que aporta a los clientes.

Capa de Negocio.

- Facade

El patrón de diseño Facade (Fachada) se ocupa de proveer una interfaz unificada y sencilla que haga de intermediaria entre un cliente y una interfaz ó grupo de interfaces más complejas. La idea principal es la de ocultar todo lo posible la complejidad de un sistema, el conjunto de clases ó componentes que lo forman, de manera que solo se ofrezca un (o unos pocos) punto de entrada al sistema a través de la fachada permitiendo así la flexibilidad en el desarrollo de sistemas.

- Service Locator

Service Locator constituye un patrón de diseño que provee un punto de acceso centralizado a determinados servicios. Abstrae la complejidad al encapsular procesos de búsqueda y creación manteniéndolos ocultos del cliente, proporcionándoles a su vez un acceso uniforme a los servicios. Facilita además la adición de nuevos componentes de negocio sin que los clientes se vean impactados.

- Abstract Factory

El patrón Abstract Factory (Factoría Abstracta) pertenece al grupo de los patrones creacionales de Objetos que proveen soluciones más escalables y dinámicas comparadas con los patrones creacionales de Clases. Tienen la responsabilidad de crear instancias de objetos de otras clases y el conocimiento necesario para encapsular la forma en que se crean determinados tipos de objetos en una aplicación. Abstract Factory proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

Capa de Acceso a Datos.

- Data Access Object (DAO)

El patrón DAO constituye la solución a la necesidad de gestionar una diversidad de fuentes de datos, encapsula dicha fuente y oculta la forma de acceder a los datos. Este patrón de diseño posibilita que el software cliente se centre en la información que necesita y se olvide de cómo se realiza el acceso a los recursos o de cuál es la fuente de almacenamiento.

2.2.2.2 Tarjetas CRC.

Diligencia	
Description: Entidad persistente que guarda los datos de una Diligencia.	
Responsibilities:	
Name	Collaborator
Fecha y Hora	
Acta Investigativa	ActaInvestigativa
Acta Procesal Externa	ActaProcesalExterna
Relaciones entre Diligencias y Personas	DiligenciaPersona
Relaciones entre Diligencias y Elementos	DiligenciaElemento
Relaciones entre Diligencias Internas y Personas	DiligenciaPersonaInterna
Relaciones entre Diligencias Internas y Elementos	DiligenciaElementoInterna
Relaciones entre Diligencias y Funcionarios	FuncionarioDiligencia
Tipo de la Diligencia	TipoDiligencia
Estado de la Diligencia	EstadoDiligencia
Comunicaciones	ComunicacionDiligencia
Artículos de una Ley	Articulo
Auto Inicio de la Diligencia	AutoInicioTDiligencia
Comentarios	ComentarioRevision

Informe	
Super Classes: Diligencia	
Description: Entidad persistente que guarda los datos de un Informe.	
Responsibilities:	
Name	Collaborator
Solicitud	SolicitudExperticia
Número de Informe	
Tipo de Informe	SubInforme
Evidencias asociadas	Evidencia
Sitio de Suceso asociado	SitioSuceso
Fotos asociadas	Foto

Ilustración 13: Tarjetas CRC para la jerarquía de Diligencia.

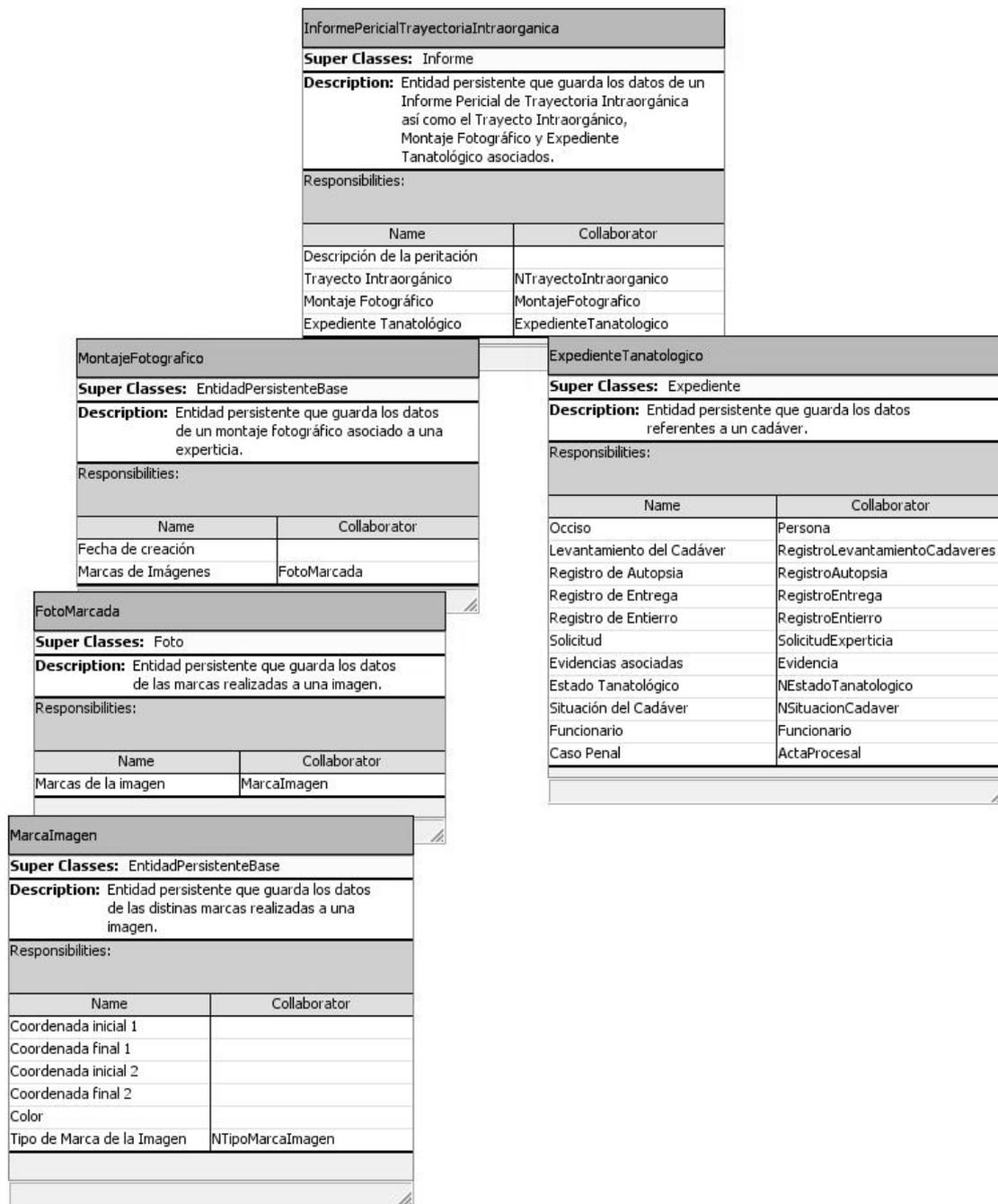


Ilustración 14: Tarjetas CRC. CU Gestionar Trayectoria Intraorgánica.

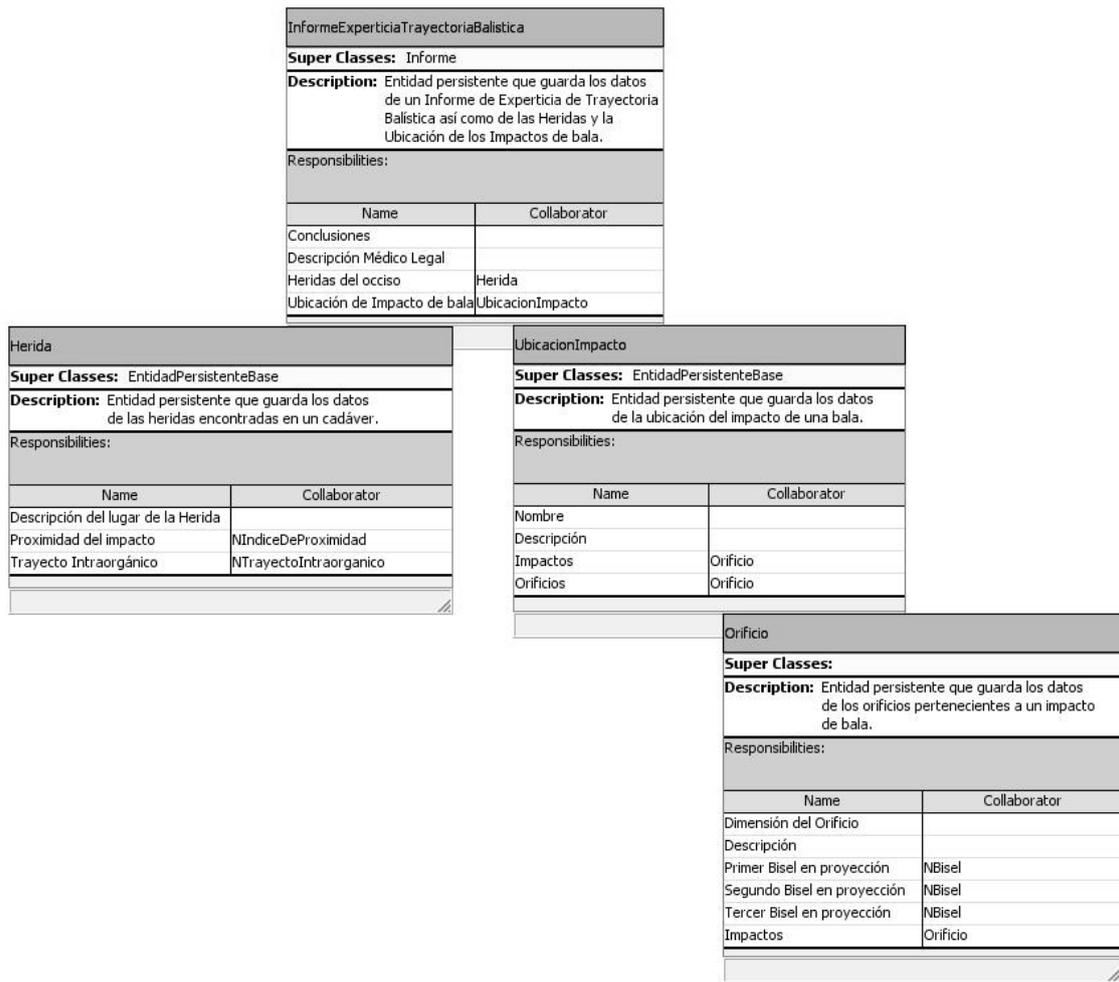


Ilustración 15: Tarjetas CRC. CU Gestionar Trayectoria Balística.

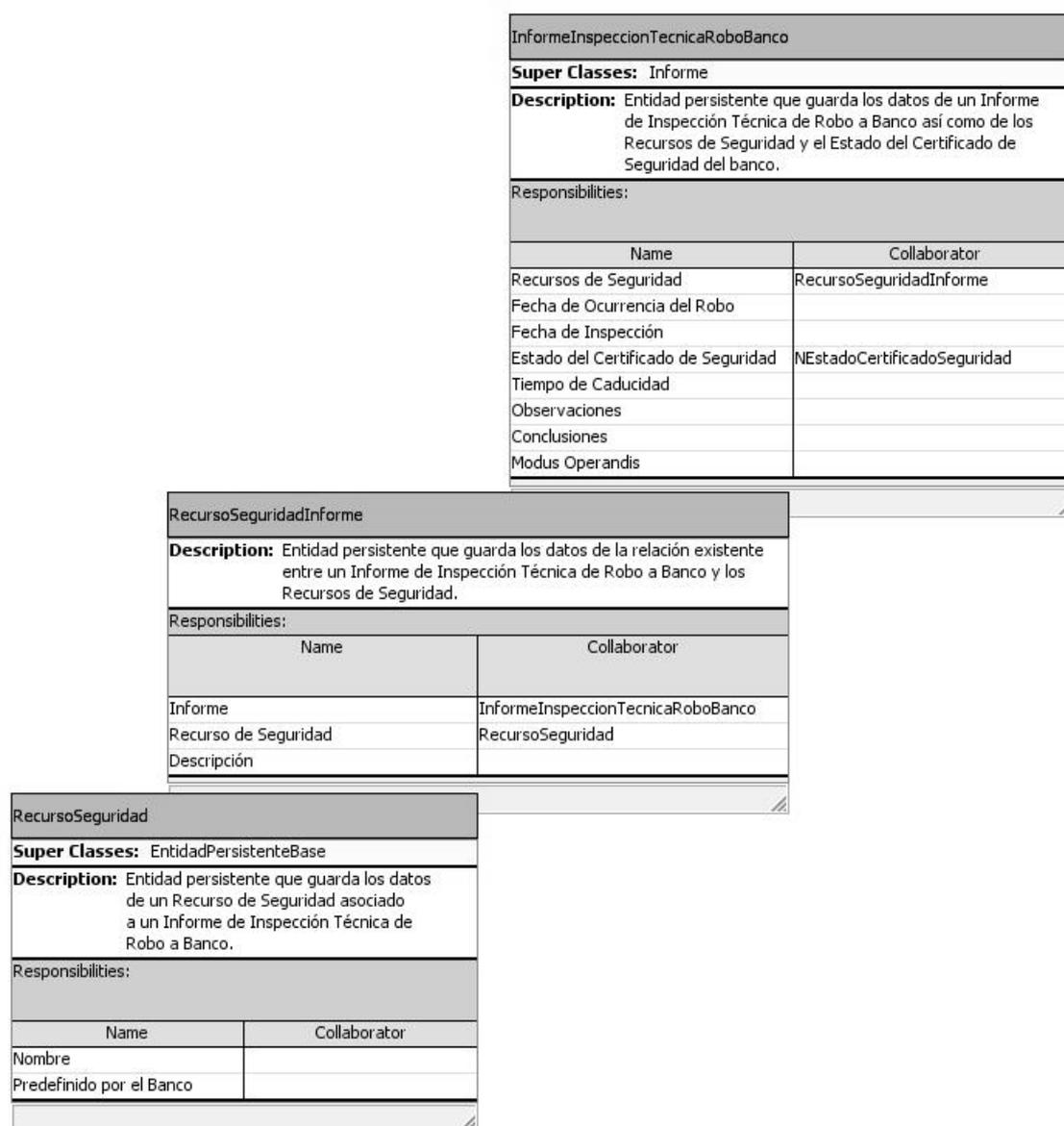


Ilustración 16: Tarjetas CRC. CU Gestionar Resultado de Inspección Técnica de Robo a Banco.

Para la solución se tuvo en cuenta la creación de clases que agrupan funcionalidades comunes, con el fin de eliminar la repetición innecesaria de código y lograr una mayor flexibilidad en el sistema a la hora de realizar futuras refactorizaciones, pedidos de cambio solicitados por el cliente ó la incorporación de nuevos componentes desarrollados por el equipo de Arquitectura, evidenciando así la marcada importancia a la simplicidad y la capacidad de respuesta a cambios durante el proceso de desarrollo, beneficios que ofrecen las metodologías ágiles como AUP. Estas clases son descritas a continuación.

Nombre: InformeUtil	
Descripción: Clase que agrupa funcionalidades comunes en consecuencia a la lógica de negocio de Experticias Criminalísticas.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	
Nombre:	asociarElementosACasoViaDiligencia(Diligencia informe, TipoRelElemento tipoRelacionElemento, NTRelAveriguacionPreliminarElemento tipoRelacionElementoInterna)
Descripción:	Funcionalidad encargada de asociar Elementos (arma, vehículo, objeto) a un Acta Investigativa especificando el tipo de relación que se establece de acuerdo al tipo acta asociada a la diligencia: Acta Procesal ó Acta Disciplinaria.
Nombre:	disociarElementosACasoViaDiligencia(Diligencia informe)
Descripción:	Funcionalidad encargada de disociar Elementos (arma, vehículo, objeto) del Acta Investigativa asociada a la diligencia.
Nombre:	poblarDiligencia(Diligencia diligencia, boolean emitido)
Descripción:	Funcionalidad encargada de asociar a la diligencia los datos que se necesitan para insertar la entidad en la base de datos.

Nombre: Informe	
Descripción: Clase genérica que agrupa elementos comunes de un informe y de la cual heredan los informes del submódulo Experticias Criminalísticas.	
Atributo	Tipo
Solicitud	SolicitudExperticia
noInforme	String
Tipo	SubInforme
evidenciasAsociadas	Set<Evidencia>
sitioSuceso	SitioSuceso
fotosAsociadas	Set<Foto>
Responsabilidades:	
Nombre:	getDependenciaSolicitante()
Descripción:	Funcionalidad encargada de determinar la dependencia a la que pertenece el acta investigativa asociada al informe.
Nombre:	getComunicacion()
Descripción:	Funcionalidad encargada de obtener la comunicación asociada a la solicitud relacionada al informe.
Nombre:	getDestinatarios()
Descripción:	Funcionalidad que se encarga de obtener los despachos interesados de la solicitud y a los cuales será enviado el informe.
Nombre:	getEstadoActual()
Descripción:	Funcionalidad que se encarga de obtener el estado actual en el que se encuentra el informe (creado, revisado, en curso, entre otros).

Nombre:	getFuncionarioCreador()
Descripción:	Funcionalidad encargada de obtener el funcionario creador del informe.
Nombre:	getFuncionarioCorrector()
Descripción:	Funcionalidad encargada de obtener el funcionario que corrigió el informe.
Nombre:	getFuncionarioResponsable()
Descripción:	Funcionalidad encargada de obtener el funcionario que responsable del informe.

Nombre: ExperticiaCriminalisticaDaoImpl	
Descripción: Clase encargada de implementar la interfaz que presenta métodos de igual firma y tiene el mismo nombre sin el sufijo Impl, Engloba funcionalidades comunes de interacción con la base de datos para el submódulo Experticias Criminalísticas.	
Atributo	Tipo
carga	Carga
Responsabilidades:	
Nombre:	obtenerPorId(Class<? extends T> clase, Integer id)
Descripción:	Funcionalidad encargada de obtener una entidad dado su identificador.
Nombre:	obtenerTodos(Class<? extends T> clase, List<String> cargas)
Descripción:	Funcionalidad encargada de obtener todos los elementos de un tipo específico de entidad.
Nombre:	obtenerNomenclador(Class<? extends T> clase, Integer id)
Descripción:	Funcionalidad encargada de obtener un nomenclador dado su identificador.
Nombre:	inicializar(T object, List<String> cargas)
Descripción:	Funcionalidad que se encarga de inicializar un conjunto de propiedades pasadas por parámetro de la entidad especificada.
Nombre:	inicializarAtributosComunesInforme(Informe informe)
Descripción:	Funcionalidad que se encarga de inicializar los atributos comunes de un informe determinado.
Nombre:	inicializarSolicitud(Informe informe)
Descripción:	Funcionalidad que se encarga de inicializar los datos de la solicitud asociada a un informe determinado.
Nombre:	inicializarSolicitud(SolicitudExperticia solicitud)
Descripción:	Funcionalidad que se encarga de inicializar los datos de una diligencia de tipo solicitud de experticia.
Nombre:	inicializarComunicacion(Informe informe)
Descripción:	Funcionalidad que se encarga de inicializar los datos de la comunicación asociada a un informe determinado.

Nombre:	inicializarFuncionariosDiligencia(Diligencia diligencia)
Descripción:	Funcionalidad que se encarga de inicializar los datos de los funcionarios relacionados con una diligencia determinada.

Nombre: InformeCriminalisticaServiceImpl	
Descripción: Clase encargada de implementar la interfaz que presenta métodos de igual firma y tiene el mismo nombre sin el sufijo Impl. Constituye la súper clase inmediata de todos los servicios de informes criminalísticos y utiliza la clase InformeUtil que engloba funcionalidades comunes de dichos informes.	
Atributo	Tipo
investigacionPenalFacade	InvestigacionPenalFacade
nomencladorInvestigacionPenalFacade	NomencladorInvestigacionPenalFacade
informeUtil	InformeUtil
experticiaCriminalisticaDao	ExperticiaCriminalisticaDao
Responsabilidades:	
Nombre:	guardarEnviarInformeExperticia(Informe informe, boolean accion)
Descripción:	Funcionalidad encargada de realizar las acciones de incluir/actualizar un informe determinado de acuerdo a la acción especificada por parámetro invocando la clase InformeUtil para proveer el objeto de los datos necesarios para ejecutarlas.

Nombre: InvestigacionCriminalisticaExperticiasFacadeImpl	
Descripción: Clase encargada de implementar la interfaz que presenta métodos de igual firma y tiene el mismo nombre sin el sufijo Impl. Constituye la implementación de la fachada del submódulo Experticias Criminalísticas donde tienen lugar las llamadas a los diferentes servicios para la ejecución de las funcionalidades requeridas.	
Atributo	Tipo
informePericialSiniestroService	InformeCriminalisticaService
informePrevioService	InformePrevioService
informeInspeccionTecnicaEspecializadaService	InformeInspeccionTecnicaEspecializadaService
informeSitioSuceso	InformeCriminalisticaService
informeMicElectronica	InformeCriminalisticaService
informePericialService	InformeCriminalisticaService
gestionSolicitudService	GestionSolicitudService
informePericialReconstruccionHechosService	InformeCriminalisticaService
informePericialTrayectoriaIntraorganicaService	InformePericialTrayectoriaIntraorganicaService
informeRetratoHabladoService	InformeCriminalisticaService
informeExperticiaDeVehiculoService	InformeExperticiaDeVehiculoService
informeInspeccionTecnicaRoboBancoService	InformeInspeccionTecnicaRoboBancoService
informePericialExperticiaContableService	InformeCriminalisticaService
informeInspeccionTecnicaInmuebleService	InformeCriminalisticaService
informeExperticiaDeAvaluoService	InformeCriminalisticaService
registroHistoricoInformeService	RegistroHistoricoInformeService
informeIdentificacionDelIndividuoService	InformeIdentificacionDelIndividuoService

	vice
informeExperticiaTrayectoriaBalisticaService	InformeCriminalisticaService
informeResultadoExperticiaBalisticaService	InformeResultadoExperticiaBalisticaService
informeNotificacionReveladoRolloFotograficoService	InformeNotificacionReveladoRolloFotograficoService
seguridadFacade	SeguridadFacade
Responsabilidades:	
Nombre:	guardarEnviarInformeInspeccionTecnicaEspecializada(InformeInspeccionTecnicaEspecializada informe, boolean accion)
Descripción:	Funcionalidad encargada de realizar las acciones de incluir/actualizar un informe de inspección técnica especializada de acuerdo a la acción especificada por parámetro.
Nombre:	obtenerInformeInspeccionTecnicaespecializada(Integer idInforme)
Descripción:	Funcionalidad encargada de obtener un informe de inspección técnica especializada dado su identificador.
Nombre:	guardarEnviarInformePericialTrayectoriaIntraorganica(InformePericialTrayectoriaIntraorganica informe, boolean accion)
Descripción:	Funcionalidad encargada de realizar las acciones de incluir/actualizar un informe pericial de trayectoria introaorgánica de acuerdo a la acción especificada por parámetro.
Nombre:	obtenerInformePericialTrayectoriaIntraorganica(Integer id)
Descripción:	Funcionalidad encargada de obtener un informe pericial de trayectoria introaorgánica dado su identificador.
Nombre:	obtenerInformeIntraorganicaSegunActaProcesal(String actaProcesal, String actaProcesalExterna)
Descripción:	Funcionalidad encargada de obtener los informes periciales de trayectoria introaorgánica asociados a un acta procesal ó un acta procesal externa.
Nombre:	guardarEnviarInformeInspeccionTecnicaRoboBanco(InformeInspeccionTecnicaRoboBanco informe, boolean accion)
Descripción:	Funcionalidad encargada de realizar las acciones de incluir/actualizar un informe de inspección técnica de robo a banco de acuerdo a la acción especificada por parámetro.
Nombre:	obtenerInformeInformeInspeccionTecnicaRoboBancoPorId(Integer idInforme)
Descripción:	Funcionalidad encargada de obtener un informe de inspección técnica de robo a banco dado su identificador.
Nombre:	obtenerRecursosSeguridadPredeterminados()
Descripción:	Funcionalidad encargada de obtener la lista de los recursos de seguridad predeterminados por la Entidad Financiera.
Nombre:	guardarEnviarInformeExperticiaTrayectoriaBalistica(InformeExperticiaT

	rayectoriaBalistica informe, boolean accion)
Descripción:	Funcionalidad encargada de realizar las acciones de incluir/actualizar un informe de experticia de trayectoria balística de acuerdo a la acción especificada por parámetro.
Nombre:	obtenerPorIDInformeExperticiaTrayectoriaBalistica(int id)
Descripción:	Funcionalidad encargada de obtener un informe de experticia de trayectoria balística dado su identificador.

Nombre: CriminalisticaBaseBean	
Descripción: Clase que representa el bean manejado padre de la herencia de todos los beans manejados del submódulo Experticias Criminalísticas.	
Atributo	Tipo
Acción	String
nomencladorComunFacade	NomencladorComunFacade
gestionarBaseLegalSubviewManejado	GestionarBaseLegalSubviewManejado
Responsabilidades:	
<i>Esta clase no presenta responsabilidades, solo métodos de acceso a sus atributos.</i>	

Nombre: IIncluirExperticia	
Descripción: Clase que representa la interfaz a implementar por todos los beans manejados del submódulo Experticias Criminalísticas con la responsabilidad de incluir una experticia.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	
Nombre:	cancelar(ActionEvent e)
Descripción:	Funcionalidad encargada de cancelar las acciones y regresar a la página que invocó el escenario actual del caso de uso.
Nombre:	guardarEnviar()
Descripción:	Funcionalidad encargada de incluir un elemento en el sistema. Si la opción seleccionada por el usuario es la de <i>Guardar</i> , el elemento toma estado <i>En Curso</i> dando la posibilidad de ser modificado posteriormente, no así si es seleccionada la opción <i>Incluir</i> que coloca el elemento en estado <i>Emitido</i> .
Nombre:	isInicializarIncluidos()
Descripción:	Funcionalidad encargada de inicializar las subvistas incluidas en el bean manejado.
Nombre:	limpiarCampos(ActionEvent e)
Descripción:	Funcionalidad encargada de retornar al estado inicial los campos editables de la página.
Nombre:	personalizarComunicacion(ActionEvent e)
Descripción:	Funcionalidad encargada de navegar hacia el caso de uso extendido Configurar Encabezado

Nombre:	vistaPrevia()
Descripción:	Funcionalidad encargada de navegar hacia la UI Vista Previa.

Nombre: IVerExperticia	
Descripción: Clase que representa la interfaz a implementar por todos los beans manejados del submódulo Experticias Criminalísticas con la responsabilidad de ver los datos de una experticia.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	
Nombre:	cerrar(ActionEvent actionEvent)
Descripción:	Funcionalidad encargada de regresar a la página que invocó el escenario actual del caso de uso.
Nombre:	eliminar()
Descripción:	Funcionalidad encargada de navegar hacia la UI Eliminar Informe.
Nombre:	isInicializarIncluidos()
Descripción:	Funcionalidad encargada de inicializar las subvistas incluidas en el bean manejado.
Nombre:	modificar()
Descripción:	Funcionalidad encargada de navegar hacia la UI Modificar Informe.

Nombre: IModificarExperticia	
Descripción: Clase que representa la interfaz a implementar por todos los beans manejados del submódulo Experticias Criminalísticas con la responsabilidad de modificar una experticia.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	
Nombre:	cancelar(ActionEvent e)
Descripción:	Funcionalidad encargada de cancelar las acciones y regresar a la página que invocó el escenario actual del caso de uso.
Nombre:	guardarEnviar()
Descripción:	Funcionalidad encargada de incluir un elemento en el sistema. Si la opción seleccionada por el usuario es la de <i>Guardar</i> , el elemento toma estado <i>En Curso</i> dando la posibilidad de ser modificado posteriormente, no así si es seleccionada la opción <i>Incluir</i> que coloca el elemento en estado <i>Emitido</i> .
Nombre:	isInicializarIncluidos()
Descripción:	Funcionalidad encargada de inicializar las subvistas incluidas en el bean manejado.
Nombre:	limpiarCampos(ActionEvent e)
Descripción:	Funcionalidad encargada de retornar al estado inicial los campos editables de la página.

Nombre:	personalizarComunicacion(ActionEvent e)
Descripción:	Funcionalidad encargada de navegar hacia el caso de uso extendido Configurar Encabezado
Nombre:	vistaPrevia()
Descripción:	Funcionalidad encargada de navegar hacia la UI Vista Previa.
Nombre:	reset()
Descripción:	Funcionalidad auxiliar encargada de eliminar todas a las referencias utilizadas en la realización de la experticia, dejando listas las instancias para la creación de un nuevo elemento.

Nombre: IVistaPreviaExperticia	
Descripción: Clase que representa la interfaz a implementar por todos los beans manejados del submódulo Experticias Criminalísticas con la responsabilidad de mostrar los datos de la vista previa en una experticia.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	
Nombre:	cerrar(ActionEvent actionEvent)
Descripción:	Funcionalidad encargada de regresar a la página que invocó el escenario actual del caso de uso.
Nombre:	aceptar(ActionEvent e)
Descripción:	Funcionalidad encargada de ejecutar los cambios realizados en el informe.
Nombre:	isInicializarIncluidos()
Descripción:	Funcionalidad encargada de inicializar las subvistas incluidas en el bean manejado.

Para el desarrollo del software, se modelaron clases cuyas funcionalidades responden a las necesidades de cada uno de los casos de uso del módulo Experticias Criminalísticas. Seguidamente se describen las clases relevantes concebidas para dar solución al proceso Trayectoria Intraorgánica.

CU Gestionar Informe Pericial de Trayectoria Intraorgánica.

Nombre: InformePericialTrayectoriaIntraorganicaDaoImpl	
Descripción: Clase encargada de implementar la interfaz que presenta métodos de igual firma y tiene el mismo nombre sin el sufijo Impl. Posee las funcionalidades necesarias para realizar las acciones de acceso a dato de la entidad InformePericialTrayectoriaIntraorganica.	
Atributo	Tipo
<i>Esta clase no presenta atributos.</i>	
Responsabilidades:	

Nombre:	obtenerInformeIntraorganicaSegunActaProcesal(String actaProcesal, String actaProcesalExterna)
Descripción:	Funcionalidad encargada de obtener los informes periciales de trayectoria introaorgánica asociados a un acta procesal ó un acta procesal externa.

Nombre: InformePericialTrayectoriaIntraorganicaServiceImpl	
Descripción: Clase encargada de implementar la interfaz que presenta métodos de igual firma y tiene el mismo nombre sin el sufijo Impl. Posee las funcionalidades necesarias para realizar la lógica de negocio correspondiente a la entidad InformePericialTrayectoriaIntraorganica.	
Atributo	Tipo
informePericialTrayectoriaIntraorganicaD ao	InformePericialTrayectoriaIntraorganicaDa o
Responsabilidades:	
Nombre:	obtenerInformeIntraorganicaSegunActaProcesal(String actaProcesal, String actaProcesalExterna)
Descripción:	Funcionalidad encargada de obtener los informes periciales de trayectoria introaorgánica asociados a un acta procesal ó un acta procesal externa.
Nombre:	obtenerInformePericialTrayectoriaIntraorganica(Integer id)
Descripción:	Funcionalidad encargada de obtener un informe pericial de trayectoria introaorgánica dado su identificador.

Nombre: InformeTrayectoriaIntraorganicaVerManejado	
Descripción: Clase que representa el bean de respaldo a la página encargada de mostrar una interfaz que permita al usuario ver los datos del Informe de Trayectoria Intraorgánica.	
Atributo	Tipo
Informe	InformePericialTrayectoriaIntraorganica
expedienteTanatologico	ExpedienteTanatologico
idInformeVer	int
articulosAsociados	List<Articulo>
nombreFuncCreador	String
credencialFuncCreador	String
Asociado	boolean
notificarCambiosEntidad	NotificarCambiosEntidad
investigacionCriminalisticaExperticiasFacad e	InvestigacionCriminalisticaExperticiasFac ade
montajeFotograficoVerManejado	MontajeFotograficoVerManejado
informeTrayectoriaIntraorganicaModificarMa nejado	InformeTrayectoriaIntraorganicaModificar Manejado
informeCriminalisticaEliminarManejado	InformeCriminalisticaEliminarManejado
Responsabilidades:	
Nombre:	cerrar(ActionEvent e)
Descripción:	Funcionalidad encargada de regresar a la página que invocó el escenario actual del caso de uso.

Nombre:	eliminar()
Descripción:	Funcionalidad encargada de navegar hacia la UI Eliminar Informe.
Nombre:	getParametrosReporte()
Descripción:	Funcionalidad encargada de obtener los recursos necesarios para mostrar el reporte correspondiente al informe.
Nombre:	getDatos()
Descripción:	Funcionalidad encargada de concatenar los elementos necesarios para ser mostrados en forma de texto en la página.
Nombre:	getDespachoFuncionario()
Descripción:	Funcionalidad encargada de obtener el despacho al que pertenece el funcionario responsable del informe.
Nombre:	modificar()
Descripción:	Funcionalidad encargada de navegar hacia la UI Modificar Informe.

Nombre: InformeTrayectoriaIntraorganicaIncluirManejado	
Descripción: Clase que representa el bean de respaldo a la página encargada de mostrar una interfaz que permita al usuario incluir un nuevo Informe de Trayectoria Intraorgánica.	
Atributo	Tipo
Informe	InformePericialTrayectoriaIntraorganica
solicitudAsociada	SolicitudExperticiaTrayectoriaIntraorganica
expedienteTanatologico	ExpedienteTanatologico
Trayectos	List<SelectItem>
listaArticulos	List<Articulo>
investigacionCriminalisticaExperticiasFacade	InvestigacionCriminalisticaExperticiasFacade
montajeFotograficoSubviewManejado	MontajeFotograficoSubviewManejado
validarActaProcesalSubviewManejado	ValidarActaProcesalSubviewManejado
informeTrayectoriaIntraorganicaVistaPreviaManejado	InformeTrayectoriaIntraorganicaVistaPreviaManejado
Responsabilidades:	
Nombre:	guardarEnviar()
Descripción:	Funcionalidad encargada de incluir un informe en el sistema. Si la opción seleccionada por el usuario es la de <i>Guardar</i> , el elemento toma estado <i>En Curso</i> dando la posibilidad de ser modificado posteriormente, no así si es seleccionada la opción <i>Incluir</i> que coloca el elemento en estado <i>Emitido</i> .
Nombre:	vistaPrevia()
Descripción:	Funcionalidad encargada de navegar hacia la UI Vista Previa.
Nombre:	limpiarCampos(ActionEvent e)
Descripción:	Funcionalidad encargada de retornar al estado inicial los campos editables de la página.
Nombre:	cancelar(ActionEvent e)
Descripción:	Funcionalidad encargada de regresar a la página que invocó el escenario actual del caso de uso.
Nombre:	reset()
Descripción:	Funcionalidad auxiliar encargada de eliminar todas a las referencias utilizadas en la realización del informe, dejando listas las instancias para la creación de uno nuevo.
Nombre:	personalizarComunicacion(ActionEvent e)
Descripción:	Funcionalidad encargada de navegar hacia el caso de uso extendido Configurar Encabezado
Nombre:	isInicializarIncluidos()
Descripción:	Funcionalidad encargada de inicializar las subvistas incluidas en el escenario actual.
Nombre:	verExpediente(ActionEvent e)
Descripción:	Funcionalidad encargada de navegar hacia la interfaz que muestra los datos del expediente tanatológico asociado al informe.

2.2.3 Realización de los Casos de Uso

Los contratos describen el comportamiento de un sistema, lo que hace, sin explicar cómo lo hace. Los diagramas de contrato constituyen un artefacto generado, a partir de necesidades propias del proyecto para ilustrar de forma sencilla la interacción entre paquetes.

CU Gestionar Trayectoria Intraorgánica.

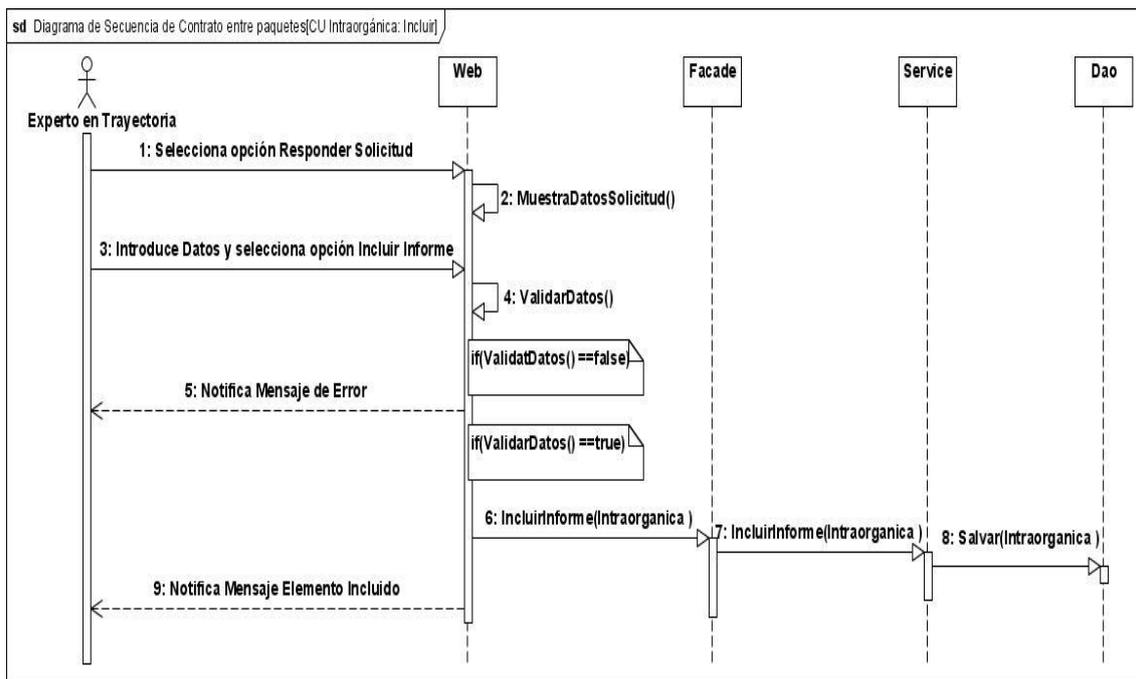


Ilustración 17: Diagrama de Contrato entre Paquetes CU Intraorgánica: Incluir.

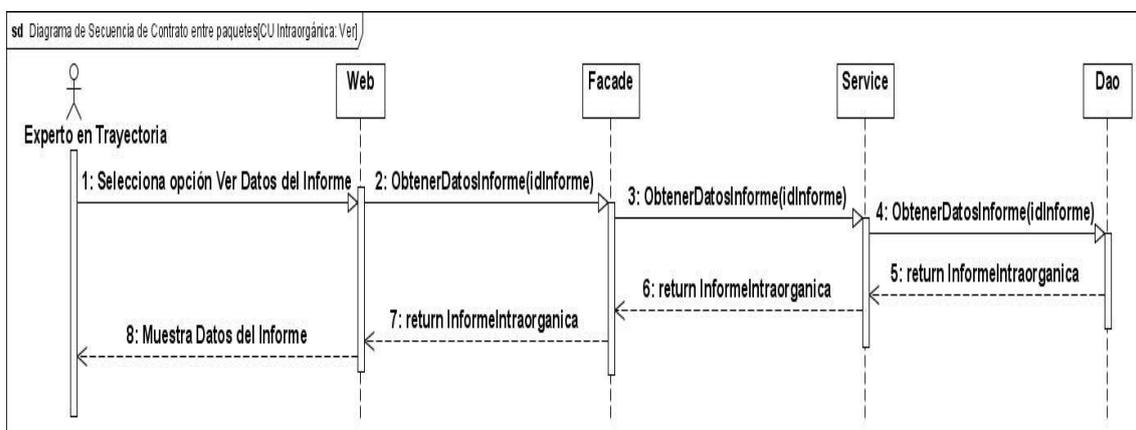


Ilustración 18: Diagrama de Contrato entre Paquetes CU Intraorgánica: Ver.

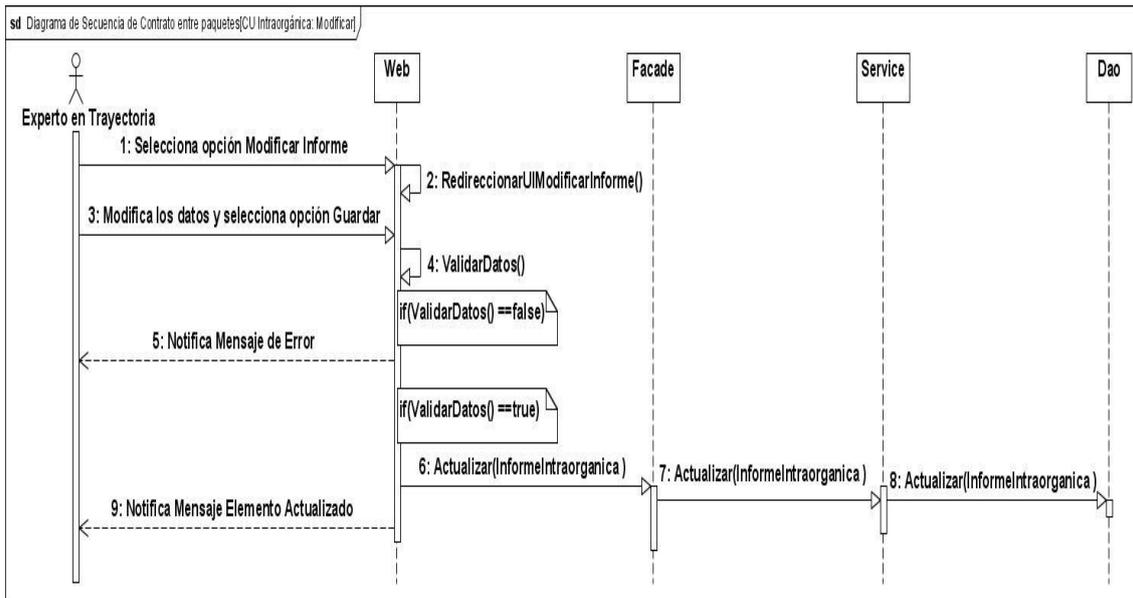


Ilustración 19: Diagrama de Contrato entre Paquetes CU Intraorgánica: Modificar.

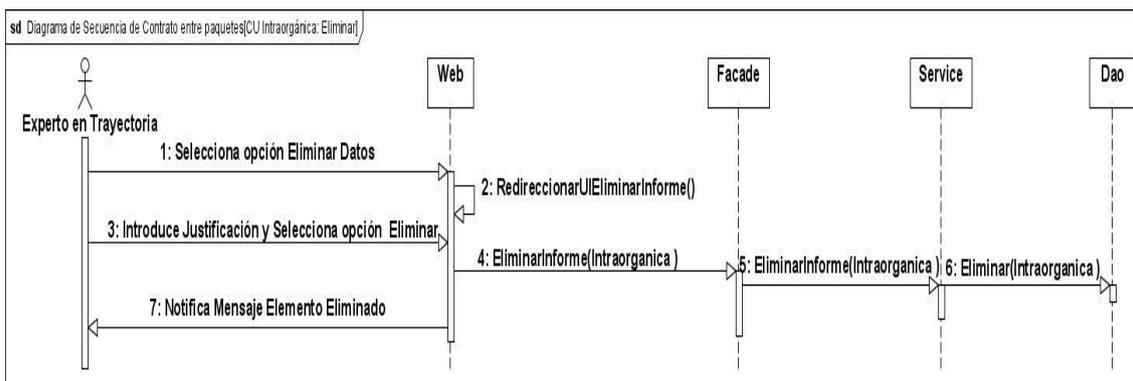


Ilustración 20: Diagrama de Contrato entre Paquetes CU Intraorgánica: Eliminar

CU Gestionar Trayectoria Balística.

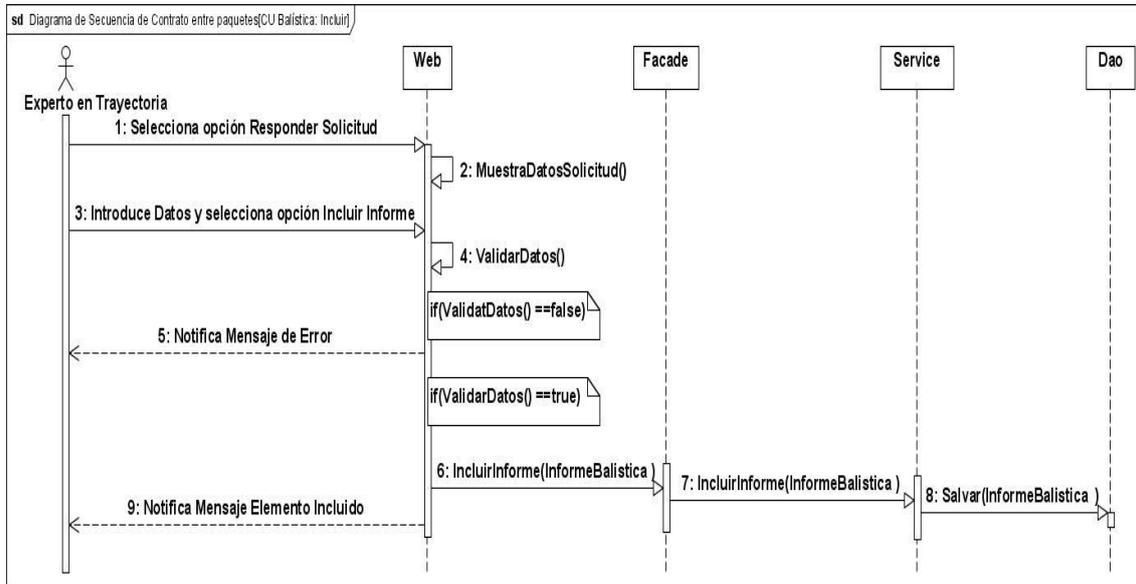


Ilustración 21: Diagrama de Contrato entre Paquetes CU Balística: Incluir.

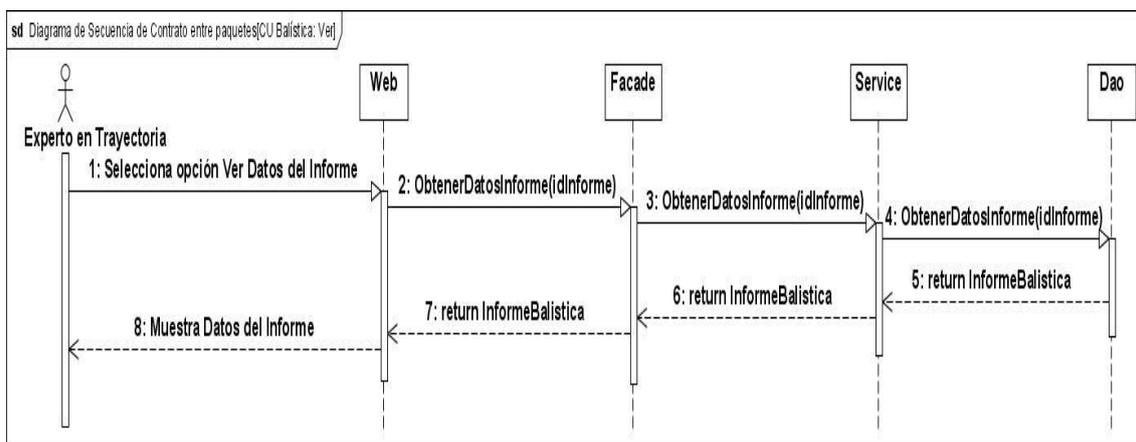


Ilustración 22: Diagrama de Contrato entre Paquetes CU Balística: Ver.

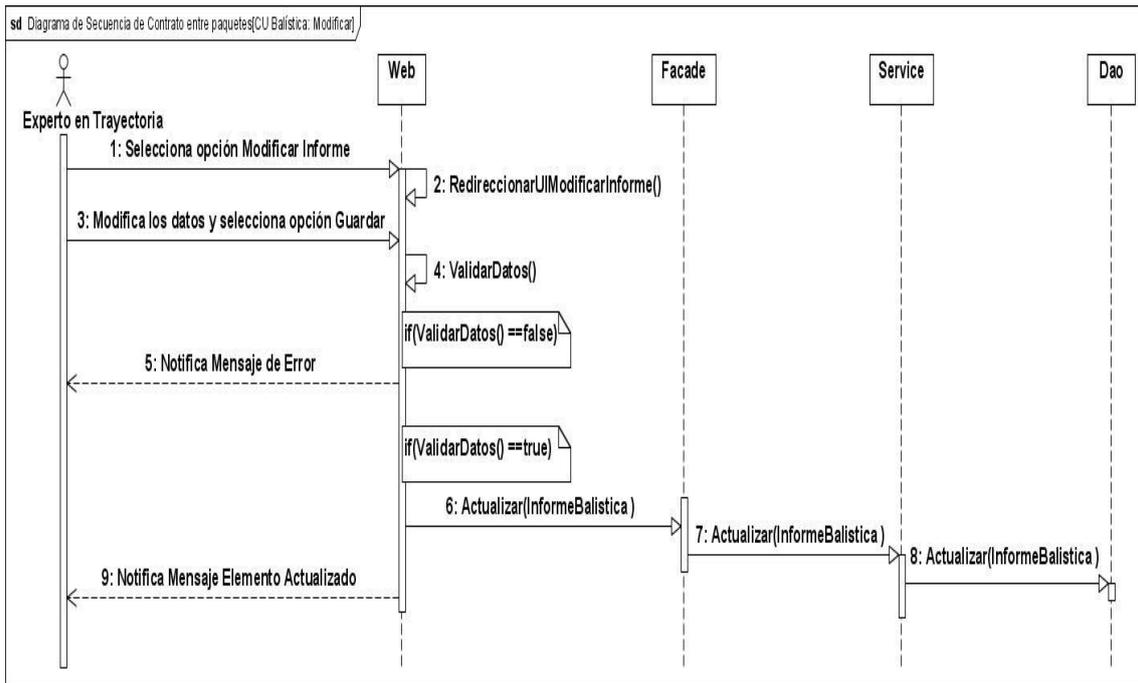


Ilustración 23: Diagrama de Contrato entre Paquetes CU Balística: Modificar.

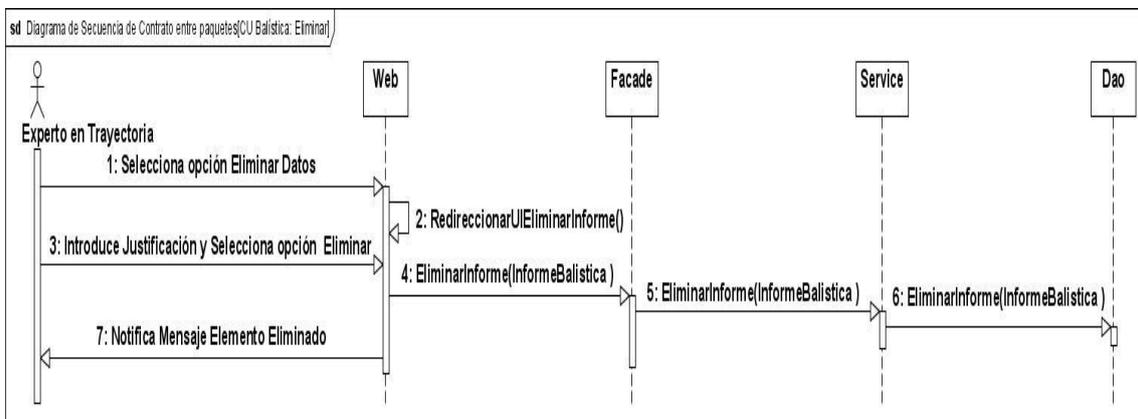


Ilustración 24: Diagrama de Contrato entre Paquetes CU Balística: Eliminar.

CU Gestionar Resultado de Inspección Técnica de Robo a Banco.

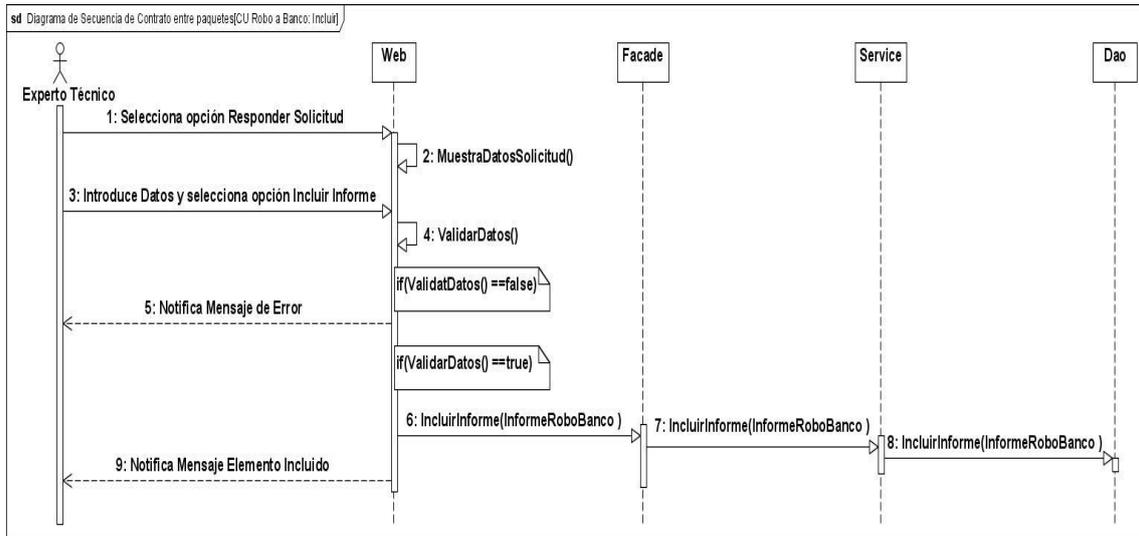


Ilustración 25: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Incluir.

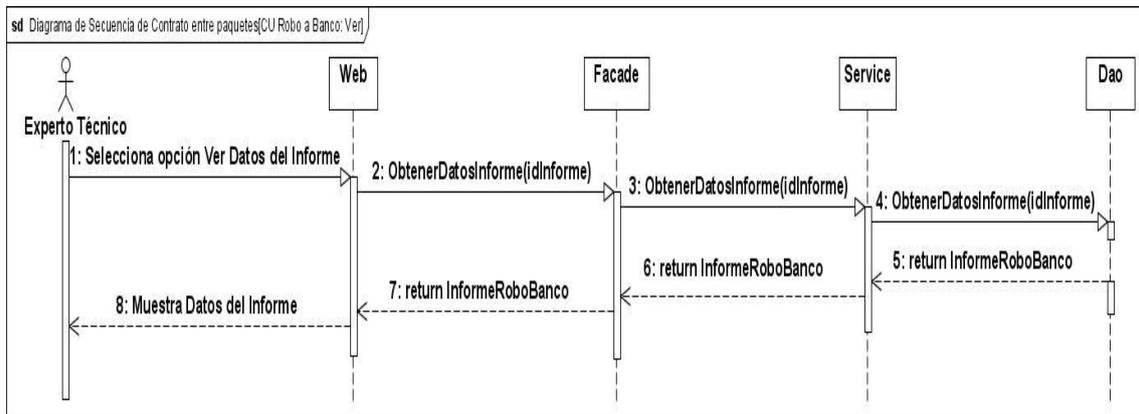


Ilustración 26: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Ver.

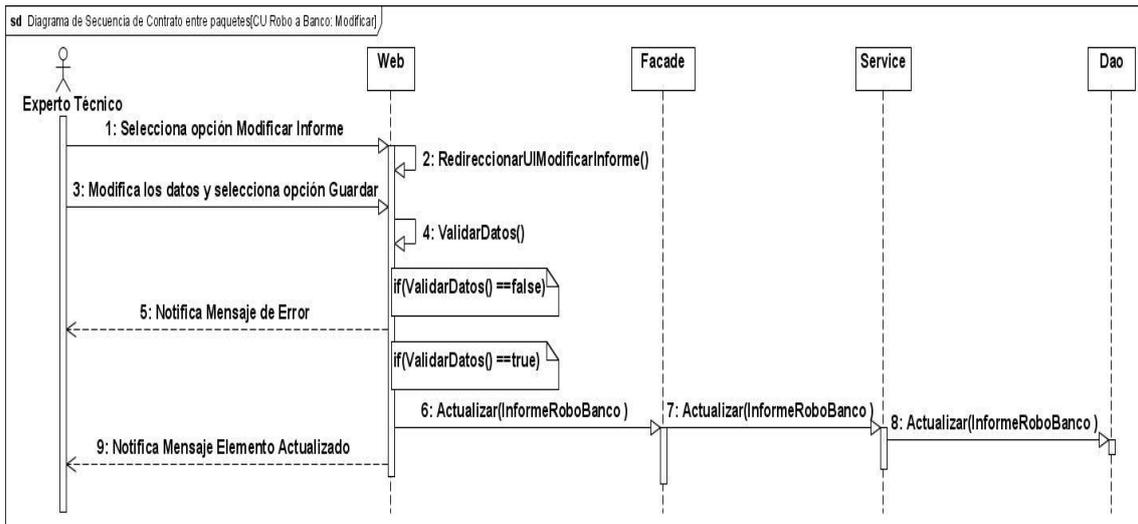


Ilustración 27: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Modificar.

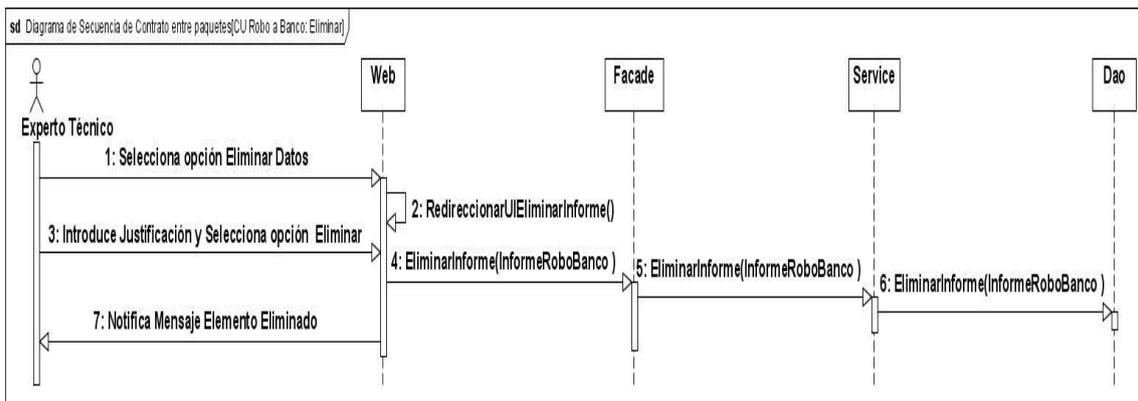


Ilustración 28: Diagrama de Contrato entre Paquetes CU Inspección Robo a Banco: Eliminar.

2.3 Modelo de Datos

Un modelo de datos permite describir las estructuras presentes en la base de datos, el tipo de los datos y la forma en que se relacionan. Posibilita la representación de las restricciones de integridad, que no son más sino un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada. De igual forma permite detallar las operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base. Un modelo de datos es un lenguaje orientado a describir una Base de Datos.

2.3.1 Diagrama de Clases Persistentes

El diagrama de clases persistentes ilustra de forma gráfica la organización estructural de las entidades que almacenan los datos asociados al módulo Experticias Criminalísticas.

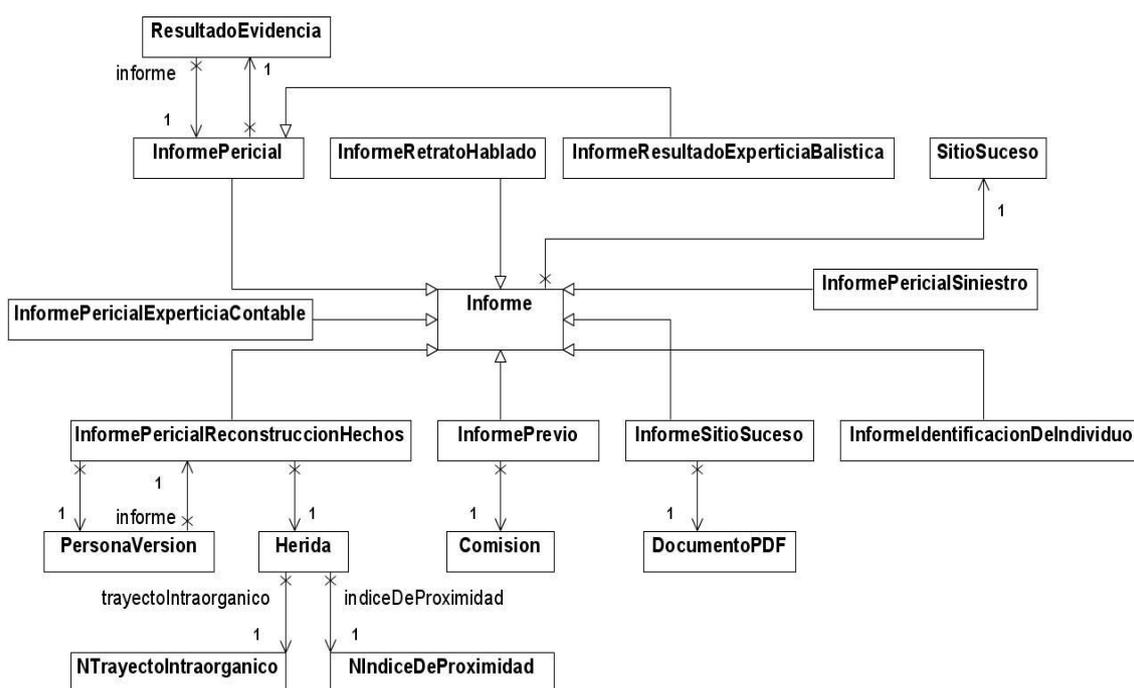


Ilustración 29: Diagrama de clases persistentes. Vista I.

2.4 Modelo de Implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

Este artefacto describe cómo se implementan los componentes, congregándolos en subsistemas organizados en capas y jerarquías, y señala las dependencias entre éstos.

2.4.1 Diagramas de subsistemas de implementación

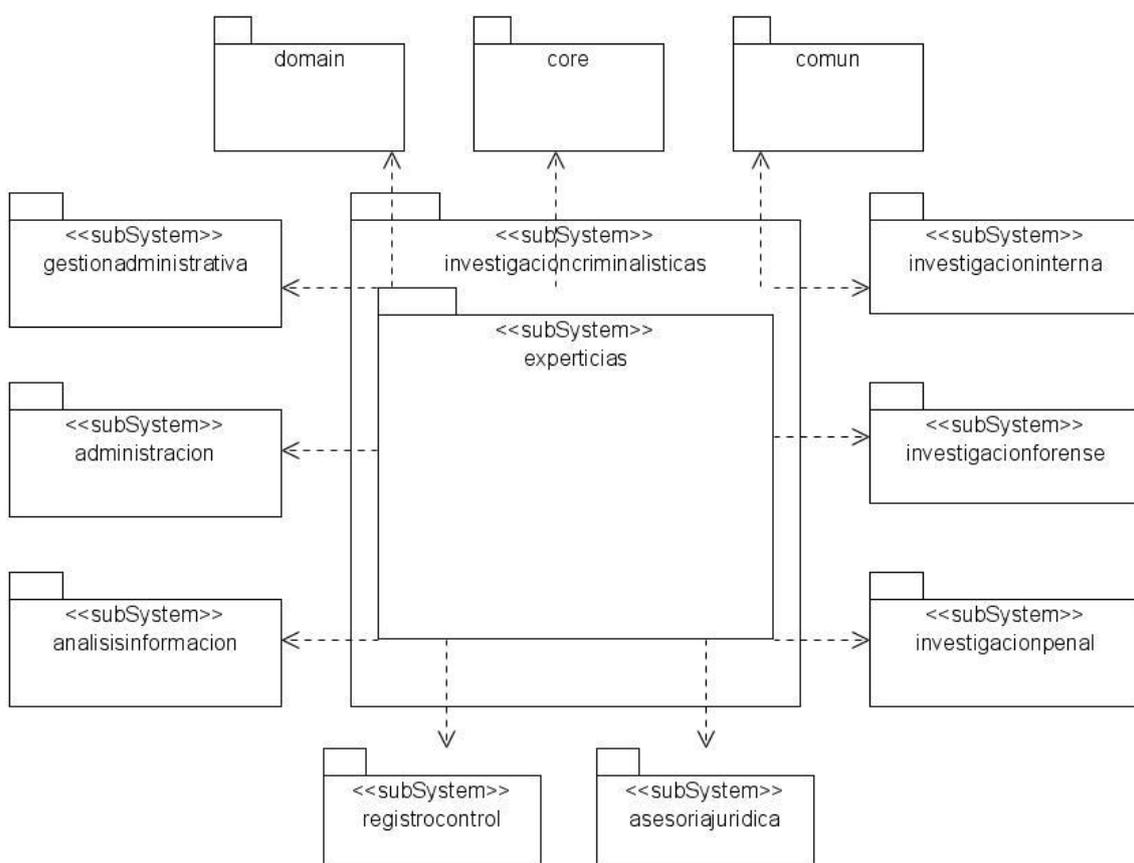


Ilustración 32: Diagrama de subsistemas de Implementación.

2.4.2 Diagrama de componentes

Los Diagramas de Componentes ilustran las piezas del software que conformarán un sistema. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema.

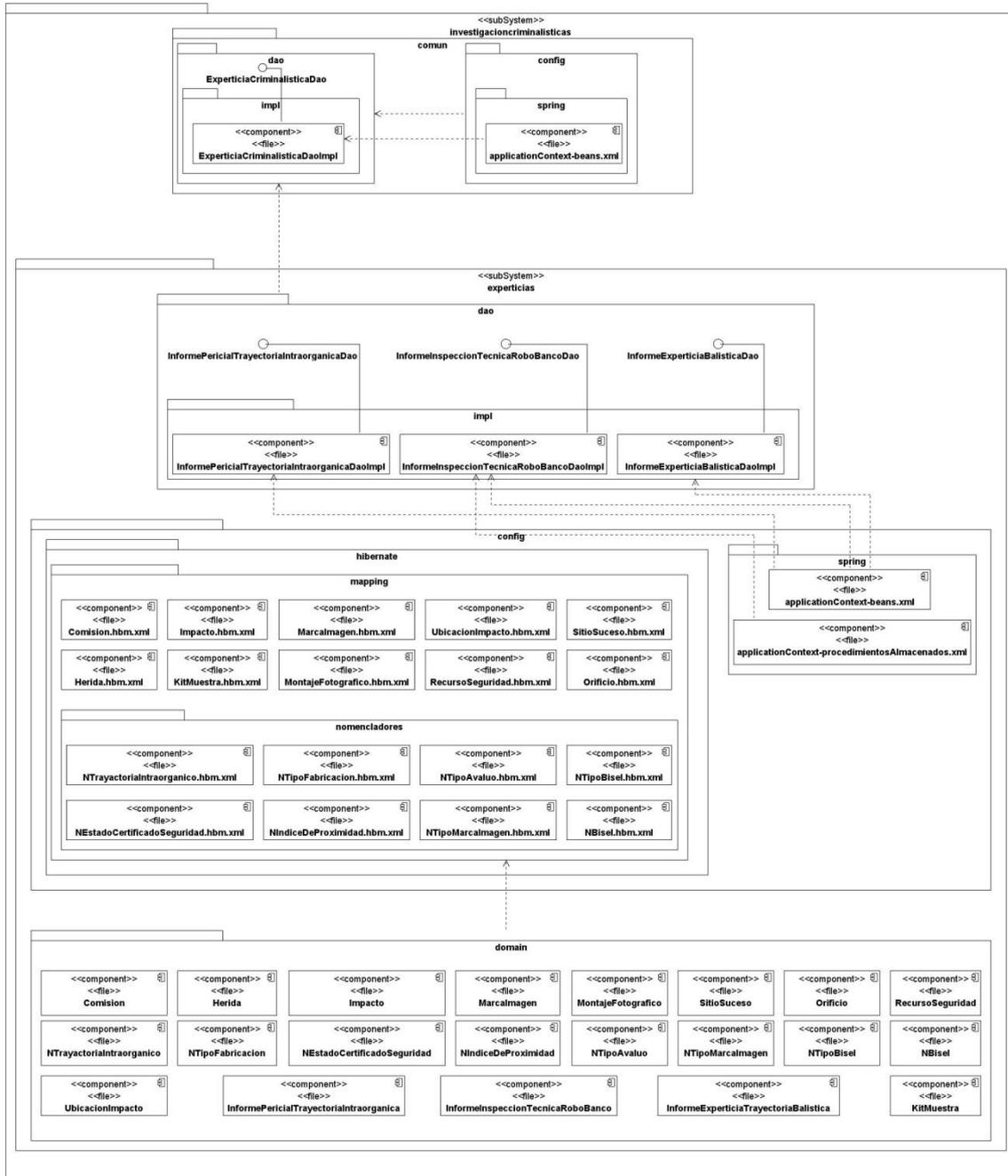


Ilustración 33: Diagrama de Componentes. Vista Acceso a Datos.

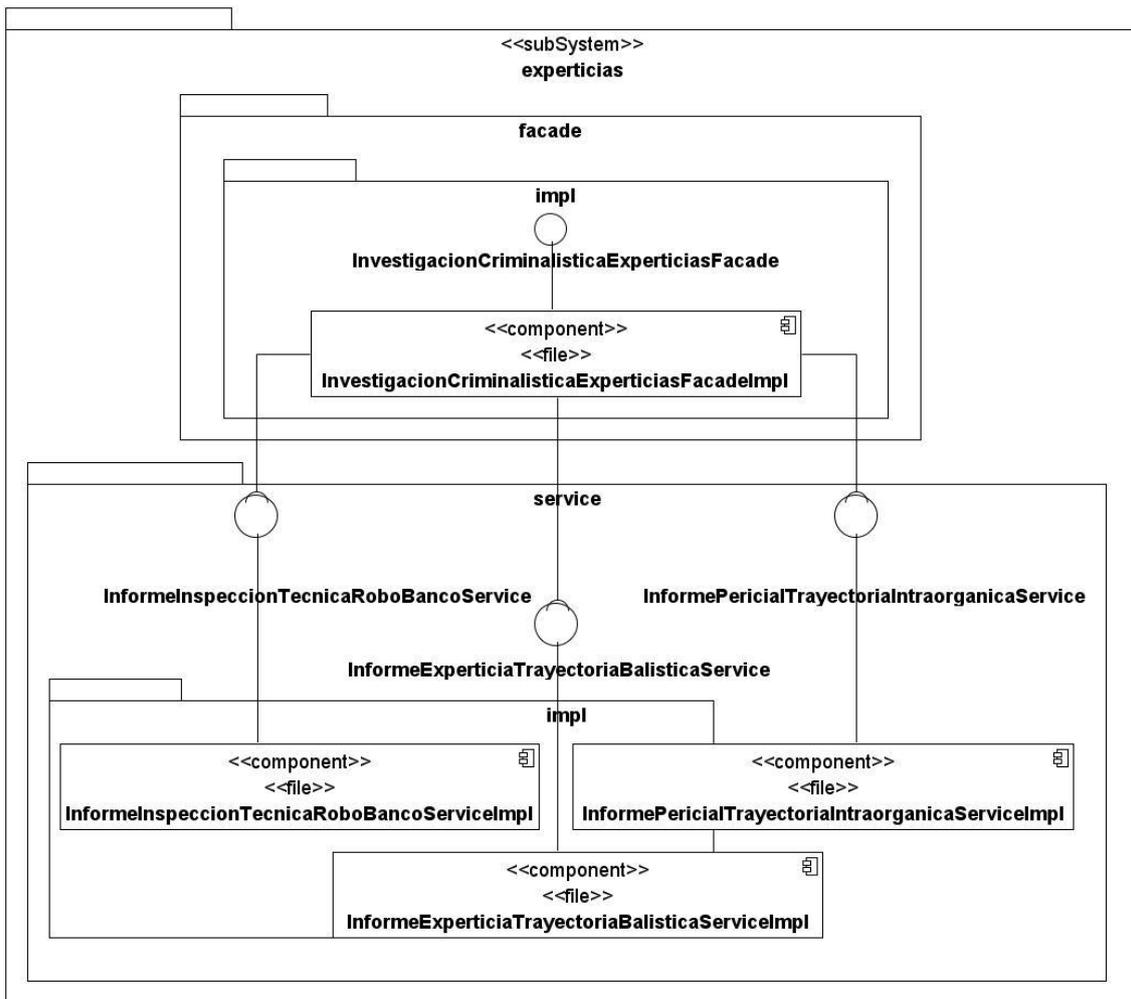


Ilustración 34: Diagrama de Componentes. Vista Lógica de Negocio.

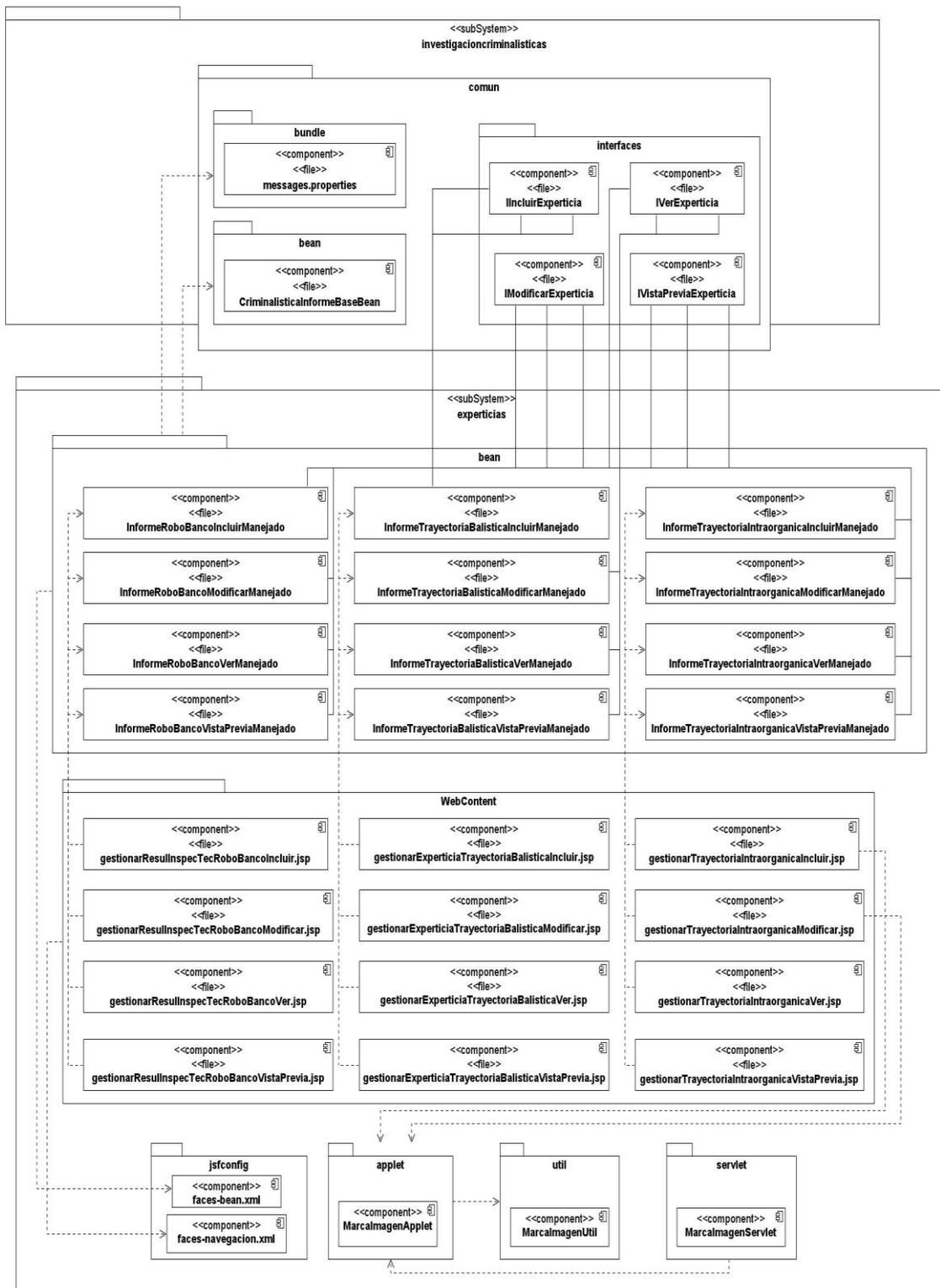


Ilustración 35: Diagrama de Componentes. Vista Presentación.

2.4.3 Estándar de codificación

La convención de código utilizada para el desarrollo de la solución propuesta constituye una decisión centralizada del equipo de Arquitectura. Para ello se elaboró un documento llamado Guía de Estilo de Código para el proyecto CICPC que resume las características fundamentales del estilo de código puesto en práctica, siguiendo las Convenciones de Código Java (Java Code Conventions), pues el estilo de programación es dependiente del lenguaje con el que se quiere desarrollar el sistema.

2.4.4 Interfaces de Usuario

La interface de usuario (IU) es la parte del programa que permite a éste interactuar con el usuario. Las interfaces de usuario pueden adoptar muchas formas, que van desde la simple línea de comandos hasta las interfaces gráficas que proporcionan las aplicaciones más modernas.

La interface de usuario es el aspecto más importante de cualquier aplicación ya que esta sin una interfaz fácil, impide que los usuarios saquen el máximo rendimiento del programa. Por tal motivo se diseñó, por parte de un equipo especializado, para el Módulo Experticias Criminalísticas y el SIIPOL en general, un conjunto de interfaces de usuario de fácil manejo y aspecto amigable.

Las Ilustraciones de la 24 a la 29 muestran las IU correspondientes a los casos de uso Gestionar Informe Pericial de Trayectoria Intraorgánica y Gestionar Informe Pericial de Trayectoria Balística, seleccionadas específicamente por los numerosos componentes que contienen que ilustran de manera general los existentes en el resto de la IU.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): CG Emiliano Sánchez Gallardo 26/03/2009 ... 01:16 Inicio ? Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Investigación Interna
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
 - Solicitudes
 - Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica

INFORME PERICIAL DE TRAYECTORIA INTRAORGÁNICA

Informe **Montaje Fotográfico**

Datos Generales

No.Expediente	K-09-0001-0003
Dependencia Solicitante	Fiscalía General
Representante	Jose Garcia

Descripción de la Peritación

Trayecto Intraorgánico
Ligeramente ascendente

Un policía murió por bala este lunes cuando patrullaba en Irlanda del Norte, dos días después del asesinato de dos soldados británicos, anunciaron responsables políticos norirlandeses a la AFP.

Un policía murió por bala este lunes cuando patrullaba en Irlanda del Norte, dos días después del asesinato de dos soldados británicos, anunciaron responsables políticos norirlandeses a la AFP.

Base Legal

Ley Artículo

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPP	Artículo 1	Eliminar
Total: 1		

Ilustración 36: IU Incluir Informe Pericial de Trayectoria Intraorgánica.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): CG Emiliano Sánchez Gallardo 26/03/2009 01:16 Inicio Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Investigación Interna
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
 - Solicitudes
 - Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica

INFORME PERICIAL DE TRAYECTORIA INTRAORGÁNICA

Informe **Montaje Fotográfico**

Imágenes Asociadas

Nombre Imagen	Editar Imagen	Disociar
Víctima I	Editar	Disociar
Total: 1		

Vista Previa



[Ampliar](#)

Asociar imagen

 [Ampliar](#)

Nombre:

Ubicación: [Examinar...](#)

Descripción:

Fecha de Captura:

[Incluir Imagen](#)

[Vista Previa](#) [Incluir](#) [Guardar](#) [Limpiar](#) [Cancelar](#)

Ilustración 37: IU Incluir Informe Pericial de Trayectoria Intraorgánica. Montaje Fotográfico.

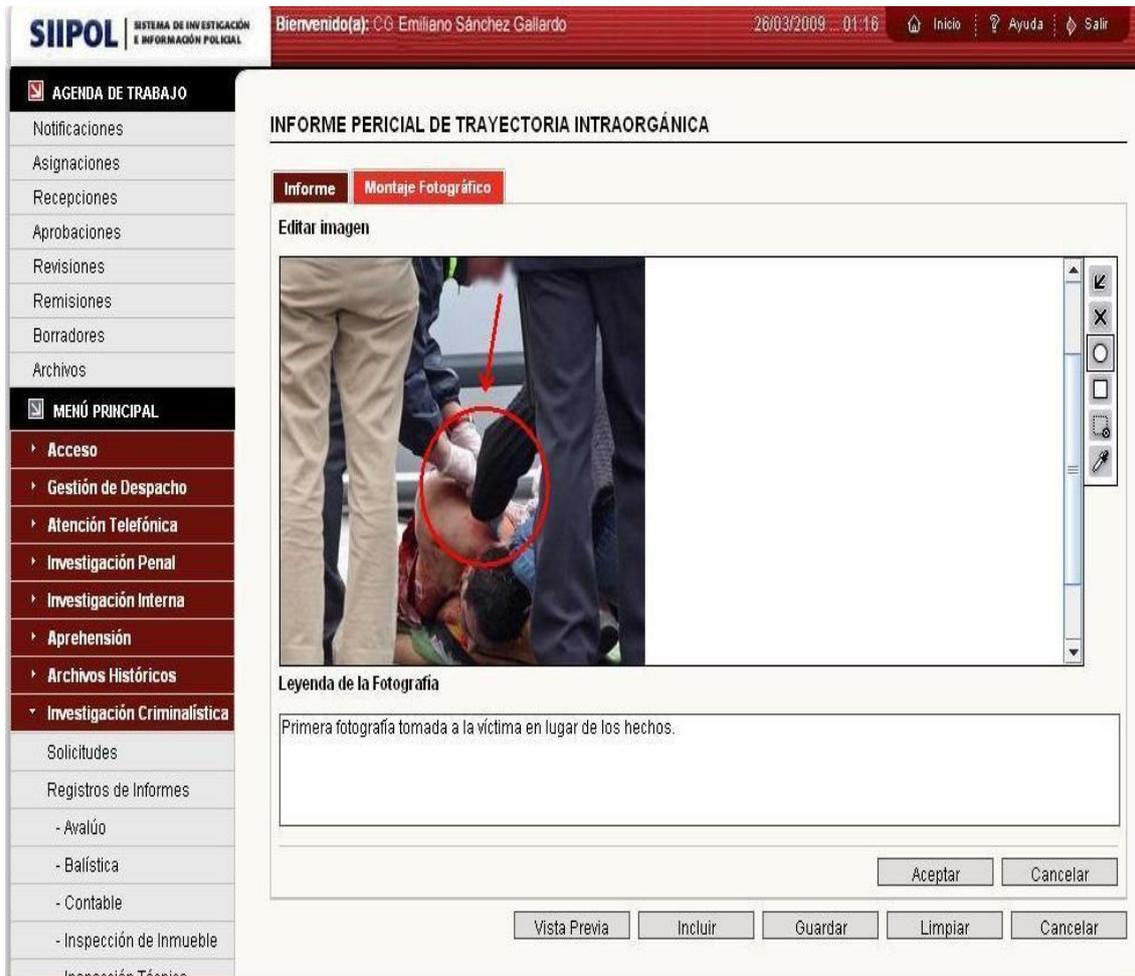


Ilustración 38: IU Incluir Informe Pericial de Trayectoria Intraorgánica. Editar Imágenes.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): CG Emiliano Sánchez Gallardo 26/03/2009 ... 01:16 Inicio ? Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Investigación Interna
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
 - Solicitudes
 - Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica
 - Levantamiento Planimét...
 - Microscopía Electrónica
 - Pericial Criminalístico
 - Reconstrucción de Hech...

INCLUIR INFORME PERICIAL DE TRAYECTORIA BALÍSTICA

Informe **Imágenes**

Datos Generales

No. Expediente	K-09-0005-01234
Dependencia Solicitante	Fiscalía General
Representante	Jose Garcia

Informe Pericial de Trayectoria Balística

Elementos Técnico-Criminalísticos | Elementos Médico-Legal | Conclusiones

Sitios de Sucesos | Seleccionar Elementos | Elementos Relacionados | Ver Orificios y / o Impactos Asociados

Nombre: Esquina Norte

Descripción: Punto de encuentro de malandros y criminales.

Sitios de Sucesos

Nombre	Eliminar
No existen registros	

Incluir

Base Legal

Ley: [dropdown] Artículo: [dropdown] Adicionar

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPP	Artículo 1	Eliminar
Total: 1		

Vista Previa Incluir Guardar Limpiar Cancelar

Ilustración 39: IU Incluir Informe Pericial de Trayectoria Balística.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

Bienvenido(a): CG Emiliano Sánchez Gallardo 28/03/2009 01:16 Inicio Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Investigación Interna
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
 - Solicitudes
 - Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica
 - Levantamiento Planimét...
 - Microscopía Electrónica
 - Pericial Criminalístico
 - Reconstrucción de Hech...
 - Siniestro
 - Trayectoria Balística
 - Trayectoria Intraorranónica

INCLUIR INFORME PERICIAL DE TRAYECTORIA BALÍSTICA

Informe **Imágenes**

Datos Generales

No. Expediente	K-09-0005-01234
Dependencia Solicitante	Fiscalía General
Representante	Jose Garcia

Informe Pericial de Trayectoria Balística

Elementos Técnico-Criminalísticos | Elementos Médico-Legal | Conclusiones

Sitios de Sucesos | Seleccionar Elementos | Elementos Relacionados | **Ver Orificios y / o Impactos Asociados**

Sitio: Esquina Norte

Descripcion: Punto de encuentro de malandros y criminales.

Impacto	Dimensión	Descripción	Bisel en Proyección	Eliminar
1	3x5	Impactos detectados en el lugar de los hechos	Adentro hacia Afuera, Derecha a Izquierda, Abajo hacia Arriba	Eliminar

Orificio	Dimensión	Descripción	Bisel en Proyección
1	2x5	Orificio encontrado en la pared	Adentro hacia Afuera, Derecha a Izquierda, Abajo hacia Arriba

Base Legal

Ley: [dropdown] Artículo: [dropdown] Adicionar

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPP	Artículo 1	Eliminar
Total: 1		

Vista Previa Incluir Guardar Limpiar Cancelar

Ilustración 40: Incluir Informe Pericial de Trayectoria Balística. Orificios e/o Impactos.

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL **Bienvenido(a):** CG Administrator Administrator Administrator. 26/03/2009 ... 01:32 Inicio ? Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Investigación Interna
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
 - Solicitudes
 - Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica
 - Levantamiento Planimét...
 - Microscopía Electrónica

INCLUIR INFORME PERICIAL DE TRAYECTORIA BALÍSTICA

Informe **Imágenes**

Imágenes Asociadas

Nombre Imagen	Descripción	Disociar
Arma	Arma utilizada	Disociar
Total: 1		

Vista Previa



[Ampliar](#)

Asociar Imagen

 [Ampliar](#)

Nombre:

Ubicación: [Examinar...](#)

Descripción:

Fecha de Captura: 

[Incluir Imagen](#)

[Vista Previa](#) [Incluir](#) [Guardar](#) [Limpiar](#) [Cancelar](#)

Ilustración 41: IU Incluir Informe Pericial de Trayectoria Balística. Montaje Fotográfico.

2.5 Conclusiones

Con el desarrollo de la propuesta de solución se garantizó la automatización de un elevado por ciento de los procesos de la Coordinación Nacional de Criminalística referente a las experticias.

La utilización de los patrones de diseño seleccionados para el desarrollo de la solución propuesta, formalizó un vocabulario común entre el quipo de desarrollo y proporcionó elementos reusables en el diseño del sistema de software.

A través de la realización del modelo de diseño utilizando las tarjetas CRC se logró agilizar el proceso de desarrollo, generando solo la documentación necesaria, que sirviera de guía para la construcción del software.

La representación del modelo de implementación, diagrama de componentes y diagrama de subsistemas de implementación, así como el de clases persistentes, brindaron grandes facilidades al equipo en el momento de desarrollar la solución propuesta.

Capítulo 3 Validación de la Solución Propuesta

3.1 Introducción

Los proyectos de desarrollo de software han padecido tradicionalmente problemas de calidad, tanto en el propio proceso de desarrollo como en los productos que entregan. Esta problemática tiene su origen en las habituales **desviaciones de plazos y esfuerzo** sobre los valores previstos y en la frecuente **aparición de fallos** durante la implantación y operación de los productos resultantes.

El primer problema pone de manifiesto una falta de calidad en el proceso de gestión de los proyectos software: cuanto menor es ésta, peor es el grado de adherencia a los plazos y esfuerzos previstos. El segundo problema indica falta de calidad de los productos desarrollados: cuanto menor es ésta, mayor es el número de defectos y, consecuentemente, mayor será el número de fallos que aparecerán durante la ejecución del software. La calidad del producto es lógicamente muy dependiente de la calidad del proceso, de esta manera si se lleva a cabo un proceso defectuoso, lo más probable es que se obtenga un producto insuficiente.

La validación de software es de vital importancia ya que proporciona un alto grado de confianza y seguridad no solo en el producto sino en los resultados que se obtienen en implantarlo. Validar el software que se produce es un requisito que debe ser cumplido con rigor.

Para lograr una correcta validación del software se hace necesario una especificación documentada de los requisitos para que sea posible chequear su cumplimiento. Se requiere además tiempo y esfuerzo y la conclusión de que el sistema está validado debe estar respaldada por evidencia recolectada a partir de las actividades planificadas a lo largo del proceso de desarrollo.

La validación de la porción de software desarrollado, correspondiente al módulo de Experticias Criminalísticas estuvo regida por iteraciones de prueba a diferentes niveles, de las cuales se expondrán los resultados obtenidos.

3.2 Tipos de Prueba. Niveles de Prueba.

El módulo Experticias Criminalísticas fue sometido a diversas pruebas en cada una de las iteraciones por las que pasó. Es válido destacar que las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto.

Pruebas de Caja Blanca: se denomina **cajas blancas** a un tipo de prueba de software que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas (integración).

Pruebas de Caja Negra: se denomina **caja negra** a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una *caja negra* nos interesará su forma de interactuar con el medio que le rodea (en ocasiones, otros elementos que también podrían ser *cajas negras*) entendiendo **qué es lo que hace**, pero sin dar importancia a **cómo lo hace**. Por tanto, de una *caja negra* deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

3.2.1 Niveles de Prueba

Pruebas Unitarias: Comienzan con la prueba de cada módulo. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de estos funcione correctamente por separado. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el fragmento de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas, fomentan el cambio, simplifica la integración, documenta el código, separa la interfaz del código y hace que los errores estén más acotados y sean fáciles de localizar.

Pruebas de Integración: A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una

vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes de software funcionan juntos.

Prueba del Sistema. El software ensamblado totalmente con cualquier componente hardware que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de software completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. A este tipo de pruebas donde se estudia el producto completo se les llama Pruebas de Sistema.

Pruebas de Aceptación. El cliente comprueba que el software funciona según sus expectativas.

Para evaluar la solución desarrollada se planificaron varias iteraciones donde se estuvo probando el software exhaustivamente. Se abarcaron los niveles de prueba expuestos anteriormente en cada una de las iteraciones, arrojando estas últimas resultados visibles. Como material base para el inicio de la validación de la solución propuesta, se estructuraron los Casos de Prueba a partir de los cuales se chequeó el cumplimiento de los requisitos funcionales del software.

Las pruebas se dividieron de la siguiente forma:

Pruebas Internas: Pruebas realizadas al sistema por el equipo de calidad interna del proyecto, se realizaron con el fin de entregar un producto lo más libre de errores posible al tercero encargado de validar la factibilidad del sistema. Se centraron en el cumplimiento de las funcionalidades descritas en el listado de requisitos y de casos de uso.

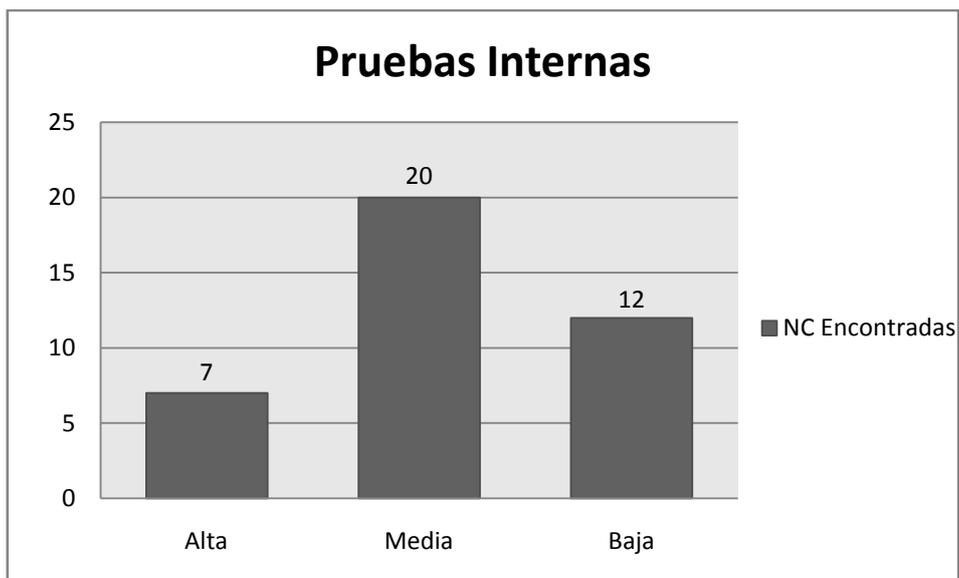


Ilustración 42: NC detectadas durante la Pruebas Internas

Pruebas Cruzadas: Pruebas realizadas al sistema por el resto de los equipos de desarrollo del proyecto. Se realizaron con el fin de encontrar la mayor cantidad de errores posibles en término de validaciones, pautas definidas por arquitectura de información, formato de los campos, entre otras. Estas pruebas, al ser realizadas por los desarrolladores en rol de probadores, tienen un beneficio extra, y es que el conocimiento del funcionamiento interno del software que tienen estas personas les permite aportar valiosa información al desarrollo y detectar errores altamente complejos antes de que se conviertan en un riesgo.

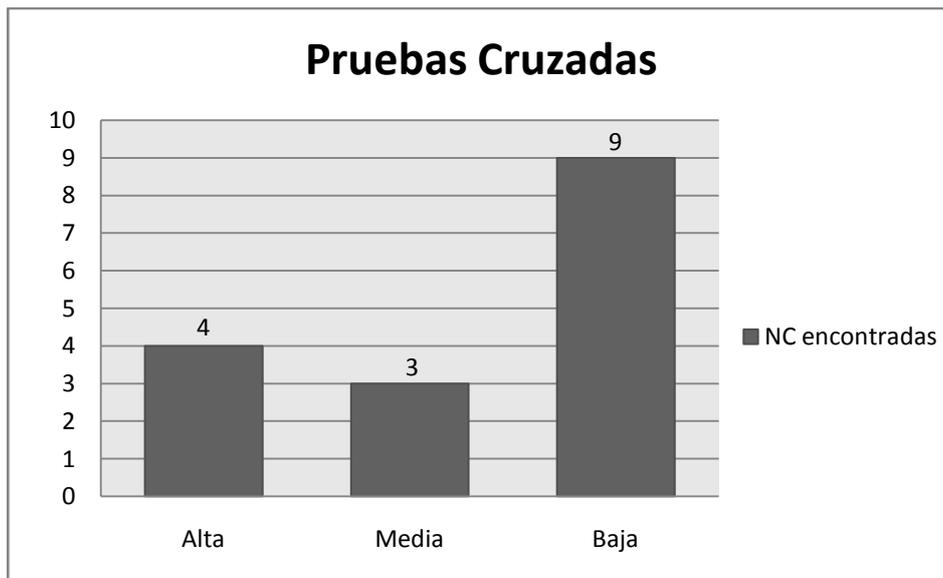


Ilustración 43: NC detectadas durante las Pruebas Cruzadas

Pruebas de Liberación: Pruebas realizadas por un tercero, en este caso Calisoft, institución encargada de validar que el software cuente con la calidad requerida para ser entregado a los clientes finales.

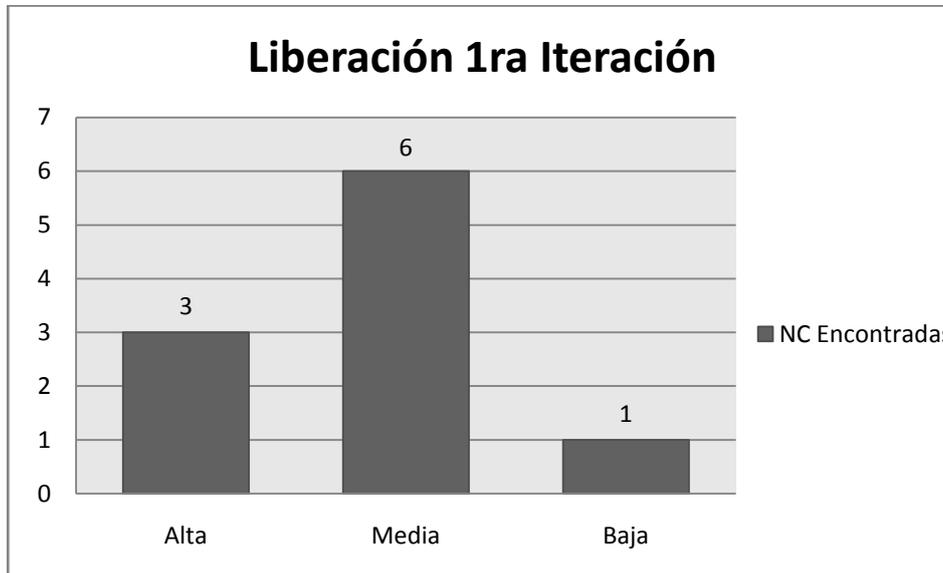


Ilustración 44: NC detectadas durante las Pruebas de Liberación

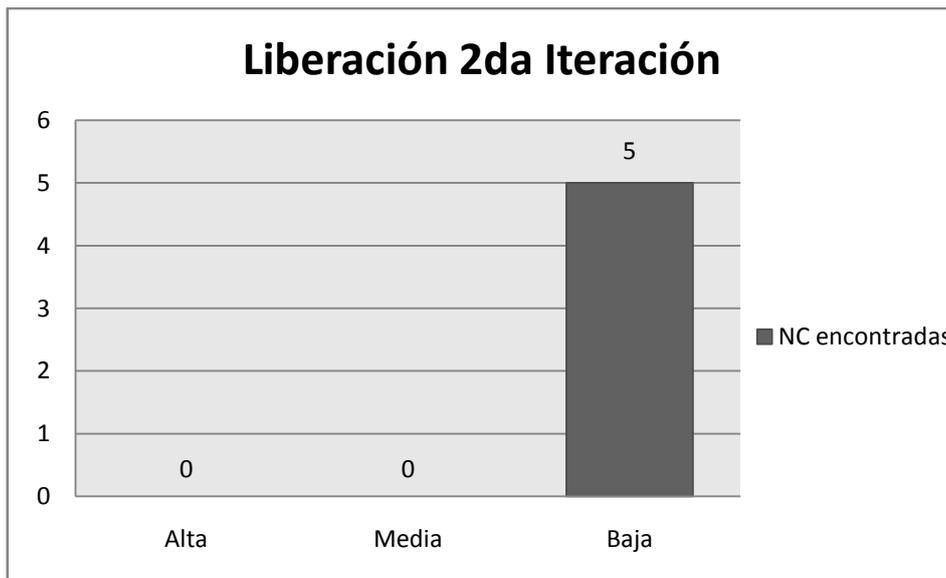


Ilustración 45: NC detectadas durante las Pruebas de Liberación

Pruebas de Aceptación: Pruebas realizadas por los clientes para comprobar que el sistema cumple con sus expectativas desde el punto de vista funcional y no funcional.

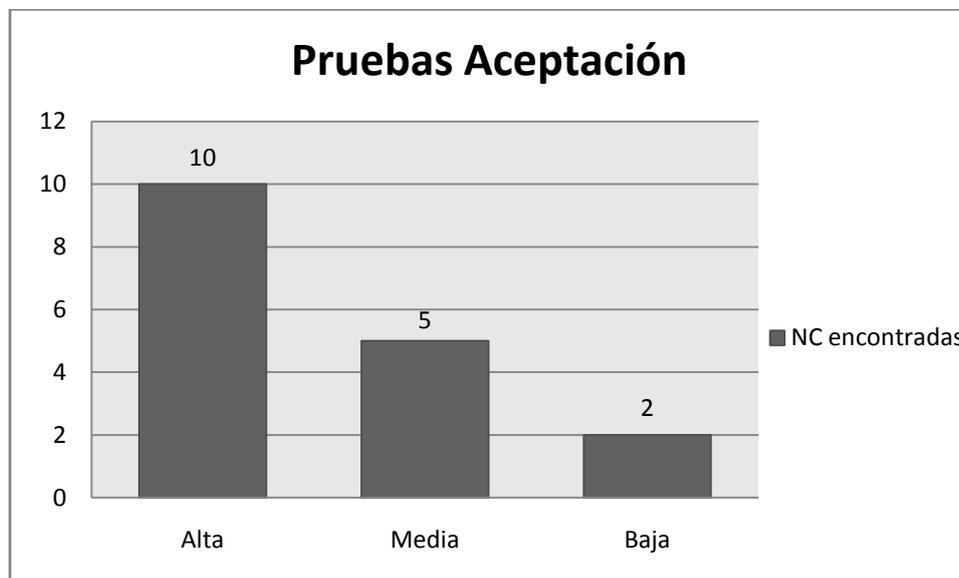


Ilustración 46: NC detectadas durante las Pruebas de Aceptación

Pruebas Piloto: Pruebas realizadas por los clientes para comprobar el rendimiento de la aplicación y su respuesta en un entorno real de trabajo. Así como el cumplimiento de las funcionalidades requeridas.

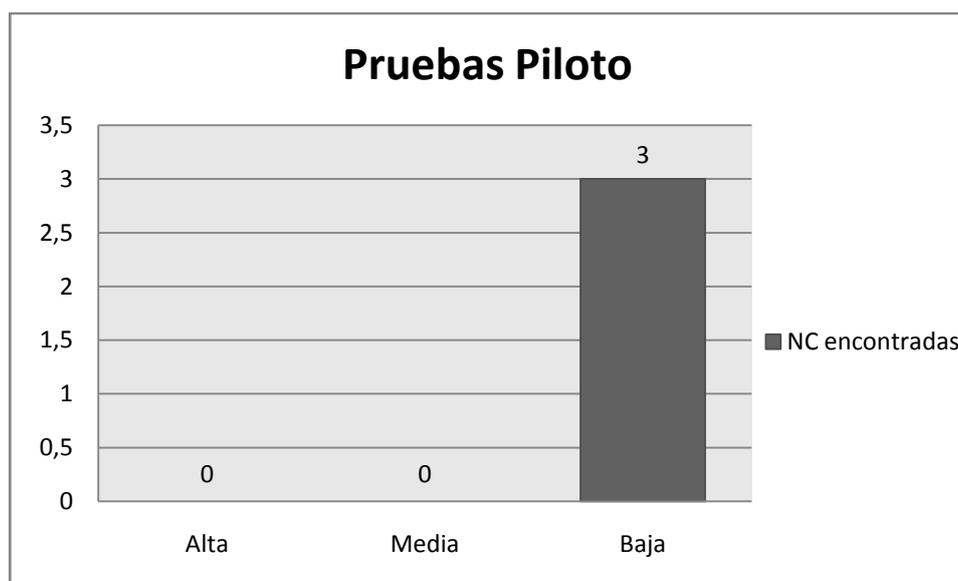


Ilustración 47: NC detectadas durante las Pruebas Piloto

A partir del estudio de estas pruebas se elaboró la siguiente tabla:

NC asociadas al Módulo Experticias Criminalísticas						
	Internas	Cruzadas	Liberación1	Liberación2	Aceptación	Piloto
Alta	7	4	3	0	10	0
Media	20	3	6	0	5	0
Baja	12	9	1	5	2	3
Total	39	16	10	5	17	3

Las pruebas internas concluyeron con un total de treinta y nueve defectos encontrados, estas marcaron el inicio del plan de validación de la porción de software desarrollado. Los problemas estuvieron condicionados principalmente por la falta de experiencia por parte de los probadores y el desconocimiento que tenían del negocio a partir del cual se desarrolló el sistema. Asimismo la elaboración de los casos de prueba fue ineficiente y se contaba con muy pocos y los que estaban terminados poseían errores que dificultaron de manera sustancial el buen desenvolvimiento de las pruebas.

Ya en una segunda iteración y tomando como punto de referencia las pruebas internas, se observó una disminución considerable de los defectos encontrados, aproximadamente en un 59%. Resultado que aportó confianza y experiencia en el equipo de desarrollo. En esta etapa de prueba las dificultades más relevantes se concentraron en errores de funcionalidad, estas pruebas tuvieron un valor extra ya que los desarrolladores participaron directamente en ellas y a partir del conocimiento de los casos de uso previamente estudiados se efectuaron las pruebas con la calidad requerida. Una vez concluida esta fase y corregidas las no conformidades encontradas el software fue sometido a la liberación por parte de Calisoft, para comenzar a probarlo luego de cara al cliente final.

Durante las pruebas de liberación se observó una evolución en cuanto a los casos de prueba confeccionados ya que se le prestó especial atención a este aspecto, lo que permitió una disminución de los errores detectados centrados básicamente en validaciones del lado del cliente y estilos de diseño, y dejó el software listo para próximas iteraciones.

Las pruebas de aceptación por su parte mostraron un incremento significativo en la cantidad de errores detectados, situación que estuvo dada por constituir estas pruebas, el primer impacto del cliente con el software desarrollado y dar un vuelco sustancial a los requisitos levantados en un principio. Estos errores fueron solucionados ya para el Piloto, a lo cual se debió el resultado alcanzado.

Es importante resaltar que una vez concluida cada iteración de prueba se realizó un análisis por parte del equipo de desarrollo de cada no conformidad detectada, con el fin de determinar cuál o cuáles realmente constituyeron defectos del sistema o presentaban alguna variación según la descripción de los casos de uso y necesitaban ser solucionadas con urgencia.

Las pruebas se realizaron de forma tal que al comenzar cada ciclo se verificó la corrección de los defectos encontrados en la iteración anterior, lo que permitió de forma significativa un aumento de la calidad del software, promoviendo así la satisfacción del cliente sobre el producto.

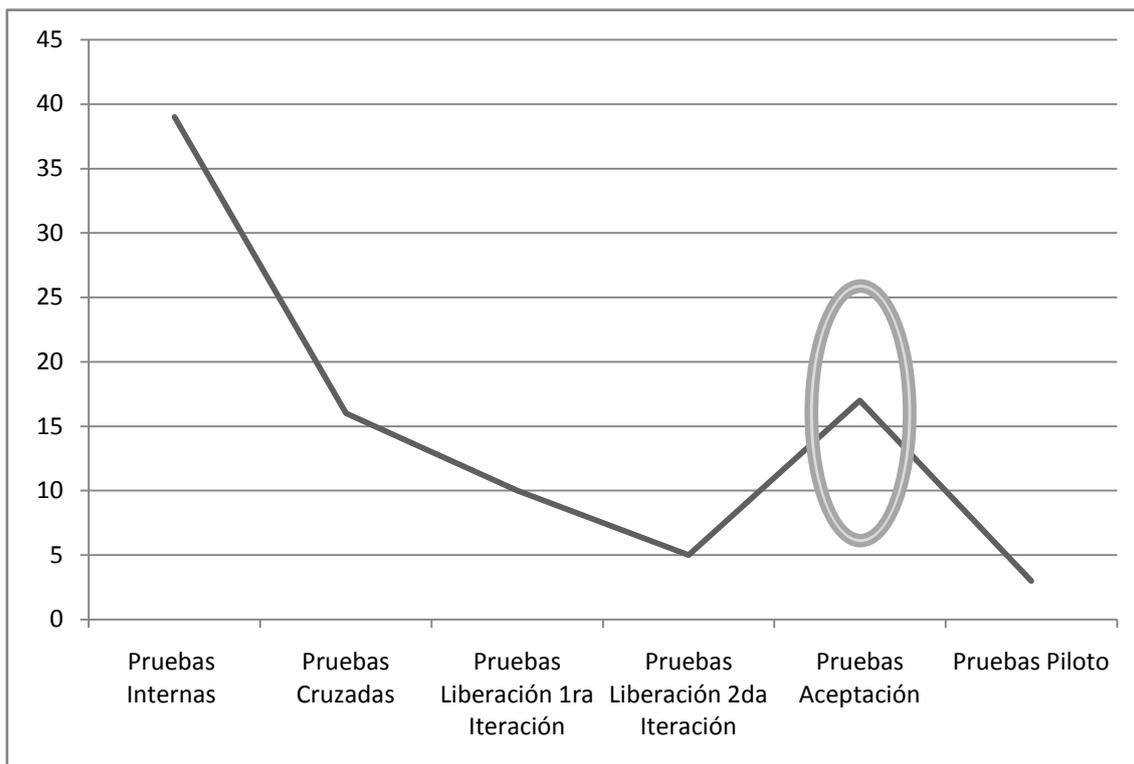


Ilustración 48: Resumen Gráfico de la Pruebas realizadas

El resumen de los errores detectados en cada iteración de pruebas, muestra la correcta evolución del software en cuanto a la reducción de las no conformidades se refiere. El pico observado en las pruebas de aceptación, constituye un fenómeno ocurrido debido a la presentación, por primera vez, del software al cliente final y a los cambios efectuados en la institución que dejaron obsoletos varios de los requisitos levantados al inicio del desarrollo. Esto generó una serie de no conformidades y peticiones de cambio mayor a las generadas en iteraciones anteriores.

Si comparamos los resultados obtenidos en las iteraciones de prueba con los resultados del Sistema de Investigación e Información Policial en general, se puede observar como el módulo Experticias Criminalísticas aportó un mínimo de defectos encontrados y se comportó de manera estable sin grandes diferencias de una iteración a otra.

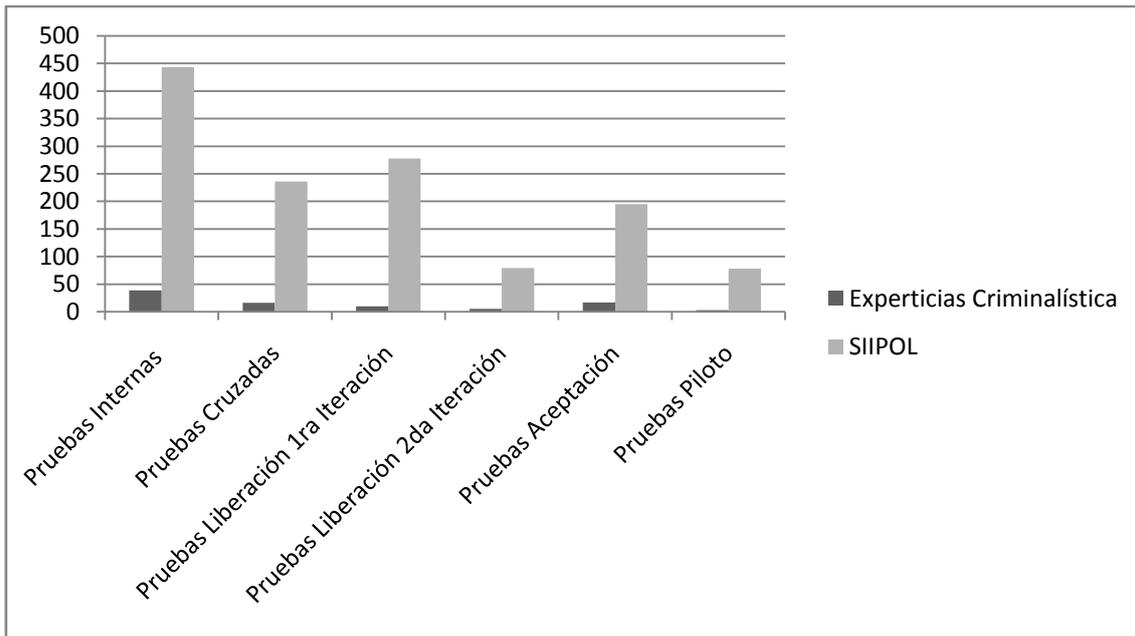


Ilustración 49: Experticias Criminalísticas respecto al SIIPOL

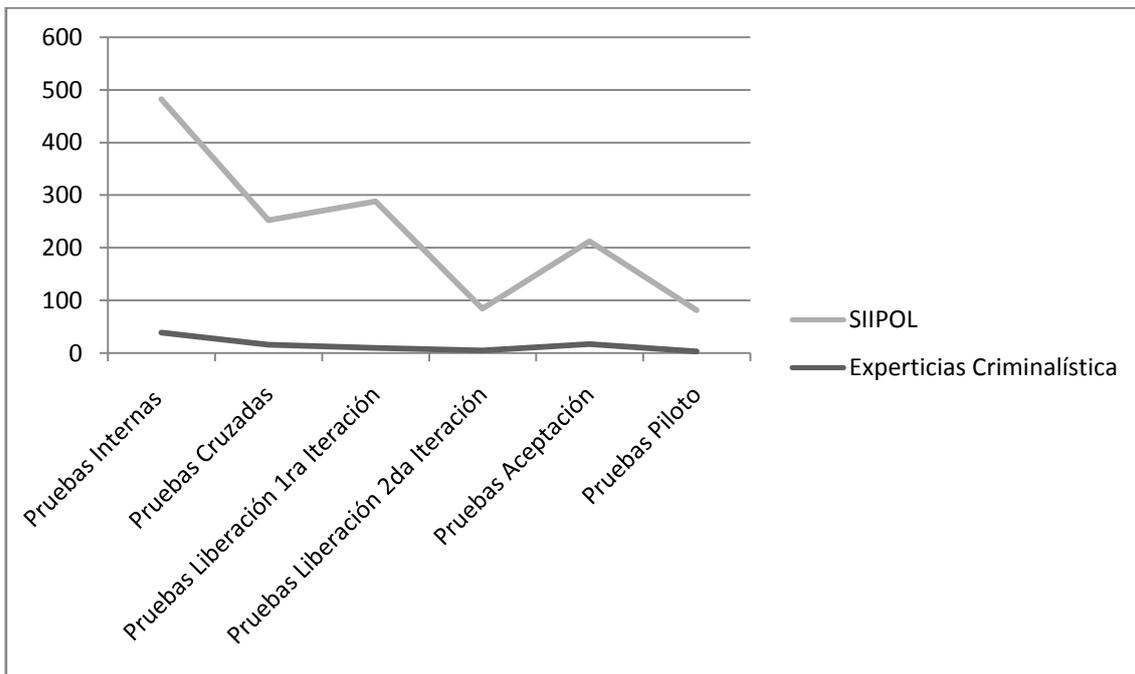


Ilustración 50: Experticias Criminalísticas respecto al SIIPOL

3.3 Conclusiones

La validación de software constituye un requisito de obligatorio cumplimiento en el proceso de desarrollo de un producto software con calidad, por eso la atención esmerada que se le presta a esta actividad, como parte del proceso, probando desde las partes más pequeñas y su engranaje, hasta su total funcionamiento, fuera y dentro de un ambiente real.

Las pruebas se realizaron de forma incremental lo que permitió la revisión detallada de cada uno de los componentes desarrollados y la aceptación final del cliente de una segunda versión, solidifica las bases y brinda confianza y experiencia para las versiones posteriores.

Conclusiones

El estudio del estado actual en el que se encuentra inmersa la Gestión Policial y la exposición de las tendencias mayormente utilizadas en el mundo contribuyeron de manera importante en la creación de la propuesta general para el nuevo SIIPOL.

La utilización de la metodología seleccionada facilitó el seguimiento y monitoreo de un proceso de desarrollo que agilice el trabajo de los implicados en el mismo.

Las herramientas y definiciones arquitectónicas puestas en práctica permitieron el desarrollo claro y fluido de un sistema construido sobre bases sólidas y un entorno bien definido.

Los patrones de diseño seleccionados para el desarrollo de la solución propuesta, formalizó un vocabulario común entre el quipo de desarrollo y proporcionó elementos reusables en el diseño del sistema de software.

La validación de software constituyó un requisito de obligatorio cumplimiento en el proceso de desarrollo de la solución propuesta.

Las pruebas realizadas posibilitaron la revisión detallada de cada uno de los componentes desarrollados así como el análisis del comportamiento de la aplicación tras su integración con el resto del sistema.

Con el desarrollo de la propuesta de solución se garantizó la automatización de un elevado por ciento de los procesos de la Coordinación Nacional de Criminalística referente a las experticias.

Recomendaciones

Como recomendaciones, una vez concluido el desarrollo de la presente investigación, se debe tener en cuenta:

- ✓ Realizar un proceso de refactorización no solo al módulo Experticias Criminalísticas sino al sistema de forma general, con el fin de limar cualquier aspereza detectada tras la validación del software.
- ✓ Confeccionar un informe que resuma las experiencias técnicas obtenidas en el desarrollo del sistema así como los resultados observados en las iteraciones de pruebas, que sirva de guía o material de consulta de nuevos proyectos con características similares al SIIPOL.
- ✓ Promover la utilización de metodologías ágiles en el desarrollo de futuros proyectos a partir de las experiencias positivas obtenidas.
- ✓ Estudiar la posibilidad de realizar un sistema similar para los órganos del MININT de Cuba, específicamente para el departamento de Criminalística.

Bibliografía

1. IAFIS. [En línea] 2009. <http://www.iafisgroup.com/html/afis.html>.
2. Forensic Technology. [En línea] 2009. <http://www.fti-ibis.com/>.
3. Dmapas. *Dmapas*. [En línea] 2009. http://dmapas.cl/productos_stegpol.htm.
4. Ministerio del Poder Popular para la Comunicación y la Información. [En línea] 2009. http://www.minci.gob.ve./entrevistas/3/181453/se_redujo_violencia.html.
5. Cuerpo de Investigaciones Científicas Penales y Criminalísticas. [En línea] 2009. <http://www.cicpc.gov.ve/>.
6. Conferencias de ISW. UCI, departamento de Ingeniería de Software. [En línea] 2009. <http://teleformacion.uci.cu>.
7. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. 2000.
8. Proceso Ágil Unificado. *sitio web The Agile Unified Process (AUP)*. [En línea] 2008. <http://www.ambyssoft.com/unifiedprocess/agileUP.html>.
9. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado. Manual de Referencia*.
10. Sitio Oficial JAVA. [En línea] 2009. <http://java.com/es/>.
11. Java 2 Platform, Enterprise Edition (J2EE) Overview. [En línea] 2009. <http://java.sun.com/j2ee/overview.html>.
12. Red Hat Developer Studio. *sitio Web de Red Hat Developer Studio*. [En línea] 2009. <http://www.redhat.com>.
13. Visual Paradigm for UML. *sitio Web de Visual Paradigm for UML*. [En línea] 2009. <http://www.visual-paradigm.com>.
14. Framework AME. [En línea] 2009. <http://ame.endesa.es/confluence/display/AMELib/Ficha+AMELib>.
15. **Walls, Craig y Breidenbach, Ryan.** *Spring in Action*. 2005.

16. **Bauer, Christian y King, Gaving.** *Hibernate in Action*. 2005.
17. Oracle. [En línea] 2009. <http://www.oracle.com/index.html>.
18. **Allamaraju, Subrahmanyam, y otros.** *Programación Java Server con J2EE*.
19. **Craig, Larman.** *UML y Patrones*. 2008.
20. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico*.
21. **Rodriguez Baryolo, Yunexis y Monagas Reyes, Miguel Ángel.** *Trabajo de Diploma: Ingeniería de Requerimientos del proceso de Criminalística del CICPC*. Ciudad de La Habana : s.n., 2007.
22. Eclipse. [En línea] 2009. <http://www.eclipse.org/>.
23. Programación en Castellano. [En línea] 2009. <http://www.programacion.com/>.

Glosario de Términos

A continuación, en orden alfabético, se muestra el significado de algunos términos usados en este documento que pueden dificultar la comprensión del mismo:

AFIS: Sistema Automático de Identificación de Huellas Dactilares, sistema informático compuesto de Hardware y Software integrados que permite la captura, consulta y comparación automática de huellas dactilares agrupadas por fichas decadactilares, o en forma de rastro latente (parte degradada de huella levantada en la escena de un crimen).

AJAX: Iniciales de Asíncrono Javascript y XML, es una técnica de desarrollo para crear aplicaciones web interactivas.

Ajax4JSF: Es un framework libre y Open Source que adiciona capacidades a las páginas JSF, a través de componentes implementados.

Apache TomCat: es un servidor web con soporte de servlet y JSP. Incluye el compilador Jasper, que compila las páginas JSP convirtiéndolas en servlet.

CASE: Sigla que corresponde a las iniciales de Computer Aided Software Engineering; y en su traducción al español significa Ingeniería de Software Asistida por Computación.

CICPC: Cuerpo de Investigaciones Científicas Penales y Criminalísticas de Venezuela, institución gubernamental encargada de investigar hechos delictivos.

CNC: Coordinación Nacional de Criminalística: coordinación del CICPC encargada de los asuntos de investigaciones criminalísticas.

Exadel: compañía de capital privado fundada en 1998. Líder en el suministro de componentes de aplicación avanzados para crear una nueva generación de soluciones empresariales que impulsen el código abierto y un modelo de código global innovador.

Faces Servlet: Es el Servlet para las componentes visuales del framework JSF.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Herramienta CASE: Aplicación informática destinada a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero, se utiliza para la modelación del sistema.

Hibernate: Framework de capa de persistencia para el lenguaje Java.

HQL: Lenguaje de consultas del framework Hibernate similar al SQL, pero que se refiere a clases y objetos no a tablas de la base de datos.

IDE: Entorno Integrado de Desarrollo, es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse exclusivamente para un lenguaje de programación o bien para varios (6)

J2EE: Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida.

JSF: framework de la capa de presentación para Java.

MVC: Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web.

NetBeans: Entorno Integrado de Desarrollo para crear aplicaciones en lenguaje Java.

Open Source: Representa el software de dominio público, esto significa sin licencia, cuyo código fuente está disponible y se le permite usar y modificar.

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.

Red Hat: compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux que lleva el mismo nombre: Red Hat Enterprise Linux, y de otra más, Fedora.

Rich-Faces: Conjunto de librerías para el framework JSF.

Servlet: Es un objeto que se ejecuta en un servidor o contenedor JEE, fue especialmente diseñado para ofrecer contenido dinámico desde un servidor web.

SIGEP: Sistema de Gestión de Penitenciaria.

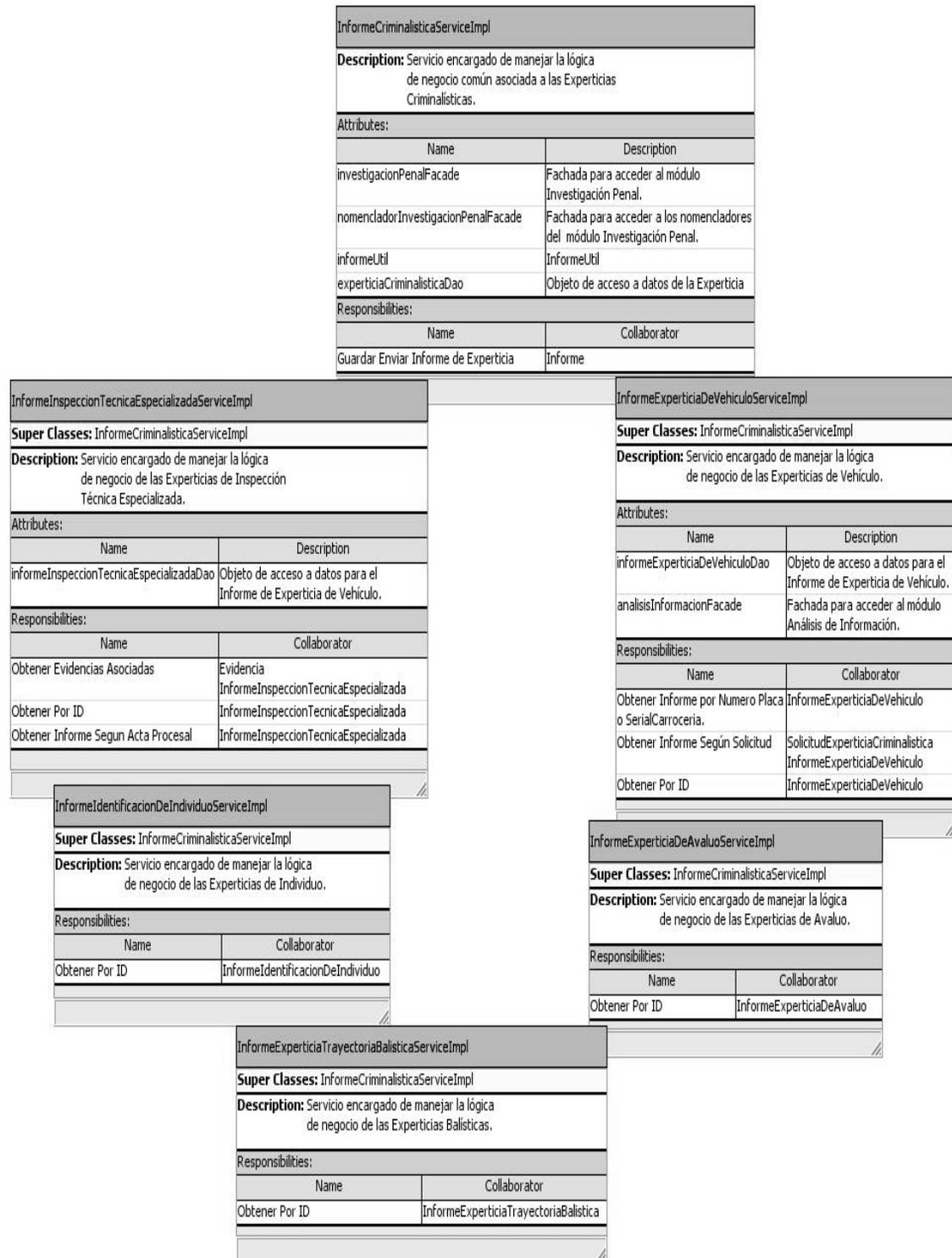
SIIPOL: Sistema de Investigación e Información Policial.

Spring: Framework de la capa de lógica de negocio para Java.

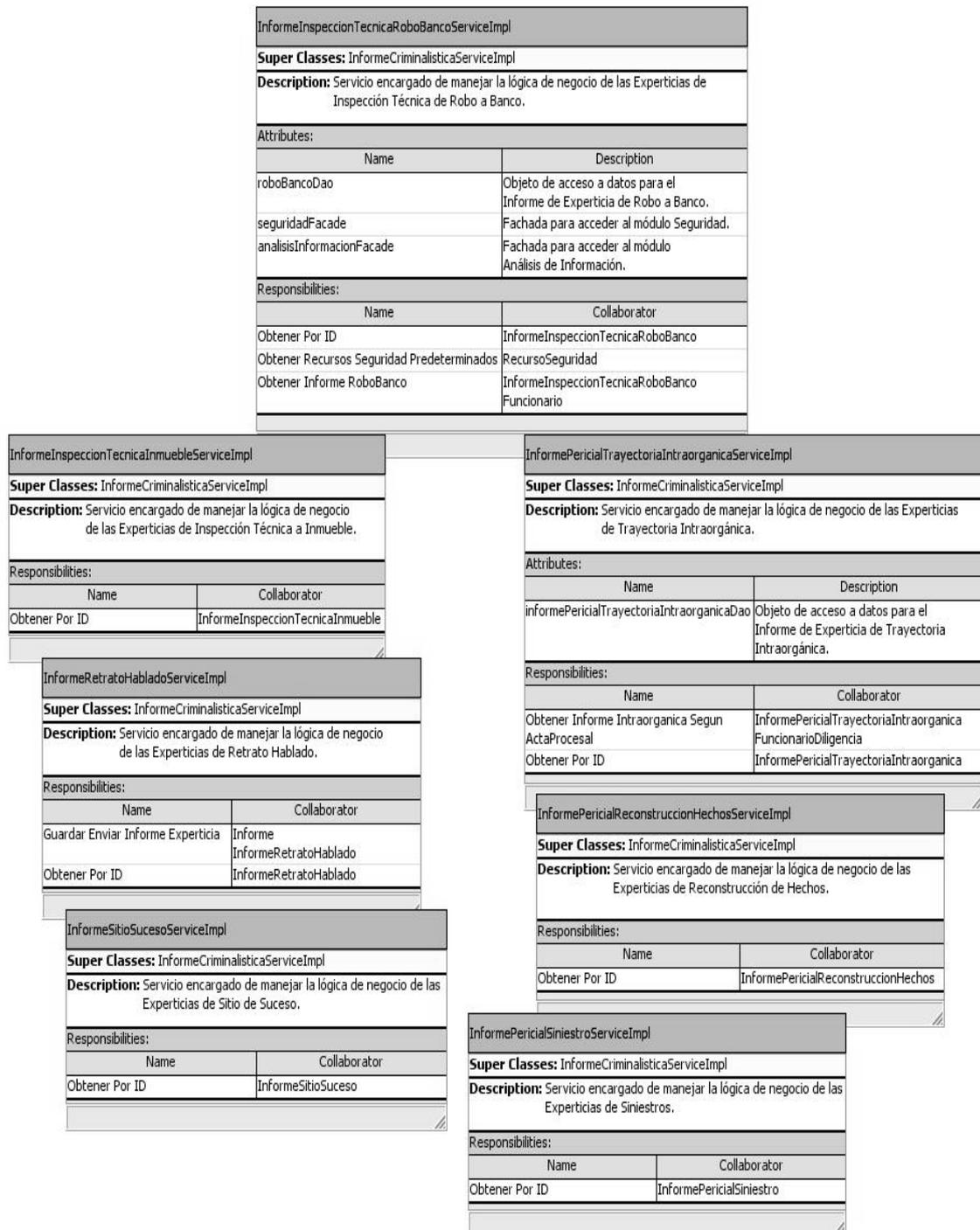
STEGPOL: Es un Sistema de Información Geográfica con tecnología de punta chileno.

Anexos

Anexo 1: Tarjetas CRC para la capa de negocio del Módulo Experticias Criminalísticas.



Anexo 2: Tarjetas CRC para la capa de negocio del Módulo Experticias Criminalísticas.



Anexo 3: Interfaces de Usuario.

Anexo 3.1: IU Incluir Planilla de Retrato Hablado

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL

 Bienvenido(a): AAIL YADIEL RAMOS RODRIGUEZ
 24/04/2009 ... 13:13
Inicio
Ayuda
Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
- Asesoría Jurídica
- Investigación en Ciencias ...
- Análisis de Información
- Estadísticas
- Dotación de Equipos Polic...
- Evidencia
- Auditoría
- Administración

INCLUIR PLANILLA DE RETRATO HABLADO

Planilla Retrato Hablado
Imagen Retrato Hablado
Personas Inculminadas

Datos Generales

No. Solicitud	S-09-0001-00007
No. Expediente	K-08-0001-00001
Importancia	Normal
Dependencia Solicitante	Dirección General
Destinatario	Dirección General

Persona que Aporta los Datos

Nombres y Apellidos	José Castro
Identificación	T-329
Condición respecto al caso	Testigo

Datos de la Planilla de Retrato Hablado

Edad: Igual a # 22

Sexo: Femenino

Comentarios

Mujer joven es sorprendida cometiendo robo en Centro Comercial.

Otros Detalles

Boca: Grande	Cara: Redonda	Peso (kg): Igual a # 45
Mentón: Prominente	Color de Piel: Trigueña	Estatura (m): Menor que # 2
Contextura: Débil		

Barba y Bigote

Barba:

Bigote:

Patilla:

Cabello

Tipo: Crespo Color: Rojizo

Tipo	Eliminar
Liso	<input type="button" value="Eliminar"/>
Total: 1	

Color	Eliminar
Amarillo	<input type="button" value="Eliminar"/>
Total: 1	

Nariz

Tamaño: Pequeña Tipo: Respingada

Características: Tabique Nasal Hundido

Característica	Eliminar
Torcida	<input type="button" value="Eliminar"/>
Total: 1	

Labios

Superior	Inferior
Espesor: Delgado	Espesor: Delgado
Características: Prominente	Características: Prominente
<input type="button" value="Adicionar"/>	<input type="button" value="Adicionar"/>

Característica	Eliminar
Leporino	<input type="button" value="Eliminar"/>
Total: 1	

Característica	Eliminar
Leporino	<input type="button" value="Eliminar"/>
Total: 1	

Ojos

Izquierdo	Derecho
Color del Iris: Azul	Color del Iris: Azul
Tipo: Achinados	Tipo: Achinados

Base Legal

Ley: COPPC Artículo:

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPPC	Artículo 1	<input type="button" value="Eliminar"/>
Total: 1		

99

Anexo 3.2: IU Incluir Informe de Inspección Técnica

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL
Bienvenido(a): AAI-YADIEL RAMOS RODRIGUEZ 24/04/2009 ... 13:21 Inicio Ayuda Salir

AGENDA DE TRABAJO

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

MENÚ PRINCIPAL

- Acceso
- Gestión de Despacho
- Atención Telefónica
- Investigación Penal
- Aprehensión
- Archivos Históricos
- Investigación Criminalística
- Asesoría Jurídica
- Investigación en Ciencias ...
- Análisis de Información
- Estadísticas
- Dotación de Equipos Polic...
- Evidencia
- Auditoría
- Administración



INCLUIR INFORME DE INSPECCIÓN TÉCNICA

Informe
Elementos Asociados
Montaje Fotográfico

Datos Generales

Tipo de Informe	Informe Inspección Técnica	No. Solicitud	S-09-0001-00074
No. Expediente	K-08-0001-00001	Importancia	Normal
		Dependencia Solicitante	DISIP
		Destinatarios	DISIP
		Dirección del Sitio del Suceso	NO ESPECIFICADO, NO ESPECIFICADO

Evidencias Relacionadas

No. Evidencia	Nombre Evidencia	Descripción
EK-08-0001-00001-15	Arma	121212
Total: 1		

Datos del Informe

Fecha en que se realizó la Inspección

Fecha Hora :

Descripción

B I U x₂ x₂ Fuente Tamaño

Se realizó inspección técnica a entidad estatal situada en Avenida Bolívar.

Base Legal

Ley Artículo Adicionar

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPPC	Artículo 1	Eliminar
Total: 1		

Personalizar Comunicación
Vista Previa
Incluir
Guardar
Limpiar
Cancelar

Anexo 3.3: IU Incluir Informe Pericial de Experticia de Vehículo

SIIPOL SISTEMA DE INVESTIGACIÓN E INFORMACIÓN POLICIAL
Bienvenido(a): AAIL YADIEL RAMOS RODRIGUEZ 24/04/2009 ... 13:18 Inicio Ayuda Salir

AGENDA DE TRABAJO
 Notificaciones
 Asignaciones
 Recepciones
 Aprobaciones
 Revisiones
 Remisiones
 Borradores
 Archivos

MENÚ PRINCIPAL
 Acceso
 Gestión de Despacho
 Atención Telefónica
 Investigación Penal
 Aprehensión
 Archivos Históricos
 Investigación Criminalística
 Asesoría Jurídica
 Investigación en Ciencias ...
 Análisis de Información
 Estadísticas
 Dotación de Equipos Polic...
 Evidencia
 Auditoría
 Administración

INCLUIR INFORME PERICIAL DE EXPERTICIA DE VEHÍCULO

Datos Generales

No. Solicitud	S-09-0001-00002
No. Expediente	K-08-0001-00001
Importancia	Normal
Dependencia Solicitante	Dirección General
Destinatarios	Dirección General

Descripción de la Evidencia

No. Evidencia	EK-08-0001-00001-11 (Vehículo)
No. Placa	999999
Serial de Carrocería	
Serial de Motor	
Clase	Aeronave
Marca	ALFA ROMEO
Modelo	No Indica

Informe Pericial de Experticia de Vehículo

Peritaje: Vehículo Exposición Motivo Conclusiones

Informe / Peritaje

B I U x₂ x² Fuente: Tamaño: T

Vehículo hurtado en calle residencial.

Lugar de Revisión

Estado	Municipio	Tipo de Lugar	Nombre
Distrito Capital	Libertador	Chivera	Parqueo Anauco

Experto Técnico Revisor

Nombres y Apellidos: AAIL Humberto Rivero Guevara Incluir

Nombres y Apellidos	Credencial	Cargo	Eliminar
AAIL ADONYS ALEA BOFFILL	0000012	Operador de Computadores	Eliminar
Total: 1			

Base Legal

Ley: COPPC Artículo: Adicionar

De Conformidad con lo establecido en la Ley	Artículo	Eliminar
COPPC	Artículo 1	Eliminar
Total: 1		

Personalizar Comunicación
 Vista Previa
 Incluir
 Guardar
 Limpiar
 Cancelar

101