



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE

INGENIERO EN CIENCIAS INFORMÁTICAS

**Desarrollo de los Submódulos Solicitudes y Archivo Internacional del Módulo Interpol del
Sistema de Investigación e Información Policial**

Autor:

Rolando Gual Pérez

Tutor:

Ing. Kenny López Espinosa

Cotutora:

Ing. Yunexis Rodríguez Baryolo

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Firma del Autor

Rolando Gual Pérez

Firma del Tutor

Ing. Kenny López Espinosa

AGRADECIMIENTOS

A mi familia, en especial a mi madre, mi padre y mis hermanos por su dedicación y amor. Tengo una deuda de amor que no sé cómo pagar.

A Estrella Abreu Aquino, Leonel Rodríguez y Yuri Yero por su amor y por ser como una familia.

A mis amigos nuevos y viejos, en especial a quienes considero mi familia: Nadian, Leandro, Ernesto, Adrián, Alexis, Yisel, Patricia y Kirenia.

A otros excelentes amigos Yunislema, Orquidea, Ezequiel y Yordany.

A mi tutor y su esposa Mailín Díaz por su confianza, sus enseñanzas y sus consejos.

A Oigres y Albey por ser excelentes personas y hacerme crecer como ser humano.

A Leydis, Breissy y Tahimi por darme la oportunidad de ser un hermano mayor.

A Yunet Gasca Suárez por haber sido mi amiga, mi amante y mi confidente. Por haber sido durante mucho tiempo, mi más grande inspiración.

DEDICATORIA

A mis abuelos y en especial a Alberto Pérez Burgos.

A Miladis María Pérez Beltrán y Reynaldo Gual Hernández.

A Reinier Gual Pérez y Reynaldo Gual Pérez.

A Yunet Gasca Suárez.

Si la vida me preguntara por que los amo, diría porque han hecho de mi todo lo que soy, porque no existe momento en el que no piense en ustedes, porque el tiempo que hemos pasado juntos me sabe a poco, porque junto a ustedes no siento miedo.

Porque son y siempre formarán parte de mí... hasta el fin de mis días.

RESUMEN

El Sistema de Investigación e Información Policial (SIIPOL) se perfila como la principal herramienta para la lucha contra el delito en Venezuela. Permite el intercambio rápido de información entre divisiones y departamentos responsables de darle solución en tiempo breve a un caso investigativo. Al SIIPOL, como sistema de gestión de información policial para el CICPC¹, se le han integrado paulatinamente las diferentes divisiones y dependencias de esta institución. Para ello, se han informatizado la mayor cantidad de procesos que se realizan en el CICPC para la lucha contra el delito.

El presente documento, recoge la investigación y proceso de desarrollo de los submódulos Solicitudes Internacional y Archivo Internacional del SIIPOL. El módulo de Solicitudes agrupa las funcionalidades asociadas a la realización de solicitudes internacionales y sus respuestas, además de otros documentos como ordenes, cartas y oficios, que forman parte de una investigación internacional. El submódulo Archivo Internacional agrupa las funcionalidades asociadas a la gestión de información en el Archivo Internacional, donde se guarda la información obtenida durante años de investigaciones internacionales realizadas.

En el documento se encuentran las principales características de otros sistemas de gestión de información policial, que se consultaron como parte de la investigación. Se recogen las principales características de las herramientas empleadas para la modelación e implementación de la solución. Se tratan las principales características de la metodología empleada y los principales artefactos generados por los flujos de trabajo Análisis, Diseño e Implementación según la metodología, además de la validación de la propuesta de solución.

¹ *Cuerpo de Investigaciones Científicas Penales y Criminalísticas.*

ÍNDICE

INTRODUCCIÓN.....	1
Capítulo 1. Fundamentación Teórica.	5
1.1 Introducción.....	5
1.2 Sistemas de Gestión de Información Policial.....	5
1.2.1 Sistema Mundial de Comunicación Policial de Interpol (I-24/7).....	6
1.2.2 Plataforma México.....	6
1.2.3 Sistema de Información Schengen (SIS).....	7
1.2.4 Sistema Integrado de Información de Justicia Criminal (SIJC).....	8
1.2.5 Sistema de Investigación e Información Policial.....	8
1.2.6 Procesos de la Dirección de Policía Internacional.....	10
1.3 Metodología, Lenguajes y Herramientas de Desarrollo.....	11
1.3.1 Rational Unified Process (RUP). Proceso y metodología de desarrollo.	11
1.3.2 Lenguaje de Modelado. Unified Modeling Language (UML).....	14
1.3.3 Herramienta Case.....	14
1.3.4 Plataforma de Desarrollo: Java Enterprise Edition (JEE6).....	15
1.3.5 Lenguaje de Programación. Java.....	16
1.3.6 Entorno de Desarrollo Integrado (IDE). Eclipse Helios.....	16
1.3.7 Frameworks de Desarrollo.....	17
1.3.8 Sistema Gestor de Base de Datos. Oracle 10g reléase 2.	21
1.4 Propuesta de solución.	22
1.5 Conclusiones.....	25
Capítulo 2. Análisis, Diseño e Implementación de la propuesta de solución.	26
2.1 Introducción.....	26
2.2 Análisis.....	26
2.2.1 Entrada al Análisis.....	26
2.2.2 Modelo de Análisis.....	32
2.2.3 Análisis de los casos de uso más significativos de la solución.....	32
2.3 Diseño.....	37
2.3.1 Arquitectura.....	37
2.3.2 Modelo de Diseño.....	43
2.3.3 Modelo de datos.....	51

2.4	Implementación	55
2.4.1	Modelo de Implementación.....	55
2.5	Conclusiones	59
Capítulo 3. Validación de la implementación de la propuesta de solución.....		60
3.1	Introducción	60
3.2	Prueba.....	60
3.3	Métodos de Prueba	61
3.3.1	Pruebas de Caja Blanca	61
3.3.2	Pruebas de Caja Negra	61
3.4	Niveles de Prueba	62
3.4.1	Pruebas de Unidad.....	62
3.4.2	Pruebas de Integración.....	63
3.4.3	Pruebas de Sistema	63
3.5	Conclusión.....	69
Conclusiones Generales.....		70
Recomendaciones		71
Referencias Bibliográficas.....		72
Bibliografía Consultada.....		74
Glosario de términos.....		76
Anexos.....		78

Fig. 1 Fases y flujos de trabajo según RUP en dos dimensiones.....	13
Fig. 2 Funcionamiento JSF (20).....	18
Fig. 3 Estructuras de los módulos de Spring (22).....	19
Fig. 4 Configuración Básica de Hibernate (23).....	21
Fig. 5 Propuesta de solución.....	23
Fig. 6 Estructura lógica del SIIPOL.....	24
Fig. 7 Diagrama de Casos de Uso del Sistema.....	31
Fig. 8 Vista de análisis. Gestionar Solicitud de Información Archivo Internacional.....	34
Fig. 11 Vista de colaboración. CU Gestionar Solicitud de Información Archivo Internacional.....	35
Fig. 14 Estructura de paquetes de la solución.....	39
Fig. 15. Vista de Presentación. CU Gestionar Solicitud de Información de Archivo Internacional.....	44
Fig. 19 Vista de Negocio. CU Gestionar Solicitud de Información de Archivo Internacional.....	45
Fig. 22 Diagrama de Secuencia. CU Gestionar Solicitud de Información de Archivo Internacional.....	51
Fig. 25 Diagrama de Clases del Dominio Módulo Archivo.....	52
Fig. 26 Diagrama de Clases del Dominio Módulo Solicitudes.....	53
Fig. 27 Modelo Relacional.....	54
Fig. 28 Subsistema de Implementación.....	56
Fig. 29 Vista Presentación módulo Archivo. Páginas jsp.....	57
Fig. 30 Vista Presentación módulo Archivo. Beans manejados.....	58
Fig. 31 Vista Negocio módulo Archivo.....	58
Fig. 35 Caso de Prueba Gestionar Respuesta Información Archivo Internacional.....	64
Fig. 36 Resultados Pruebas Cruzadas.....	65
Fig. 37 Resultados Prueba Calidad Interna.....	66
Fig. 38 Resultados Comparación Pruebas Internas.....	67
Fig. 39 Resultados Pruebas de Calisoft.....	68
Tabla 1 Gestionar Solicitud de Información de Archivo Internacional.....	33
Interfaz 1 Gestionar Solicitud de Información de Interpol.....	36
Anexo 1 Modelo Relacional principales entidades.....	78
Anexo 2 Modelo Relacional principales entidades.....	79

INTRODUCCIÓN

Los actuales índices de violencia y criminalidad existentes en el mundo, han posicionado a la seguridad ciudadana como una de las principales demandas de la sociedad hacia las autoridades estatales. La inseguridad ciudadana se define como el temor a posibles agresiones, asaltos, secuestros y violaciones de los que se puede ser víctima. Esta situación es el resultado de un proceso que lleva ya varias décadas, en el que han coincidido diversos factores de tipo social, económico, cultural e institucional que operaron como posibilitadores para la generación o reproducción de diferentes formas de comportamientos violentos (1). Venezuela, no está ajena a esta situación y ha sido declarada en varias ocasiones como uno de los países más violentos del mundo. La situación en las calles y el deterioro de los órganos policiales en este país, contribuyen a la inseguridad ciudadana (2).

Venezuela cuenta con el Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC), institución que tiene como misión garantizar la eficiencia en la investigación del delito mediante su determinación científica, asegurando el ejercicio de la acción penal que conduzca a una sana administración de justicia, y que tiene entre sus objetivos optimizar las acciones de investigación criminal tendentes a lograr el esclarecimiento de los hechos delictivos, así como dotar al capital humano del CICPC de herramientas, mecanismos logísticos y de infraestructura que garanticen el óptimo desempeño de sus funciones (3). Entre los despachos con que cuenta el CICPC está la Dirección de Policía Internacional, que tiene como misión dar apoyo a todas las organizaciones, autoridades y servicios en labores de prevención, detección y lucha contra los delitos. Provee servicios esenciales a otras agencias policiales en un esfuerzo por combatir la delincuencia a nivel mundial.

Con el inicio del gobierno del presidente Hugo Rafael Chávez Frías, se establece como unas de las líneas estratégicas de su gobierno la seguridad ciudadana. Con el objetivo de disminuir los índices de criminalidad existentes, se empiezan a introducir una serie de cambios en el aparato policial que continúa arrastrando un modelo atado a la concepción y filosofía del estado burgués, que ha redundado en policías que están sumergidas o penetradas por mafias, organizaciones dedicadas al crimen inclusive (4) (5). Como parte de estas transformaciones, en el marco de los acuerdos Cuba - Venezuela y en aras de darle cumplimiento a los objetivos que el CICPC tiene definidos como organismo policial, se desarrolla el Sistema de Investigación e Información Policial (SIIPOL). El SIIPOL actualmente está en su versión 4,

mejora la gestión de información y ofrece organización y control en las labores cotidianas del CICPC, y será empleado en todas sus oficinas en cada uno de los estados venezolanos. Este marco de trabajo común para los funcionarios de esta institución, facilita el flujo de información entre divisiones, departamentos y despachos en general. En dependencia de las necesidades existentes para el esclarecimiento de los hechos delictivos, las oficinas del CICPC coordinan mediante este sistema la intervención de determinada división, por su especialización en determinado campo de investigación. Sin embargo, el SIIPOL no abarca en su totalidad los procesos relacionados con los diferentes despachos con que cuenta el CICPC.

La Dirección de Policía Internacional, es responsable de la gestión de la información para la prevención y esclarecimiento de los hechos delictivos que puedan repercutir en el extranjero y viceversa. Los procesos en este despacho no están informatizados, y se realiza de forma manual todo el trabajo con las solicitudes, órdenes y otros documentos que permiten llevar a cabo las diferentes labores vinculadas a una investigación. Lo anterior trae como inconveniente, gran cantidad de tiempo empleado en la búsqueda y organización de la información en el archivo internacional, por el gran volumen de información existente en formato duro, lentitud en el intercambio de información entre divisiones, departamentos y despachos responsables de darle solución en un tiempo breve a los casos investigativos; lo que influye negativamente en el desempeño de la Dirección de Policía Internacional y en las labores del CICPC.

La información del Archivo Internacional se encuentran expuesta a condiciones que facilitan su deterioro con el transcurso del tiempo, y como es evidente su pérdida. Considerándose esta como un riesgo, pues la información en muchos casos puede ser insustituible.

Por lo descrito anteriormente se hace necesario que al SIIPOL se le agreguen nuevas funcionalidades que permitan una mejor gestión de la información asociada a los delitos internacionales y reducir el tiempo de respuestas de las solicitudes realizadas.

Teniendo en cuenta la problemática descrita anteriormente, se puede formular el **problema a resolver** a partir de la siguiente interrogante: ¿Cómo gestionar la información de la Dirección de Policía Internacional para el SIIPOL?

El **objeto de estudio** definido para el presente trabajo de diploma se centra en los Sistemas de Gestión de Información Policial y el **campo de acción** se enmarca en el Sistema de Investigación e Información Policial.

Como **idea a defender** se plantea: “Con el desarrollo de los submódulos Solicitudes y Archivo Internacional se permitirá gestionar la información de la Dirección de Policía Internacional del CICPC a través del SIIPOL”.

El **objetivo general** de esta investigación es desarrollar los submódulos Solicitudes y Archivo Internacional para el Sistema de Investigación e Información Policial.

Una vez definido el objetivo general y en aras de darle cumplimiento, se plantean los siguientes objetivos específicos:

- Elaborar el marco teórico orientado a sistemas de gestión de información policial.
- Elaborar la documentación que refleje el desarrollo de la solución.
- Implementar los submódulos diseñados respetando la arquitectura definida para el SIIPOL.
- Integrar la solución al Sistema de Investigación e Información Policial a medida que se obtienen resultados satisfactorios.
- Validar el correcto funcionamiento de los submódulos implementados.

Para darle cumplimiento a lo antes descrito se han trazado las siguientes tareas:

- Estudio de los conceptos fundamentales de los sistemas de gestión de información policial.
- Estudio de aplicaciones informáticas con características similares a lo que se desea desarrollar, para conocer que se ha realizado en el mundo.
- Estudio de los procesos de negocio que tienen lugar en la Dirección de Policía Internacional para entender su funcionamiento.
- Estudio de la metodología y frameworks de desarrollo para el desarrollo de la solución.
- Estudio las herramientas y lenguaje de programación para el desarrollo de los submódulos.

- Estudio del modelo de casos de uso del sistema asociado a los submódulos a implementar para conocer las funcionalidades a implementar.
- Estudio de los patrones de diseño empleados en el desarrollo del SIIPOL.
- Realización de las pruebas de Unidad, Integración y de Sistema a los componentes desarrollados.

Para el correcto desarrollo de la investigación y definición del objeto de estudio; así como garantizar el conocimiento del estado del arte, su evolución y relación con otros fenómenos, se utilizaron los siguientes **métodos científicos**:

- **Analítico-sintético** para analizar la documentación generada durante la captura de requisitos y extraer de la misma los elementos esenciales relacionados con el objeto de estudio y el campo de acción de la investigación. Para la recopilación de la bibliografía relacionada con la utilización de patrones de diseño en sistemas gestores de información policial. Además del estudio de las diferentes metodologías y herramientas usadas en la construcción de estos sistemas, con el fin de obtener las mejores prácticas de desarrollo e integrarlas en la solución.

Las **técnicas de investigación** se utilizaron en la obtención de los datos fundamentales que enriquecieron la investigación. El **análisis documental** estuvo presente en el manejo del volumen de la bibliografía consultada; en el estudio de sistemas de gestión de información policial, así como en la búsqueda de patrones arquitectónicos y de diseño.

El contenido a desarrollar en el presente trabajo está estructurado en tres capítulos que agrupan los contenidos de la siguiente manera:

Fundamentación teórica: Expone los elementos teóricos de la investigación tales como: sistemas de gestión de información policial, metodología, lenguaje y herramientas de desarrollo.

Análisis, diseño e implementación de la propuesta de solución: Expone el modelo de análisis, el de diseño y el de implementación que responden directamente a la solución del problema.

Validación de la implementación de la propuesta de solución: Contiene las definiciones y los resultados de las diferentes pruebas realizadas a los submódulos, para verificar que cumplen con lo definido en la captura de requisitos.

Capítulo 1. Fundamentación Teórica.

1.1 Introducción

En el presente capítulo se realiza un análisis de los principales elementos que se tienen en cuenta para el desarrollo de la investigación. Se consulta información de otros sistemas de gestión de información policial que tienen éxito en el mundo, por ser la base para solucionar problemas similares. Se exponen las principales características de las tecnologías, de la metodología, lenguajes y herramientas que se utilizan para la implementación del SIIPOL que tienen impacto directo en el desarrollo de la propuesta de solución. De manera general este capítulo reúne los conocimientos mínimos necesarios para llevar a cabo la investigación.

1.2 Sistemas de Gestión de Información Policial

Los órganos policiales sin importar su estructura organizativa van en busca de una mejora continua de los objetivos que como institución tienen definidos, para garantizar una mejor administración de justicia y disminuir los niveles de violencia. Las tecnologías de la información constituyen un paso decisivo en esta tarea, han permitido el desarrollo de los sistemas de gestión de información policial. Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, procedimientos y procesos de la organización (6). Los sistemas de gestión de información policial no están muy alejados de esta definición y tienen como objetivo fundamental:

- La centralización de toda la información obtenida en una institución policial.
- Informatizan los procesos que en ella se desarrollan y en muchos casos se vinculan con sistemas similares.
- Organización en el manejo de archivos clasificados (5).
- Impulso a la investigación debido a la facilidad para encontrar y manipular la información (5).

A continuación se mencionan algunas características de sistemas de gestión de información policial utilizados en diferentes países.

1.2.1 Sistema Mundial de Comunicación Policial de Interpol (I-24/7)

El Sistema Mundial de Comunicación Policial de Interpol I-24/7, es un instrumento puntero en la lucha contra la delincuencia internacional y el terrorismo. Utiliza complejos medios de lucha contra la delincuencia junto con la tecnología más avanzada para potenciar la capacidad de respuesta de los organismos encargados de la aplicación de la ley, proporcionándoles una pasarela a las bases de datos policiales de Interpol. El sistema I-24/7 aprovecha la amplia gama de recursos de Interpol para ofrecer a los agentes encargados de la aplicación de la ley, un acceso inmediato a información policial de importancia vital. Las Oficinas Centrales Nacionales (OCN) y las fuerzas del orden autorizadas como por ejemplo las patrullas de vigilancia de fronteras, pueden introducir un nombre y realizar una búsqueda instantánea para encontrar información sobre sus conexiones con la delincuencia internacional. Asimismo, pueden realizar búsquedas cruzadas, en las que se relaciona información de muy diversos tipos sobre delincuencia internacional y sobre investigaciones policiales en cuestión de segundos. El sistema I-24/7 es un instrumento único de mejora de la cooperación, colaboración y comunicación policial internacional (7).

1.2.2 Plataforma México

La Plataforma México se perfila como la principal herramienta de información policial para combatir con eficacia el crimen organizado en este país. Esta herramienta integra en un mismo espacio todas las bases de datos que tienen que ver con la seguridad pública, para que las instancias policiales y de procuración de justicia del país, lleven a cabo las actividades de prevención y combate al delito mediante metodologías y sistemas homologados. Algunos de los datos que reúne esta red informática privada son:

- Órdenes de aprehensión
- Vehículos robados, entrada y salida de extranjeros.
- Licencias de conducir, fichas de delincuentes y bandas criminales.
- Modos de operación.
- Criminales detenidos, reos sentenciados y delincuentes en proceso judicial (8).

México tendrá sus bases de datos policiales a disposición de todos los países miembros de Interpol mediante un enlace de su Plataforma México con el I-24 / 7. El enlace significa que en México rellenarán automáticamente las bases de datos de Interpol sobre personas buscadas, documentos robados o

perdidos y vehículos robados. Asimismo, cuando un funcionario policial mexicano consulte una base de datos nacional, simultáneamente y sin solución de continuidad verá en pantalla los datos de INTERPOL (9).

1.2.3 Sistema de Información Schengen (SIS)

El espacio Schengen representa un territorio donde está garantizada la libre circulación de las personas y está basado en la firma del Tratado Schengen². Los estados que firmaron dicho tratado han suprimido las fronteras interiores y han establecido una única frontera exterior. Francia, Alemania, Bélgica, España Portugal, Austria, Dinamarca, Finlandia y Suecia son algunos de los países miembros.

Como parte fundamental del dispositivo Schengen se crea el Sistema de Información Schengen que permitirá que las autoridades designadas por las partes, mediante un procedimiento de consulta automatizado, dispongan de descripciones de personas y objetos, al efectuar controles en la frontera y comprobaciones y otros controles de policía y aduanas realizados dentro del país (10) Este sistema manipula los datos referentes a:

- Personas buscadas para su detención a efectos de extradición.
- Extranjeros incluidos en la lista de no admisibles.
- Datos relativos a personas desaparecidas o que deban ser objeto de protección.
- Datos relativos a testigos y personas citadas para comparecer ante las autoridades judiciales.
- Datos relativos a personas o vehículos a efectos de vigilancia discreta o de control específico.
- Datos relativos a objetos buscados con vistas a su incautación, o como pruebas en un procedimiento penal.

Para luchar contra la delincuencia organizada e internacional y contra el terrorismo el Sistema de Información Schengen (SIS) y la base de datos de Interpol, responden a la exigencia de establecer un sistema integrado de intercambio de información sobre los pasaportes robados y extraviados. La Interpol

²*El espacio Schengen representa un territorio donde está garantizada la libre circulación de las personas. Los Estados que firmaron el Tratado han suprimido todas las fronteras interiores y en su lugar han establecido una única frontera exterior.*

recibe, almacena y difunde datos para ayudar a los servicios policiales a prevenir y combatir la delincuencia internacional.

1.2.4 Sistema Integrado de Información de Justicia Criminal (SIJC)

Garantiza el intercambio rápido y efectivo de información entre los agentes de Puerto Rico y de cualquier jurisdicción de Estados Unidos y países conectados como es el caso de Canadá. Incluye un registro de ofensores sexuales conectado al Registro Nacional de Ofensores Sexuales de Justicia Federal, para que los ciudadanos puedan estar informados y prevenir riesgos en la comunidad. Cualquier persona autorizada tendrá acceso a los bancos de datos de historiales criminales en todos los estados y jurisdicciones de los Estados Unidos (11).

Los sistemas a los que se hacen referencia con anterioridad, tiene similitud con la Dirección de Policía Internacional en cuanto a la información que gestionan en la lucha contra el delito. Sin embargo, estas similitudes no hacen posible que estos sistemas puedan ser utilizados como solución al problema que presenta la Dirección de Policía Internacional, pues son el resultado de las condiciones e intereses de cada país en particular. No existe acceso al código fuente, no existe una descripción de los procesos que automatizan, no existe una descripción de las herramientas que se utilizaron para su desarrollo y no hay referencia alguna de tener algún departamento encargado de la gestión de delitos internacionales.

1.2.5 Sistema de Investigación e Información Policial

El SIIPOL es utilizado en el CICPC en sus labores de lucha contra el delito, permite el trabajo en conjunto entre diferentes oficinas del CICPC en diferentes regiones del estado venezolano de manera rápida y sincronizada. El sistema cuenta con diferentes módulos que responden a diferentes áreas con las que cuenta esta institución, el módulo Investigación Penal es el encargado de gestionar el conjunto de diligencias orientadas al descubrimiento y comprobación científica del delito, sus características y la identificación de sus autores o partícipes. El módulo Investigación Criminalística, es el encargado de gestionar y facilitar todo el proceso de creación de las Solicitudes de Experticias Criminalísticas y permite dar una respuesta correcta en el menor plazo de tiempo posible, así como permitir el control de dicho proceso. El módulo Investigación Interna es el encargado de gestionar el proceso de investigación disciplinaria que se realiza con fines administrativos a los funcionarios del CICPC, que incurran en alguna

violación prevista en su reglamento disciplinario, entre otros módulos. El sistema tiene conexión con otros sistemas de gestión de información como el INTT³ para el control de vehículos y SAIME⁴ para el control de personas. Permite además obtener estadísticas confiables como por ejemplo: delitos más frecuentes ocurridos por zonas, días con mayor índice delictivo, horas de mayor índice delictivo y robos de autos. Este sistema de gestión de información, fue desarrollado utilizando Rational Unified Process como metodología de desarrollo, con Java Server Faces en la capa de presentación, Spring en la capa de negocio e Hibernate en la capa de acceso a datos y otras herramientas como Visual Paradigm para él modelado y Eclipse en su versión 3.6 Helios como IDE de desarrollo.

1.2.5.1 *Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC).*

El CICPC como institución rectora de la lucha contra el delito y el crimen en Venezuela, tiene como visión ser la Institución indispensable, por su reconocida capacidad científica y máxima excelencia de sus recursos, con la finalidad de alcanzar el más alto nivel de credibilidad nacional e internacional en la investigación del fenómeno delictivo organizado y criminalidad violenta (3).

El CICPC tiene entre sus funciones:

- Practicar las diligencias que le ordene el Ministerio Público, encaminadas a investigar y hacer constar la perpetración de un hecho punible, con todas las circunstancias que puedan influir en su calificación y la responsabilidad de los autores y demás partícipes, identificación de las víctimas, de las personas que tengan conocimiento de los hechos, así como el aseguramiento de los objetos activos y pasivos relacionados con el delito.
- Colaborar con los demás órganos de seguridad ciudadana en la creación de centros de prevención del delito y en la organización de los sistemas de control o bases de datos criminalísticas para compartir la información de los servicios de inteligencia, en cuanto a narcotráfico, terrorismo internacional, desaparición de personas, movimiento de capitales ilícitos, delincuencia organizada y otros tipos delictivos.
- Elaborar, analizar en coordinación con el Instituto Nacional de Estadística y presentar al Ministerio del Interior y Justicia las estadísticas de criminalidad, cuando sean requeridas, con el objeto de

³ Instituto Nacional de Transporte Terrestre.

⁴ Servicio Administrativo de Identificación, Migración y Extranjería

adoptar las políticas de prevención y se apliquen las medidas necesarias para garantizar el funcionamiento del estado en materia de seguridad.

- Desarrollar políticas de prevención, orientación, publicidad, colaboración e información a fin de aplicar medidas técnicas que permitan reducir y evitar la actividad delictiva.
- Auxiliar en caso de necesidad a la Dirección Nacional de Identificación y Extranjería, y colaborar en la identificación, localización y aprehensión de ciudadanos extranjeros solicitados por otros países.
- Las demás actuaciones o funciones que le sean atribuidas de conformidad con la ley (3).

1.2.6 Procesos de la Dirección de Policía Internacional

La Dirección de Policía Internacional tiene entre sus funciones mantener el enlace con los diferentes organismos policiales del país, así como oficinas centrales nacionales e internacionales, adoptar las resoluciones de convenios internacionales acordadas por la Organización Internacional de Policía Criminal (O.I.P.C.), representar al país en las reuniones de la Asamblea General de la O.I.P.C. en los casos que así lo decida la Dirección General Nacional, mantener contacto permanente con el Servicio Administrativo de Identificación, Migración y Extranjería (SAIME) a fin de disponer de información sobre los movimientos migratorios de ciudadanos, cumplir con los parámetros legales establecidos, ante la solicitud de extradición de ciudadanos, mantener actualizados los registros policiales de venezolanos y extranjeros solicitados internacionalmente para asegurar el adecuado flujo de información (3).

Entre los procesos fundamentales se pueden mencionar: Tratamiento de Información Internacional, encargado de las deportaciones y cualquier tipo de información internacional relacionada con personas y elementos de análisis de información que se investigan en los países que poseen tratados entre policías internacionales. Posee además el proceso de Extradiciones que describe las acciones y elementos necesarios para cada tipo de extradición, y el proceso de Archivo de Información Internacional encargado de archivar toda la información procesada por la Dirección de Policía Internacional, así como brindar la información necesaria que está archivada en esta Dirección. De manera general estos procesos tienen inicio cuando algún despacho investigativo del CICPC realiza una solicitud a la Dirección de Policía Internacional. Posteriormente la División de Archivo Internacional perteneciente a esta dirección se encarga de procesarla y clasificarla, se elabora la respuesta y es enviada al despacho que realizó la solicitud.

1.3 Metodología, Lenguajes y Herramientas de Desarrollo

Generalmente a la hora de conformar el ambiente de desarrollo de cualquier producto de software, se realiza un estudio de las metodologías de desarrollo, lenguajes de programación, lenguajes y herramientas de modelado que son reconocidos en el desarrollo de sistemas con las características del que se desea desarrollar. De estas se seleccionan las más acordes a las necesidades reales del equipo de desarrollo, ya sea por la cantidad de bibliografía existente, por el conocimiento de los desarrolladores, por el costo de su uso, por la facilidad de actualización con nuevas versiones, entre otras.

Los submódulos a desarrollar van a formar parte del SIIPOL, sistema mucho más grande que se ha desarrollado durante 5 años con RUP como metodología de desarrollo, JSF como framework de presentación, Spring como framework de negocio e Hibernate como frameworks de acceso a datos. La selección de la metodología, los frameworks de desarrollo y demás herramientas que conforman el ambiente de desarrollo en general, siempre han sido trabajo del equipo de arquitectura del proyecto SIIPOL. Al encontrarse la información del ambiente de desarrollo en el documento de la arquitectura⁵, se excluye del marco de la investigación el estudio detallado de cada una. Se hace mención en los siguientes subepígrafes de las características más relevantes de cada uno de los elementos del ambiente de desarrollo, que tienen impacto en el desarrollo de la propuesta de solución presentada más adelante.

1.3.1 Rational Unified Process (RUP). Proceso y metodología de desarrollo.

Durante el desarrollo de un producto de software la mayoría de las actividades se tornan muy engorrosas y difíciles de controlar, haciendo riesgoso el proceso de desarrollo. Para prevenir y mitigar los riesgos de un desarrollo descontrolado se utiliza una metodología de desarrollo de software, que dicta quién tiene que hacer determinada actividad, cómo hacerla y cuándo (12).

Un proceso de desarrollo de software es un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema de software. RUP describe cómo aplicar efectivamente enfoques comprobados comercialmente⁶ para el desarrollo de software, es más que un simple proceso; es una forma de trabajo genérico que puede utilizarse para el desarrollo de una gran variedad de sistemas

⁵ Descripción de la Arquitectura de Software

⁶ Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas.

informáticos, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos. No obstante, los verdaderos aspectos de esta metodología están en tres frases claves: dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental (12).

RUP ofrece organización y control en las labores llevadas a cabo para el desarrollo de un software de gran magnitud. Permite desarrollar en cada fase y flujo de trabajo, un grupo de artefactos que sirven de apoyo a las fases y flujos posteriores, lo que permite el control y el entendimiento de lo que se ha obtenido hasta ese momento y dando una visión de lo que se desea alcanzar. Tiene como inconveniente que un cambio en algún elemento del software, hace necesario la actualización de toda la documentación asociada al mismo.

Características del Proceso Unificado de Desarrollo.

Dirigido por casos de uso:

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales, son la herramienta para especificar los requisitos del sistema y guían su diseño, implementación y prueba. De manera general guían todo el proceso de desarrollo (12).

Centrado en la Arquitectura:

La arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema. Es una representación general de los elementos que lo componen. Cada uno de estos elementos va perfeccionándose y detallándose a medida que sean necesarios para el desarrollo del software. La arquitectura permite comprender el sistema, pues ofrece a todos los involucrados en el desarrollo del software, una visión de lo que se desea obtener. Ofrece organización al desarrollo, pues permite dividir el sistema en partes que puedan ser implementadas por diferentes equipos de desarrollo y cuando sea necesario, integrarlo como un todo. Fomenta la reutilización pues permite detectar elementos que puedan ser utilizados por todos y concentrarlos en un solo lugar, simplificando el tiempo de desarrollo. Cada producto tiene una forma y una función; la arquitectura es esta forma y debe diseñarse para permitir que el sistema evolucione. Para encontrar la forma los arquitectos deben trabajar sobre la comprensión de las funciones claves, es decir sobre los casos de uso más significativos del sistema. Los casos de uso claves

son especificados al detalle en términos de clases, subsistemas y componentes, lo que aporta a la estabilidad y maduración de la arquitectura (12).

Iterativo Incremental:

RUP divide el desarrollo del producto en mini proyectos y cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo y los incrementos, al crecimiento del producto. En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple su objetivo el esfuerzo resulta en un incremento y el desarrollo continúa con la siguiente. En cada iteración se realizan tareas de cada uno de los flujos principales (ver Fig. 1) propuestos por RUP (12).

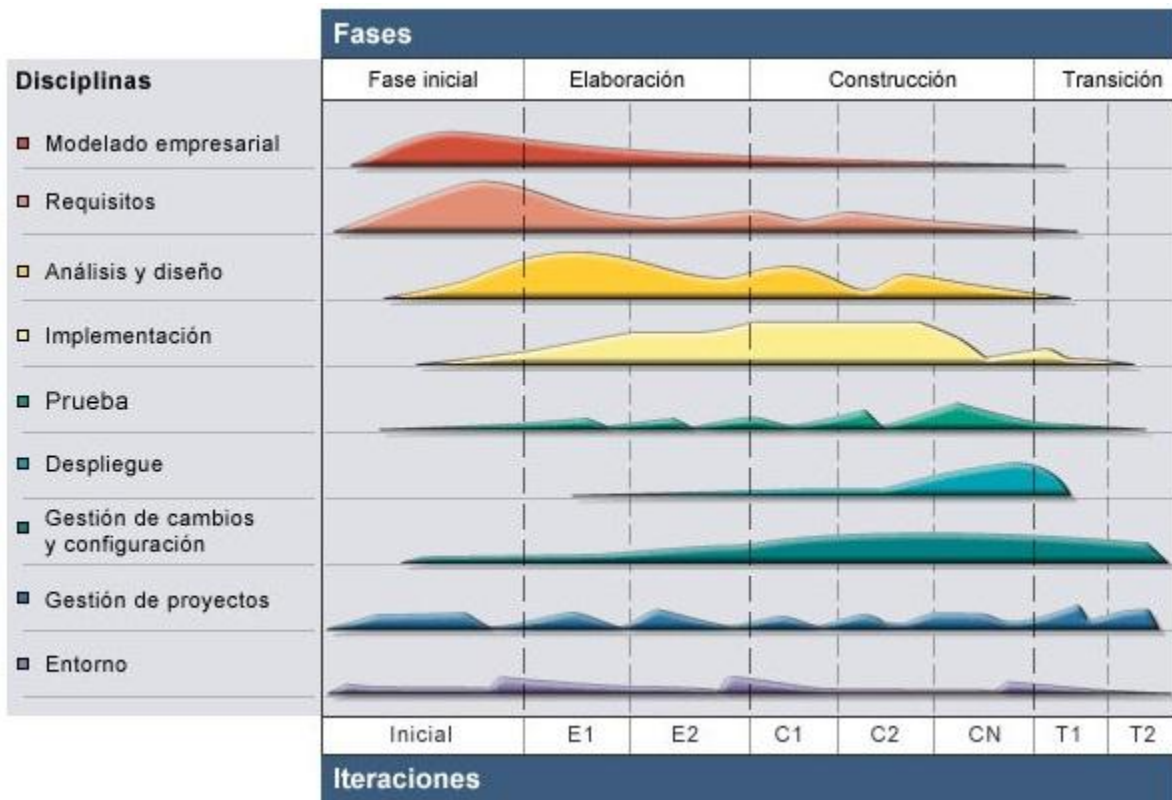


Fig. 1 Fases y flujos de trabajo según RUP en dos dimensiones.

Las características descritas anteriormente están estrechamente relacionadas, la eliminación de una de ellas le restaría valor al Proceso Unificado de Desarrollo (12). Funcionan como un engranaje para ayudar a los desarrolladores de software en la creación de un producto.

La presente investigación tiene como punto de partida los casos de uso obtenidos en la ingeniería de requerimientos desarrollada en una investigación anterior, y se enmarca en los flujos de trabajo Análisis, Diseño e Implementación.

1.3.2 Lenguaje de Modelado. Unified Modeling Language (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas software, así como para el modelado del negocio (13). Este lenguaje le permite a los desarrolladores de software, tener una representación de los elementos del sistema a desarrollar, ya sean componentes reutilizables, esquemas de base de datos, procesos de negocio o funciones de un sistema. UML establece un lenguaje común y formal entre los desarrolladores, facilitando el desarrollo de cualquier producto de software a partir de puntos de vista comunes.

1.3.3 Herramienta Case

CASE⁷ proporciona la ingeniero la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y de diseño asistidos por computadora que utilizan los ingenieros de otras disciplinas, las herramientas CASE ayudan a garantizar que la calidad se diseñe antes de llegar a construir el producto (14).

1.3.3.1 Visual Paradigm 8.0

Visual Paradigm for UML (VP-UML) es una herramienta de diseño UML y herramienta CASE UML diseñada para ayudar al desarrollo de software. VP-UML soporta el Lenguaje de Modelado Unificado (UML), ofrece un conjunto de herramientas a los equipos de desarrollo de software, útiles para la captura de requisitos, software de planificación, la planificación de controles, el modelado de clases y modelado de datos (15).

⁷ Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora)

Características fundamentales:

- Fácil de usar, pues sus componentes se encuentran relacionados lo que facilita la creación de cualquier tipo de diagrama.
- Permite la generación de código a partir de diagramas para la plataforma Java.
- Puede utilizar técnicas de ingeniería inversa para llevar de código fuente a diagramas de clases, manteniendo de esta manera la sincronización entre el modelo y el código.
- Permite la generación de scripts para varios sistemas gestores de bases de datos y aplicar ingeniería inversa en bases de datos existentes.

Esta herramienta permite generar los artefactos que exige la metodología de desarrollo, de acuerdo a la fase y el flujo de trabajo en los que se encuentra el desarrollador. Permite establecer una correspondencia entre los artefactos creados, lo implementado y lo que se desea implementar.

1.3.4 Plataforma de Desarrollo: Java Enterprise Edition (JEE6)

La plataforma Java Enterprise Edition (JEE) proporciona a los desarrolladores un conjunto de APIs tales como Java Transaction API (JTA)⁸, Java Persistence API⁹, Java Server Pages Technology¹⁰, Java Server Faces Technology, Dependency Injection para reducir el tiempo de desarrollo, mejorando el rendimiento de las aplicaciones y reduciendo su complejidad.

Está diseñada para soportar aplicaciones que implementan servicios empresariales, define una arquitectura para la implementación de servicios con múltiples niveles de aplicaciones, que ofrecen la escalabilidad, la accesibilidad y capacidad de gestión necesaria para las aplicaciones de nivel empresarial (16).

⁸ Provee una interfaz estándar para transacciones declarativas.

⁹ Utiliza un enfoque de mapeo objeto/relacional para cerrar la brecha entre un modelo orientado a objetos y una base de datos relacional.

¹⁰ Permite poner fragmentos de código servlet directamente dentro de un documento basado en texto.

1.3.5 Lenguaje de Programación. Java

Java fue desarrollado por la compañía Sun Microsystems en 1995 con el propósito de cubrir las necesidades tecnológicas de punta. Su principal característica y a su vez lo que lo diferencia ampliamente de los demás lenguajes de programación es la independencia que tiene a nivel de plataforma; esto se debe a que se le ha creado una máquina virtual para cada sistema operativo (SO).

Es un lenguaje simple, orientado a objetos, distribuido, robusto, seguro, de arquitectura neutra, portátil, de alto desempeño, dinámico y que tiene soporte para hilos múltiples. Posee gran cantidad de librerías que permiten y facilitan el rápido desarrollo de aplicaciones, es libre y de código abierto.

Características del lenguaje:

- **Portabilidad:** la existencia de distintos tipos de procesadores y ordenadores llevo a los ingenieros de Sun, que era importante conseguir un software que no dependiera del tipo de procesador utilizado. Se planteó la necesidad de conseguir un código capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado el programa no es necesario su modificación por el hecho de cambiarlo de procesador o ejecutarlo en otra máquina. La clave fue desarrollar un código neutro que estuviera preparado para ejecutarse en una Máquina Virtual¹¹. De esta manera se evita realizar un programa diferente para cada CPU o plataforma (17).
- **Seguridad:** Permite que un error producido por parte de la aplicación no pueda comprometer la seguridad ni la estabilidad del sistema en su totalidad. Todos los errores son registrados de alguna manera por las herramientas utilizadas, de forma que la información puede ser utilizada para corregir el error (17).
- **Manejo de memoria:** Java es un lenguaje administrado, lo que significa que la memoria utilizada por las aplicaciones no es manejada por ellas sino por la máquina virtual subyacente (17).

1.3.6 Entorno de Desarrollo Integrado (IDE). Eclipse Helios

El Eclipse Helios es una plataforma de desarrollo de software multilenguaje que contiene un IDE y un sistema de plugins para extenderlo. Este IDE está desarrollado en Java y brinda la funcionalidad de

¹¹ *Interpreta este código neutro convirtiéndolo a código particular de la CPU o chip utilizado.*

programar en este lenguaje por defecto, a través del plugin básico Java Development Tools (JDT). Además de poderse extender los lenguajes que puede soportar el IDE, un conjunto de plugins permite adicionar otras funcionalidades útiles como editores visuales de distintos tipos de archivos, conexión con repositorios de control de versiones, etc. El SDK¹² de Eclipse incluye por defecto el JDT¹³ que ofrece un compilador incremental¹⁴ de Java. La versión del IDE utilizada en el proyecto reúne un conjunto de plugins provistos por defecto para el desarrollo común de aplicaciones. Permite además, agregarle nuevos plugins para lograr la integración con las herramientas de trabajo del equipo de desarrollo y para soportar el desarrollo con frameworks específicos. Entre los plugins adicionados al IDE están: el WTP (permite desarrollo web, adicionando soporte para servidores), SpringIDE (para trabajo con el contexto de Spring), Subclipse para integración con el control de versiones, el Hibernate Tools para la visualización e interacción con Hibernate, además de plugins para el desarrollo de aplicaciones JSF y JBoss Tools RichFaces para soporte Ajax en las páginas JSF.

1.3.7 Frameworks de Desarrollo

“Un framework es una herramienta que provee servicios básicos a las aplicaciones que lo utilizan, define una arquitectura para la aplicación y reduce la cantidad de código cliente necesario utilizando código preprogramado para abstraer al programador de las tareas más simples permitiendo que se concentre en resolver los problemas del negocio... (18)”.

1.3.7.1 Java Server Faces Framework 1.1

La tecnología Java Server Faces es un frameworks de componentes del lado del servidor para la construcción de aplicaciones web basadas en tecnología Java. Consiste en un API para representar los componentes y la gestión de su estado, control de eventos, validación del lado del servidor y conversión de datos. Librerías de etiquetas para adicionar componentes a las páginas web y poder conectarlos con objetos del lado del servidor. Estas características alivian la carga de construcción y mantenimiento de las aplicaciones web con interfaces de usuario del lado del servidor (19). Está basado en el patrón Modelo

¹² Software Development Kit conjunto de herramientas que le permite al programador crear aplicaciones.

¹³ Java Development Tools

¹⁴ Compila un archivo de código de Java cada vez que se agrega o modifica, sin necesidad de recompilar el proyecto entero.

Vista Controlador (MVC) y utiliza Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas. Permite a los desarrolladores pensar en términos de componentes, eventos, backing beans (20). En la Fig. 2 se hace una representación básica del funcionamiento de JSF, donde el navegador web envía un HTTP Request a través de algún evento generado por los componentes de la interfaz de usuario. Este evento es capturado por el controlador que elabora la respuesta y posteriormente la envía al navegador a través de HTTP Response.

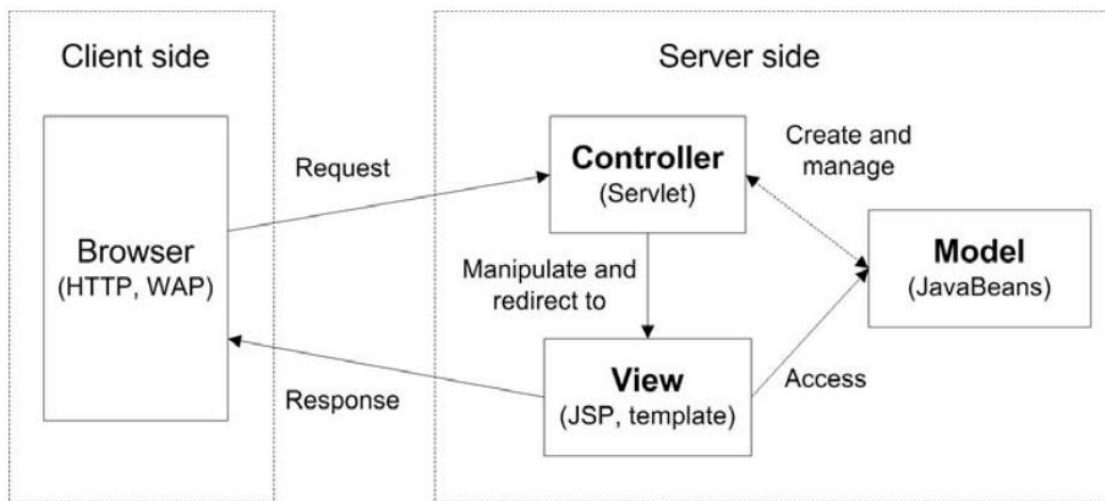


Fig. 2 Funcionamiento JSF (20).

Entre los principales componentes de la tecnología Java Server Faces están.

- Librerías de etiquetas personalizadas que permiten expresar una interfaz Java Server Faces dentro de una página JSP.
- Un modelo de eventos en el lado del servidor.
- Beans administrados.

Rich Faces es un framework de código abierto que añade capacidad Ajax a aplicaciones JSF existentes sin recurrir a JavaScript. Permite definir por medio de etiquetas JSF diferentes partes de una página JSF que desea actualizar con una solicitud Ajax y ofrecer varias opciones para enviar las peticiones Ajax al servidor.

Son características de Rich Faces las siguientes:

- Se integra perfectamente en el ciclo de vida de JSF.
- Incluye funcionalidades Ajax, de modo que nunca se observa el JavaScript y tiene un contenedor Ajax propio.
- Contiene un set (conjunto) de componentes visuales, los más comunes para el desarrollo de una aplicación web rica (Rich Internet Application), con un número bastante amplio que cubren casi todas nuestras necesidades.
- Soporta css (21).

1.3.7.2 Spring Framework 3.1

Spring Framework es una plataforma que proporciona soporte integral para el desarrollo de aplicaciones Java. Está compuesto por alrededor de 20 módulos como se muestra en la Fig. 3, en los que se encuentran funcionalidades para acceso a datos, programación orientada a aspectos, desarrollo de aplicaciones web, lo que permite que se puedan utilizar cada módulo en dependencia de las necesidades del equipo desarrollo.

Spring brinda soporte al framework de presentación Java Server Faces y se integra con el framework de acceso a datos Hibernate.

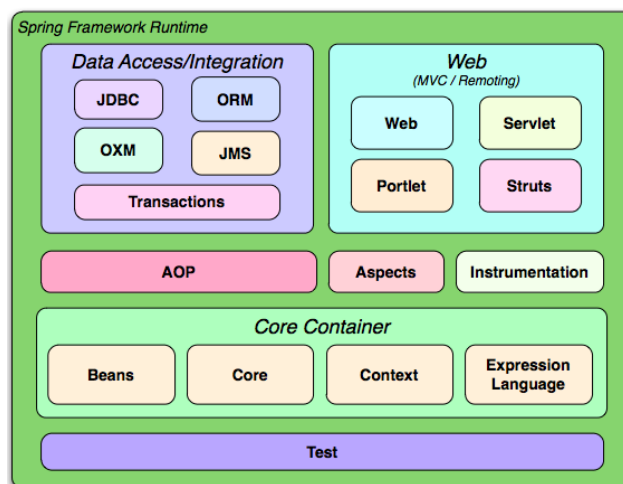


Fig. 3 Estructuras de los módulos de Spring (22).

Entre los elementos más sobresalientes de Spring están la Inversión de Control (IoC) e Inyección de Dependencia (DI), que permite resolver las dependencias de un objeto en el momento de su creación, a través de argumentos del constructor, argumentos de un método Factory (Fábrica) o propiedades que se registran en el objeto después de su construcción. El uso del principio de DI permite un código más claro y se alcanza un grado más alto de desacoplamiento, pues el bean no busca sus dependencias de manera activa, ni conoce su posición, ni su tipo concreto.

La Programación Orientada a Aspectos (AOP) complementa a la Programación Orientada a Objetos (OOP), proporcionando otra manera de pensar acerca de la estructura del programa. El módulo de AOP de Spring descompone el programa en aspectos, convirtiéndolos en servicios separados de la lógica de negocio que se ejecutan de manera transversal a la funcionalidad base en toda la aplicación. Además separa las funcionalidades propias del sistema de las del negocio.

Spring soporta la integración con Hibernate, implementación de DAO y estrategias para transacciones. Es responsable de todas las configuraciones relevantes de Hibernate cuando se encarga de la integración con él (22). Maneja las transacciones declarativas para objetos Java.

1.3.7.3 *Hibernate Framework 4.1.2*

Hibernate es una solución de Object/Relational Mapping (ORM) para entornos Java, que tiene como objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la manipulación de estos. Elimina la problemática de implementar persistencia de objetos en un soporte relacional, casando los dos modelos de manera que el programador trabaje desde Java usando el modelo objetual, y permite la persistencia en múltiples gestores de bases de datos (23).

El término mapeo objeto/relacional hace referencia a la técnica de representación del modelo de objetos a una representación relacional del modelo de datos, y viceversa. Hibernate a través de los mapeos XML sabe cómo cargar y guardar objetos de clases persistentes, y permite el mapeo de colecciones, de asociaciones y de herencia.

A Hibernate lo acompaña un API de consultas intuitivo y flexible (Criteria), que permite realizar consultas sobre una clase persistente en particular y obtener los resultados que cumplan con restricciones específicas. Además las consultas realizadas con criterias son de fácil mantenimiento.

En la Fig. 4 se brinda una perspectiva bastante general de la arquitectura de Hibernate, en la que se observa como el framework se sitúa entre la aplicación y la base de datos, relacionado con la primera a través de los objetos persistentes¹⁵. Además para realizar la conexión a un gestor de bases de datos, Hibernate utiliza el driver de conexión específico para el gestor y utiliza los parámetros de conexión que pueden ser especificados, en el XML Hibernate.cfg.xml o en el fichero HibernateProperties.

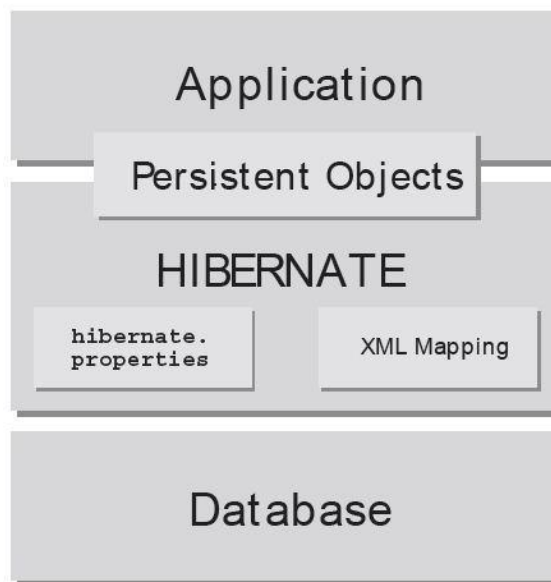


Fig. 4 Configuración Básica de Hibernate (23).

1.3.8 Sistema Gestor de Base de Datos. Oracle 10g reléase 2.

Oracle es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acudir a estos de forma recurrente con un modelo definido como relacional (24). Este sistema gestor de bases de

¹⁵ Clases en una aplicación que representan los conceptos del negocio y cuya información va a persistir en una base de datos.

datos posee características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma (25).

Entre sus características se encuentran:

- **Versatilidad:** Se pueden crear códigos en Java o en C++ utilizando dicha base de datos tanto en Windows como en Linux. Exporta los datos y los migra desde una plataforma a la otra sin causar problemas de integridad o pérdida de datos, migra con su propio software tanto de SQL Server como de MySQL funcionando sobre cualquier plataforma libre, por lo que es muy útil para todos los programadores.
- **Potencia:** Ofrece un rendimiento mucho mayor que cualquier otra plataforma de Base de Datos. Permite al administrador asignar sus propias zonas de memoria a los datos y cualidades, se controla en todo momento; tanto el crecimiento como el rendimiento de los distintos esquemas que componen una BD sobre Oracle, aunque esto suponga un problema, ya que los administradores deben estar pendientes de su configuración para no sufrir fallos debido a algún problema de almacenamiento.
- **Seguridad:** La seguridad de Oracle como sistema gestor de base de datos es alta, pues al no tener de integración con Windows lo hace invulnerable a los defectos que posee el sistema operativo, además de poseer un sistema de seguridad muy avanzado.
- **Complejidad:** Complejo para los administradores que no estén familiarizados con las bases de datos, su alto rendimiento es directamente proporcional a su nivel de complejidad (25).

1.4 Propuesta de solución.

La propuesta de solución consiste en desarrollar los submódulos Solicitudes y Archivo Internacional pertenecientes al módulo Interpol, e integrarlos al SIIPOL. El SIIPOL es una aplicación cliente-servidor, donde los usuarios se conectan mediante el protocolo HTTPS¹⁶ a una red virtual privada. La misma se conecta al servidor que aloja el contenedor web Apache Tomcat en el que se encuentra contenida la aplicación, que a su vez se conecta al servidor de bases de datos Oracle 10g como se muestra en la *Fig.*

¹⁶ *Hyper Text Transfer Protocol Secure (Protocolo seguro de transferencia de hipertexto).*

5. Por tal motivo se respetará la arquitectura base definida para el SIIPOL y se tomará como punto de partida para el análisis, diseño e implementación de las funcionalidades de cada submódulo. En correspondencia con la arquitectura definida, la Fig. 6 muestra una abstracción de la estructura lógica de la aplicación y los frameworks que intervienen en cada capa.

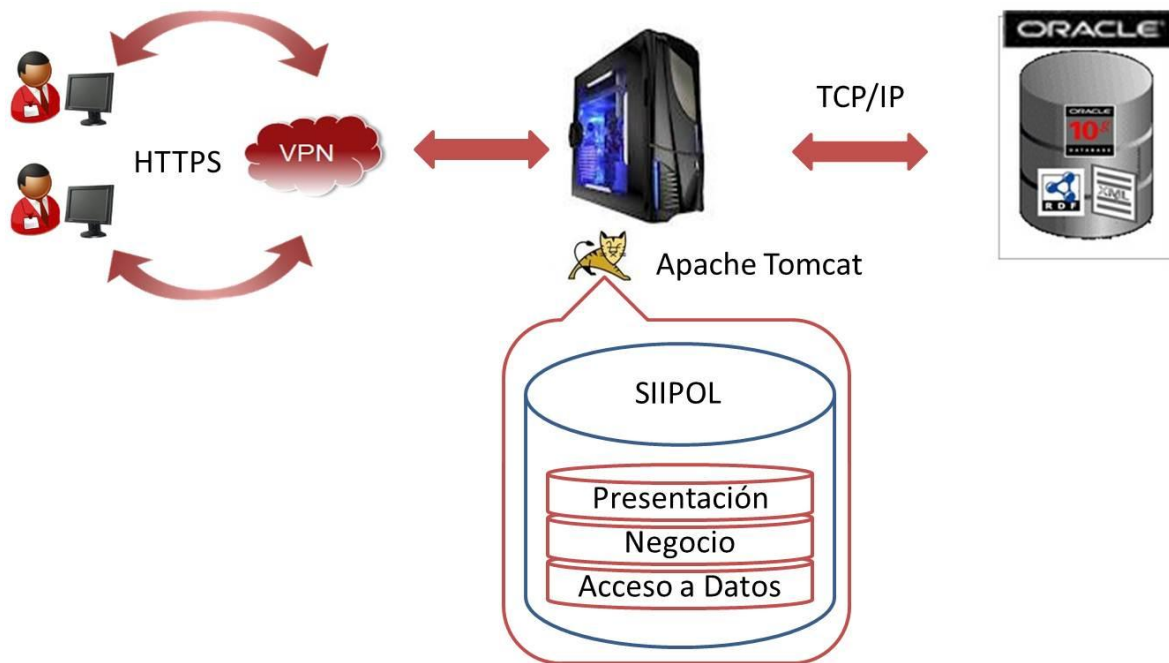


Fig. 5 Propuesta de solución.

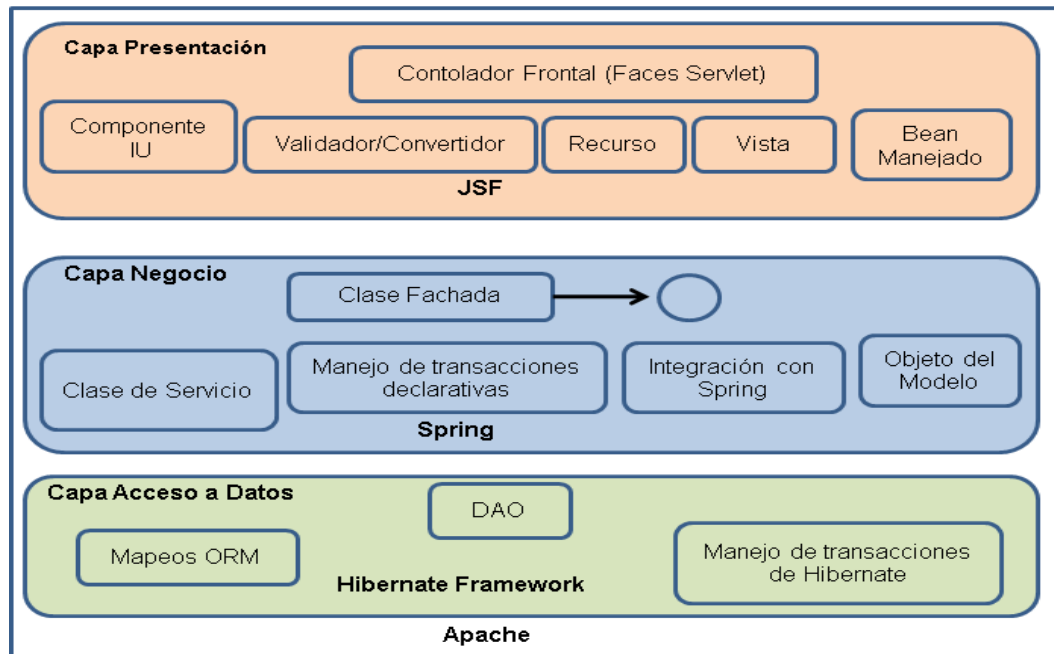


Fig. 6 Estructura lógica del SIIPOL.

1.5 Conclusiones

En el presente capítulo se elaboró el marco teórico orientado a los sistemas de gestión de información policial. Para darle cumplimiento a este objetivo específico, se estudiaron los elementos relevantes de los sistemas de gestión de información, esencialmente los de información policial. Se obtuvo como resultado, que dichos sistemas facilitan la organización y control en el manejo de la información asociada a determinado organismo policial, que se ajustan a las necesidades de cada institución y que pueden ser desarrollados para alcance nacional o internacional, como es el caso del SIIPOL o del I-24/7.

Se realizó además un estudio de los frameworks de desarrollo, metodología y herramientas empleadas para la implementación del SIIPOL, por ser el sistema para el que se implementarán los submódulos Solicitudes y Archivo Internacional del módulo Interpol.

El estudio de los frameworks arrojó como resultado, que son muy reconocidos en el desarrollo de aplicaciones JEE. Que la Inyección de Dependencia de Spring y la presencia del framework en toda la arquitectura de la aplicación, la utilización de mapeos XML para la persistencia por Hibernate y las librerías de componentes brindadas por JSF entre otras características ofrecidas por los frameworks, disminuyen considerablemente el tiempo desarrollo de aplicaciones informáticas. Permiten centrar la mayor cantidad de esfuerzo en que el sistema cumpla con los requisitos del cliente. El estudio de las características de RUP, permitió ratificarla como la metodología candidata a emplear para la implementación de los submódulos y elaboración de la documentación relacionada. A partir de este momento fue posible empezar a desarrollar la propuesta de solución.

Capítulo 2. Análisis, Diseño e Implementación de la propuesta de solución.

2.1 Introducción

En el presente capítulo se describen los principales artefactos asociados al análisis, diseño e implementación de la propuesta de solución, de los submódulos Solicitudes Internacionales y Archivo Internacional del módulo Interpol, descrita en capítulo anterior. Inicialmente se realiza una breve descripción de los casos de uso obtenidos como resultados de la Ingeniería de Requerimientos, que son el punto de partida para el análisis, diseño e implementación de los submódulos a desarrollar.

2.2 Análisis

Cuando se realiza la captura de requisitos de cualquier sistema a desarrollar, se hace con el lenguaje de los clientes, con el objetivo de establecer un acuerdo entre ambas partes de que debería hacer el sistema (12). A pesar de los esfuerzos realizados por el equipo de analistas por comprender su negocio y poder entender que es lo que se debe hacer, siempre quedan elementos sin resolver asociados a estos requisitos. Durante el análisis, se analizan los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo es comprender de manera más precisa los requisitos y estructurar el sistema en su totalidad.

Durante el desarrollo del análisis se crean una serie de artefactos que son indispensables para obtener una visión general del sistema y que son la entrada a los flujos de Diseño e Implementación. Más adelante se muestra la elaboración de los artefactos Diagramas de Clases del Análisis y la Realización de los casos de uso del Análisis

2.2.1 Entrada al Análisis

El flujo de Análisis tiene como entrada fundamental el modelo de casos de uso como resultado de la captura de requisitos, que permite que los desarrolladores y los clientes lleguen a un acuerdo sobre que debe cumplir el sistema. Los requisitos se diferencian en dos grupos: Requisitos Funcionales (RF), que son las capacidades o condiciones que debe tener el software y Requisitos No Funcionales (RNF) que son las cualidades o propiedades que este debe tener (12). De un modo más formal, comprenden

necesidades de información y control, funcionalidades del producto y comportamiento, rendimiento general del producto, diseño, restricciones de la interfaz y otras necesidades especiales (14).

Para la informatización de los procesos pertenecientes a la Dirección de Policía Internacional, específicamente los que tiene relación con los submódulos Solicitudes Internacionales y Archivo Internacional se identificaron 62 RF distribuidos en 13 casos de uso.

2.2.1.1 Requisitos Funcionales

- RF 1 Asociar Elementos (Persona, Arma, Objeto, Vehículo, Droga).
- RF 2 Validar la integridad de los datos introducidos por el usuario.
- RF 3 Mantener informado al usuario de los resultados de las operaciones.
- RF 4 Incluir una Solicitud de Información de Archivo.
- RF 5 Registrar Solicitud de Información de Archivo.
- RF 6 Ver los datos de una Solicitud de Información de Archivo.
- RF 7 Modificar los datos de una Solicitud de Información de Archivo.
- RF 8 Imprimir o Exportar una Solicitud de Información de Archivo.
- RF 9 Incluir una Respuesta de Información de Archivo.
- RF 10 Ver los datos de una Respuesta de Información de Archivo.
- RF 11 Modificar los datos de una Respuesta de Información de Archivo.
- RF 12 Imprimir o Exportar una Respuesta de Información de Archivo.
- RF 13 Registrar un Oficio de Traslado de Persona Repatriada.
- RF 14 Ver los datos de un Oficio de Traslado de Persona Repatriada.
- RF 15 Modificar los datos un Oficio de Traslado de Persona Repatriada.
- RF 16 Imprimir o Exportar un Oficio de Traslado de Persona Repatriada.
- RF 17 Incluir una Carta Rogatoria.
- RF 18 Registrar Carta Rogatoria.
- RF 19 Ver los datos de una Carta Rogatoria.
- RF 20 Modificar los datos de una Carta Rogatoria.
- RF 21 Imprimir o Exportar una Carta Rogatoria.

- RF 22 Registrar Orden de Detención Internacional.
- RF 23 Ver los datos de una Orden de Detención Internacional.
- RF 24 Modificar los datos de una Orden de Detención Internacional.
- RF 25 Imprimir o Exportar una Orden de Detención Internacional.
- RF 26 Incluir una Solicitud de Cese de Búsqueda.
- RF 27 Ver los datos de una Solicitud de Cese de Búsqueda
- RF 28 Modificar los datos una Solicitud de Cese de Búsqueda
- RF 29 Imprimir o Exportar una Solicitud de Cese de Búsqueda
- RF 30 Registrar una Solicitud de Localización de Elementos.
- RF 31 Incluir una Solicitud de Localización de Elementos.
- RF 32 Ver los datos de una Solicitud de Localización de Elementos.
- RF 33 Modificar los datos una Solicitud de Localización de Elementos.
- RF 34 Imprimir o Exportar una Solicitud de Localización de Elementos.
- RF35 Incluir una Información Documental.
- RF36 Ver los datos de una Información Documental.
- RF37 Asociar Elementos a la Información Documental (Persona, Objeto, Vehículo, Arma, Droga).
- RF38 Modificar los datos de la Información Documental.
- RF39 Imprimir o Exportar la Información Documental.
- RF40 Consultar los documentos de Interpol.
- RF41 Ordenar ascendentemente por los campos tipo de documento, fecha de realización del documento, credencial del funcionario actuante y estado del documento del listado de documentos de Interpol.
- RF42 Ordenar descendentemente por los campos tipo de documento, fecha de realización del documento, credencial del funcionario actuante y estado del documento del listado de documentos de Interpol.
- RF43 Incluir una Solicitud de Notificación Internacional.
- RF44 Registrar una Solicitud de Notificación Internacional.
- RF45 Ver los datos de una Solicitud de Notificación Internacional.
- RF46 Asociar Elementos de Información.

- RF47 Modificar los datos de una Solicitud de Notificación Internacional.
- RF48 Imprimir o Exportar una Solicitud de Notificación Internacional.
- RF49 Incluir una Solicitud de Asistencia Judicial/ Penal Internacional.
- RF50 Ver los datos de una Solicitud de Asistencia Judicial/ Penal Internacional.
- RF51 Modificar los datos de una Solicitud de Asistencia Judicial/ Penal Internacional.
- RF52 Imprimir o Exportar una Solicitud de Asistencia Judicial/ Penal Internacional.
- RF53 Incluir una Solicitud de Difusión Internacional.
- RF54 Registrar Solicitud de Difusión Internacional.
- RF55 Ver los datos de una Solicitud de Difusión Internacional.
- RF56 Modificar los datos de una Solicitud de Difusión Internacional.
- RF57 Imprimir o Exportar una Solicitud de Difusión Internacional.
- RF58 Incluir una Solicitud de Intercambio de Información.
- RF59 Registrar Solicitud de Intercambio de Información.
- RF60 Ver los datos de una Solicitud de Intercambio de Información.
- RF61 Modificar los datos de una Solicitud de Intercambio de Información.
- RF62 Imprimir o Exportar una Solicitud de Intercambio de Información.

2.2.1.2 Requisitos No Funcionales.

Los requisitos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto. Conocer cuán usable, seguro y agradable es un software, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Usabilidad

- RNF 1 Se colocará la menor cantidad de campos posibles en los formularios del sistema, solo se pondrán los que sean necesarios y suficientes.
- RNF 2 Los campos obligatorios serán señalados de forma tal que los usuarios logren identificarlos.

Acceso

- RNF 3 El sistema diferenciará los mensajes de información de los de error, mostrando mensajes de texto personalizados para cada uno de ellos.

Seguridad

- RNF 4 Un actor solo tiene acceso a una información de interpol según sus permisos en el sistema.
- RNF 5 Todo uso requiere una autenticación de los usuarios.

Fiabilidad

- RNF 6 El campo que permite validar un número de comunicación será deshabilitado después de validado el número de comunicación.

Interfaz de Usuario

- RNF 7 El sistema brindará una interfaz amigable para sus usuarios. El nivel de funcionamiento del sistema deberá corresponder al nivel medio de conocimiento informático de los usuarios.
- RNF 8 El sistema proporcionará claridad y buena organización de la información, permitiendo la interpretación correcta e inequívoca de esta.

2.2.1.3 Diagrama de Casos de Uso del Sistema.

En la Fig. 7 se muestran los casos de uso asociados a los submódulos Solicitudes y Archivo Internacional. Los casos de uso gestionar que se muestran, permiten incluir, ver o modificar una entidad del sistema.

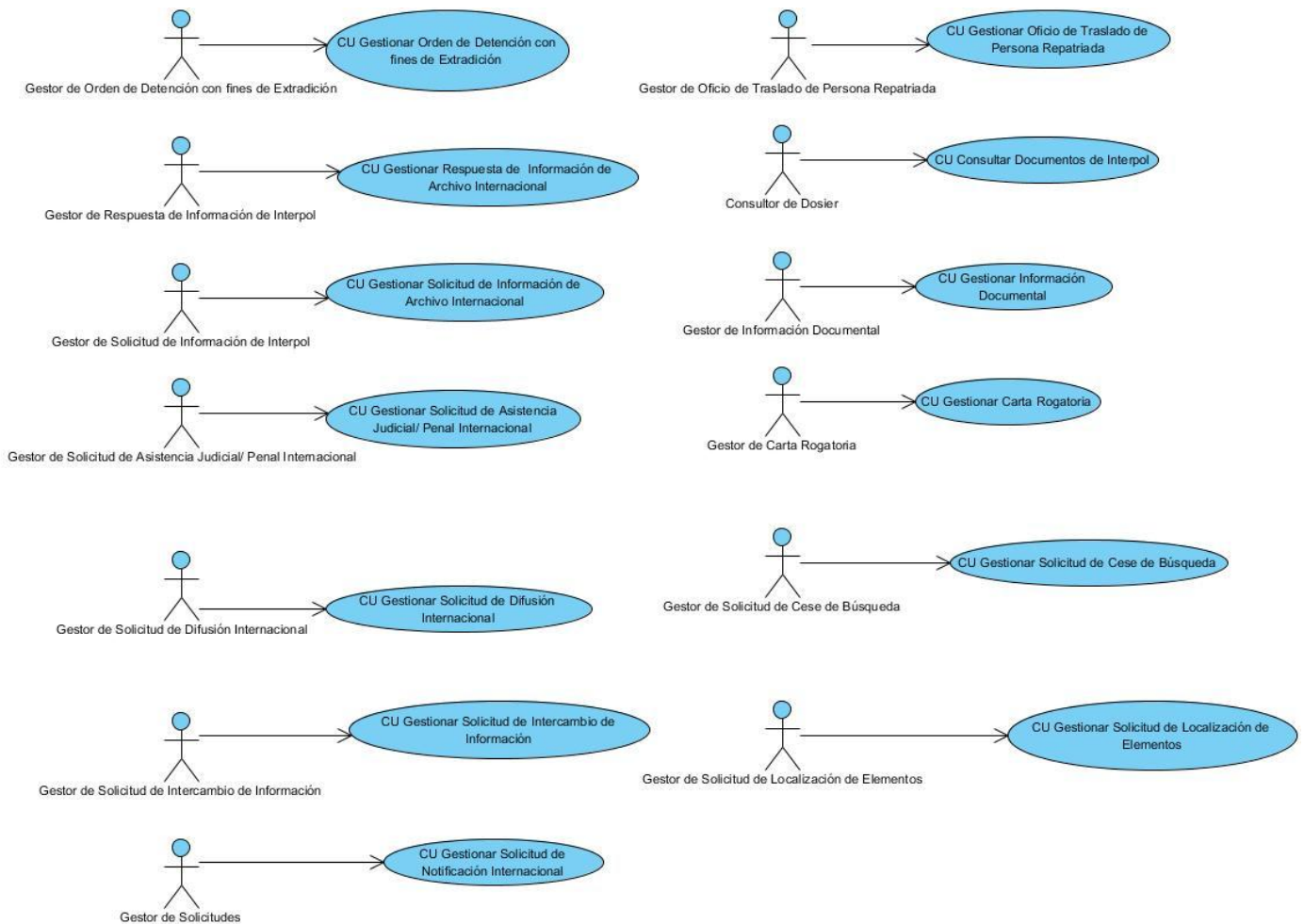


Fig. 7 Diagrama de Casos de Uso del Sistema.

2.2.2 Modelo de Análisis.

El lenguaje que se emplea en el análisis se basa en un modelo de objetos conceptual conocido como Modelo de Análisis que ayuda a refinar los requisitos y razonar sobre elementos internos del sistema (12). Es el artefacto de mayor jerarquía dentro del flujo de trabajo Análisis y está compuesto por otros artefactos como son: los paquetes de análisis, las clases de análisis y realizaciones de casos de uso. Solo se abordaron los artefactos asociados a las clases de análisis y realizaciones de caso de uso, por la importancia que tienen para el desarrollo de la solución según la metodología empleada.

2.2.3 Análisis de los casos de uso más significativos de la solución.

A continuación se muestra el análisis de los casos de uso Gestionar Solicitud de Información de Archivo Internacional, Gestionar Orden de Detención con fines de Extradición y Gestionar Solicitud de Notificación Internacional, los cuales agrupan de manera general los elementos más generales utilizados en los restantes casos de uso a implementar. Ofrecen una panorámica general del análisis de la solución.

En las tablas 1, 2 y 3 se muestran un resumen de cada uno de los casos de uso mencionados.

Caso de Uso	Gestionar Solicitud de Información de Archivo Internacional
Objetivo	Incluir, registrar, ver o modificar una Solicitud de Información de Interpol.
Actores	Gestor de Solicitud de Información de Interpol (Inicia).
Nivel	Subfunción

Resumen	<p>El caso de uso inicia cuando el actor accede a realizar alguna acción sobre una Solicitud de Información de Interpol. El actor puede incluir una nueva Solicitud de Información de Interpol, ver los detalles de una solicitud existente o modificar los datos de la misma. En caso que se seleccione la opción de incluir una nueva Solicitud de Información de Interpol el sistema permite insertar los datos de la misma como son personas, objetos, armas, vehículo, drogas. Si el actor selecciona la opción de ver una Solicitud de Información de Interpol el sistema muestra el contenido de la misma, con posibilidad de imprimir y/o exportar a formato PDF. Si el actor elige la opción de modificar una Solicitud de Información de Interpol, el sistema muestra el contenido de la misma, permitiendo la edición de los datos y una vez realizados los cambios, guarda las modificaciones. Termina el caso de uso.</p>
----------------	--

Tabla 1 Gestionar Solicitud de Información de Archivo Internacional

Por la complejidad de representar en un solo diagrama de clases del análisis, la relación entre todas las clases del análisis asociadas a un caso de uso concreto, se decidió mostrar en la *Fig. 8*, *Fig. 9* y *Fig. 10* solo las clases de análisis que se consideran necesaria para ilustrar los elementos de análisis que forman parte de la solución, para los tres casos de uso escogidos. En cada una de estas figuras, es posible observar la relación que existen entre las clases de análisis y definir sus responsabilidades desde una panorámica bastante general y que puedan en caso de ser necesario, variar para obtener un resultado.

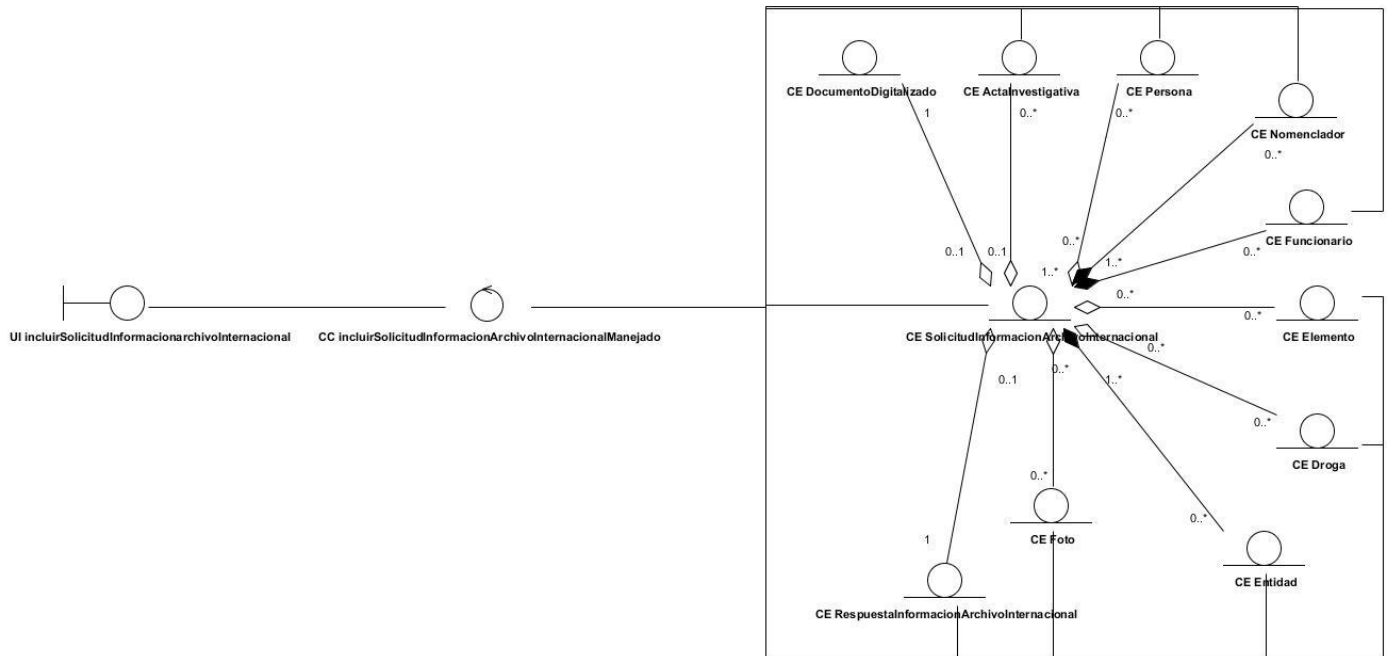


Fig. 8 Vista de análisis. Gestionar Solicitud de Información Archivo Internacional.

2.2.3.1 Realizaciones de casos de uso del Análisis.

Describen cómo se refinan los casos de uso en términos de colaboraciones dentro del Modelo de Análisis. Las realizaciones de casos de uso aíslan los cambios en los casos de uso, debido a que si cambia un caso de uso cambia su realización (12).

De manera general, el objetivo de los diagramas que se presentan a continuación es mostrar una secuencia ordenada de mensajes (interacciones) entre las clases del análisis para responder una acción realizada por el usuario. Se puede observar como las clases interfaz interactúan con las clases de control para responder la petición del usuario y está a su vez interactúa con las clases entidad para obtener lo necesario para la respuesta. Estos diagramas dan una idea bastante general de cómo los elementos de análisis interactúan entre sí de manera ordenada de acuerdo a sus responsabilidades, para obtener el resultado de un caso de uso concreto.

A continuación se muestran las vistas de los diagramas de comunicación asociados a una sola sesión de los casos de uso seleccionados.

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

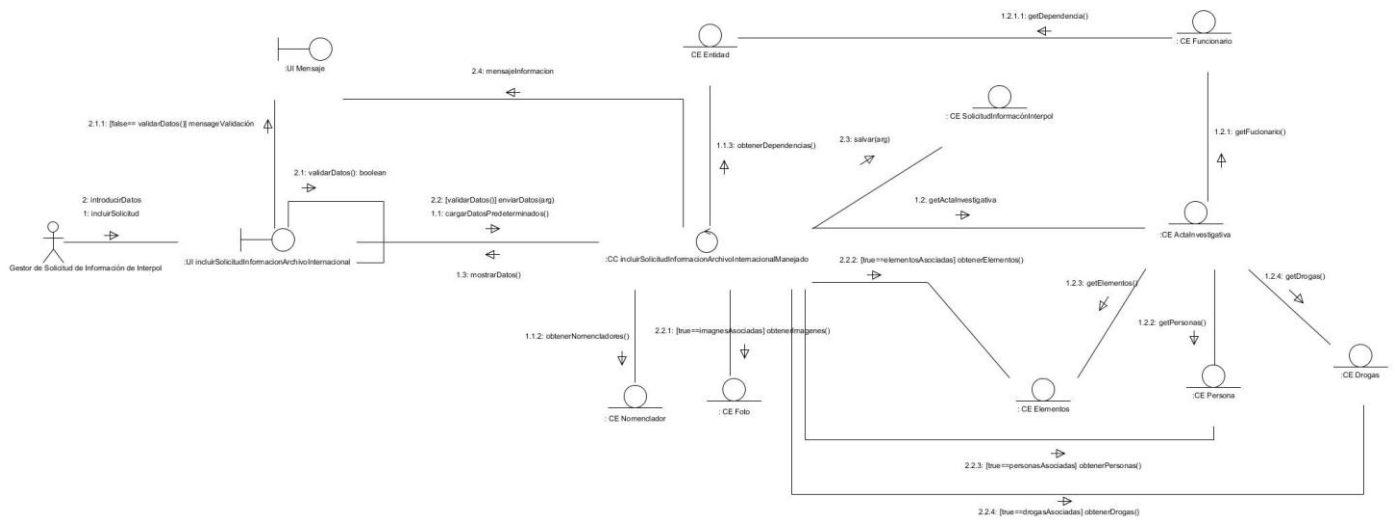


Fig. 9 Vista de colaboración. CU Gestionar Solicitud de Información Archivo Internacional.

A continuación se muestran las interfaces de usuario de las sesiones que se escogieron para realizar los diagramas de comunicación de los casos de uso seleccionados para el análisis.

The screenshot shows the SIIPOL (Sistema de Investigación e Información Policial) web application. The user is logged in as Rolando Gual Pérez, Director of the International Police Directorate, on 14/05/2012 at 16:50. The interface is titled "INCLUIR SOLICITUD DE INFORMACIÓN DE INTERPOL".

Left Menu (MENÚ PRINCIPAL):

- Alertas (1)
- Notificaciones (44)
- Recepciones (23)
- Asignaciones (108)
- Mis Asignaciones (80)
- Aprobaciones (75)
- Mis Aprobaciones (81)
- Remisiones (33)
- Borradores (2)
- Archivos
- MENÚ PRINCIPAL** (expanded):
 - Gestión de Despacho
 - Administración
 - Análisis de Información
 - Correcciones
 - Interpol
 - Investigación Penal
 - Investigación Interna
 - Captura
 - Policia Comunal
 - Acceso
 - Sustancias Químicas
 - Auditoría
 - Archivos Históricos
 - Dotación
 - Transcripciones
 - Vigilancia y Patrullaje

Main Content Area:

Tabs: Solicitud (selected), Comunicación, Elementos, Imágenes

Datos del Expediente:

Dependencia	ALBET
No. Expediente	K-12-0000-00046
Fecha Apertura	lunes, 16/04/2012 15:00
Funcionario Responsable	SYSTEM ADMINISTRATOR
Credencial Responsable	2000001
Estado	INICIADO

Entidad Receptora:

Dependencia:

Entidad:

Tipo Información Solicitada:

Entidades a Responder:

Entidad	Dirigido a	Identificación	Acciones
No existen registros			

Fundamento Legal:

Descripción:

Buttons: Incluir, Cancelar

Interfaz 1 Gestionar Solicitud de Información de Interpol

2.3 Diseño

En el diseño se adquiere una comprensión profunda de los aspectos asociados a los requisitos del sistema a desarrollar, restricciones relacionadas con el lenguaje de programación, componentes reutilizables, tecnologías de interfaz de usuario, de transacciones, de distribución. Se crea un punto de partida para las actividades de implementación. Se descompone siempre que sea posible el sistema en partes más manejables que puedan ser implementadas por diferentes equipos de desarrollo. Se reflexiona sobre el diseño utilizando una notación común (12).

2.3.1 Arquitectura

Existen múltiples definiciones de qué es la arquitectura de software, pero algunas de las más formales es la definición atribuida a Paul Clements¹⁷. *“La AS¹⁸ es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.”*

La arquitectura empieza a conformarse desde el momento que surge la primera entrevista con los clientes, pero no es hasta el Análisis y Diseño, que se pone todo el esfuerzo para definir una arquitectura que permita cumplir con los requisitos del sistema a desarrollar. Para obtener un producto es necesario coordinar personas y tecnologías; es responsabilidad de la arquitectura lograr esto.

2.3.1.1 Aspectos significativos de la arquitectura.

Como se mencionó en el capítulo anterior los submódulos a desarrollar van a formar parte del SIIPOL, y será respetada la arquitectura definida para el mismo. Este software de gran magnitud y complejidad posee una arquitectura que ha sido probada durante todo el período de desarrollo del sistema.

¹⁷ *Insigne arquitecto de la década de los '90.*

¹⁸ *Arquitectura de Software.*

No es objetivo de este epígrafe realizar una descripción detallada de los elementos arquitectónicos del software. Solo se abordarán de estos elementos los más significativos, y los utilizados para el desarrollo de los submódulos Solicitudes y Archivo Internacional.

El SIIPOL está con una arquitectura modular¹⁹ y monolítica²⁰. Es una aplicación web en capas, donde la capa de presentación es la directora del procesamiento de datos de la aplicación. Esta capa incluye componentes para validaciones de datos, navegación y otras maneras de transformación de datos de manera no persistente. Las capas de negocio y persistencia se encuentran interrelacionadas entre sí, a tal punto que forman básicamente una estructura interna dentro de la capa de aplicación (17).

Para cada módulo y submódulo de la aplicación se definió una estructura de paquetes como se muestra en la *Fig. 14*, para lograr una mayor organización de las clases. Esta estructura de paquetes se define de la siguiente manera:

Comun: Agrupa todas las clases o funcionalidades comunes para todo el módulo.

Config: Contiene las configuraciones necesarias para el correcto desempeño del módulo.

Dao: Agrupa todas las clases asociadas a la persistencia.

Facade: Agrupa las clases que son el punto de acceso a las funcionalidades de un módulo específico.

Web: Agrupa las clases para el trabajo con la presentación como los beans manejados.

Util: Contiene cualquier clase útil necesaria en el marco del módulo o submódulo al que pertenece.

¹⁹ Cada subsistema es independientes en cierto grado del resto de la aplicación (17).

²⁰ Dependiente de un conjunto central de configuraciones y componentes. Es imposible aislar completamente un módulo del sistema (17).

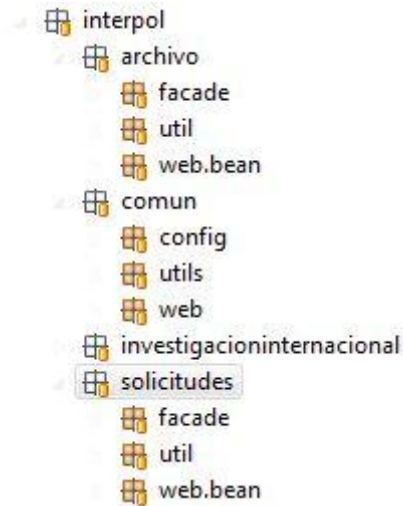


Fig. 10 Estructura de paquetes de la solución.

En el software se definen un conjunto de elementos que tienen roles particulares para desarrollar tareas específicas. Se encuentran ubicadas en diferentes partes de la aplicación y poseen un nombre distintivo. A continuación se describen algunos de estos elementos.

Convertidores: Su función es convertir los datos entrados a una manera más legible para la aplicación.

Validadores: Se encargan de validar que los datos entrados por el usuario sean válidos según las reglas de negocio definidas para la aplicación.

Beans Manejados: Manejan la lógica detrás de las páginas de presentación.

Fachadas: Las fachadas proveen el punto de acceso único a las funcionalidades de los módulos. Encapsulan la lógica de negocio del módulo o subsistema al que pertenecen, separándolos unos de otros.

Servicios: Encargados de llevar a cabo la lógica de negocio que no tenga que ver directamente con la presentación o el acceso a datos.

Daos: Encargados de la persistencia y manejo de los datos con la base de datos.

Entidades: Son el mecanismo básico de movimiento e información en la aplicación. Las entidades son leídas de la base de datos por los DAOs y luego movidas por todas las capas del sistema hasta la presentación.

Nomencladores: Tipo especial de entidad que se encarga de mantener ciertos datos relacionados con el estado de los objetos. Sus valores tienen solo un conjunto de valores predefinidos.

Útiles: Estas clases son colecciones de métodos que encapsulan lógica útil para varios lugares de la aplicación. Funcionan como librerías de funcionalidades.

Durante el diseño de la solución se tuvieron en cuenta un conjunto de clases significativas para la solución. Estas clases pertenecen a alguna de las familias de los elementos descritos anteriormente y son consideradas como las clases centrales del sistema. A continuación se realiza una caracterización de las mismas.

BaseBean: Constituye la raíz de la jerarquía de los beans de respaldo de JSF. Proporciona muchas funcionalidades necesarias para la capa de presentación (17).

ComunFacade: Constituye la raíz de jerarquía de las fachadas de todo los módulos. Proporciona funcionalidades necesarias para la capa de presentación, como funcionalidades CRUD y funcionalidades de consulta (17).

EntidadPersistenteBase: Esta clase determina la raíz de la jerarquía de Entidades que se encuentran en el paquete domain. Las clases de este estereotipo no realizan ninguna lógica y su papel se reduce a garantizar la existencia de algunos atributos obligatorios para todas las entidades del sistema (17).

Nomenclador: Representa un conjunto de valores propios de los conceptos asociados a las entidades del sistema. Unifica todos los conceptos que pueden modelarse como clasificadores (17).

DaoGenerico: Esta clase es la raíz de la jerarquía de DAOs de la aplicación. Contiene numerosas funcionalidades, métodos auxiliares de uso obligatorio que determinan la forma que deben tener las consultas. También soporta el paginado y constituye el punto único de acceso a las sesiones de Hibernate y a los datos de la aplicación (17).

2.3.1.2 Patrones

Los desarrolladores de software van creando un amplio repertorio de principios generales y de expresiones que los guían al crear el software. A unos y a otros podemos asignarles el nombre de patrones. Un patrón es una descripción de un problema y su solución que recibe un nombre y es aplicable a otros contextos con una sugerencia de usarlo en situaciones nuevas (13).

Como parte de la arquitectura definida para el SIIPOL y como elementos de la solución se utilizaron un grupo de patrones. Algunos de estos forman parte de los frameworks descritos en el capítulo anterior, utilizados en la implementación del sistema.

Composite view: Crea vistas compuestas de varias sub-vistas de forma flexible y extensible para construir vistas de páginas JSP (26). Este patrón se evidencia en las interfaces de usuario dentro de las cuales existen interfaces complejas que son divididas en varias vistas, las cuales son incluidas cuando es necesario usarlas.

Experto: Asignar una responsabilidad al experto en información: a clase que cuenta con la información necesaria para cumplir la responsabilidad (13). Un ejemplo visible dentro de la solución son los beans manejados, encargados del respaldo para la presentación. Estas clases son las únicas responsables de gestionar los datos que interactúan con la presentación.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, las conoce y recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras (13). Un ejemplo visible dentro de la solución es el momento de salvar una Entidad en la base de datos. Se conoce de la existencia de los DAOs para ese tipo de responsabilidad, por tanto la clase que desee realizarlo solo debe saber de la existencia del Dao y recurrir a él.

Alta Cohesión: La cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionada y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme (13). Las clases colaboran entre sí para producir algún comportamiento bien definido. Dentro del sistema es palpable en muchos lugares, es posible encontrarlo en cómo cada subsistema del SIIPOL, está creado

con fines específicos y se relaciona con otros para obtener un resultado bien definido. El flujo de información desde la presentación a hasta la base de datos, atravesando por diferentes elementos con responsabilidades bien definidas es otro ejemplo de alta cohesión.

Los patrones mencionados anteriormente están muy estrechamente relacionados. Cada uno tributa de alguna manera a la aplicación del otro.

Facade: Patrón de diseño estructural que provee una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea fácil de usar y la convierte en el único punto de acceso al subsistema (17). Este patrón esta visible en las fachadas existentes para cada subsistema del SIIPOL y para cada uno de los submódulos Archivo y Solicitudes Internacionales.

Dao: Concentran la lógica especial de persistencia de las entidades del dominio de la aplicación. Los DAO contienen todas las sentencias de interacción de la aplicación con la base de datos (17). Está presente dentro de la aplicación en la existencia de las clases Daos. Lo que hace transparente para el resto de la aplicación como se realiza la lectura y escritura de la información en la base de datos.

Domain Model: Este patrón descrito por Fowler²¹ consiste básicamente en usar las técnicas de diseño orientado a objetos para modelar mediante clases el dominio del problema. Un modelo de dominio consiste en una trama de objetos interconectados, una gran parte de los cuales presentan estado y comportamiento y cada uno de ellos modela un concepto del problema. El Modelo de Dominio es una vía para enfrentar con éxito negocios complejos y constituye el núcleo de la capa de negocio y de todo el sistema (17).

De manera general los patrones de diseño facilitan el desarrollo de las aplicaciones, permiten establecer un lenguaje común entre los desarrolladores, ofrecen organización y reutilización de los componentes del software.

²¹ Reconocido en el desarrollo de software, ha escrito 6 libros sobre el desarrollo de software y ha trabajado como editor de la columna de diseño de software de la revista IEEE.

2.3.2 Modelo de Diseño

El Modelo de Diseño es el principal artefacto generado durante este flujo de trabajo. Es un modelo de objetos que describe la realización física de los casos de uso centrándose en los requisitos del sistema. Sirve de abstracción a la implementación del sistema y es la entrada fundamental a las actividades de implementación. El modelo de diseño está compuesto por varios artefactos como son: las clases del diseño, los subsistemas de diseño, las realizaciones de casos de uso del diseño. Por la importancia que tiene las clases del diseño y las realizaciones de los casos de uso para ilustrar los elementos más significativos de la solución, son abordados más adelante.

2.3.2.1 Diagrama de clases del diseño.

Los diagramas de clases del diseño representan las especificaciones de las clases e interfaces software en una aplicación (12).

A continuación se muestran los diagramas de clases del diseño de los casos de uso Gestionar Solicitud de Información de Archivo Internacional, Gestionar Orden de Detención con fines de Extradición y Gestionar Solicitud de Notificación Internacional. Estos casos de uso fueron seleccionados porque reflejan de manera general la interacción de los elementos del diseño de los casos de uso de los submódulos a implementar. Con el objetivo de ofrecer claridad y entendimiento en los diagramas de clases del diseño se decidió dividirlos en partes más pequeñas (vistas). En las *Fig. 14, 15, 16 y 17* se muestran las vistas de presentación de los casos de uso mencionados anteriormente, en las que se observan las relaciones de los componentes del diseño más relevantes de la capa de presentación.

En las vistas de presentación de los casos de uso se observan los componentes con estereotipo <<*serverPage*>>, que representan las páginas que son creadas en el servidor. Estas páginas se relacionan mediante una asociación unidireccional con estereotipo <<*built*>> desde el *serverPage* (página servidora) al componente con estereotipo <<*clientPage*>>, que indica que los *serverPage* son los responsables de crear las páginas clientes que interactúan con el usuario. Los *clientPage* (página cliente) tienen una relación de composición con el componente de estereotipo <<*htmlForm*>>, que indica que las páginas clientes están compuestas por un formulario HTML para él envío de datos del servidor. Para cada uno de los formularios, existe una relación unidireccional hacia un componente con estereotipo <<*serverPage*>>

pero con nombre FacesServlet en un paquete con estereotipo `<<framework>>`. Este componente, representa al ofrecido por el Framework JSF para atender todas las peticiones realizadas al servidor. Después de realizada la petición al servidor, el FacesServlet se encarga de enviar la respuesta a las páginas servidoras para que la muestren al usuario y el proceso se representa mediante la asociación unidireccional con estereotipo `<<redirect>>`.

En los diagramas se representa además la relación bidireccional de las páginas servidoras con los componentes de estereotipo `<<beanManejado>>`. Esta relación hace referencia al componente de respaldo para cada página servidora y encargados del manejo de los datos de la misma.

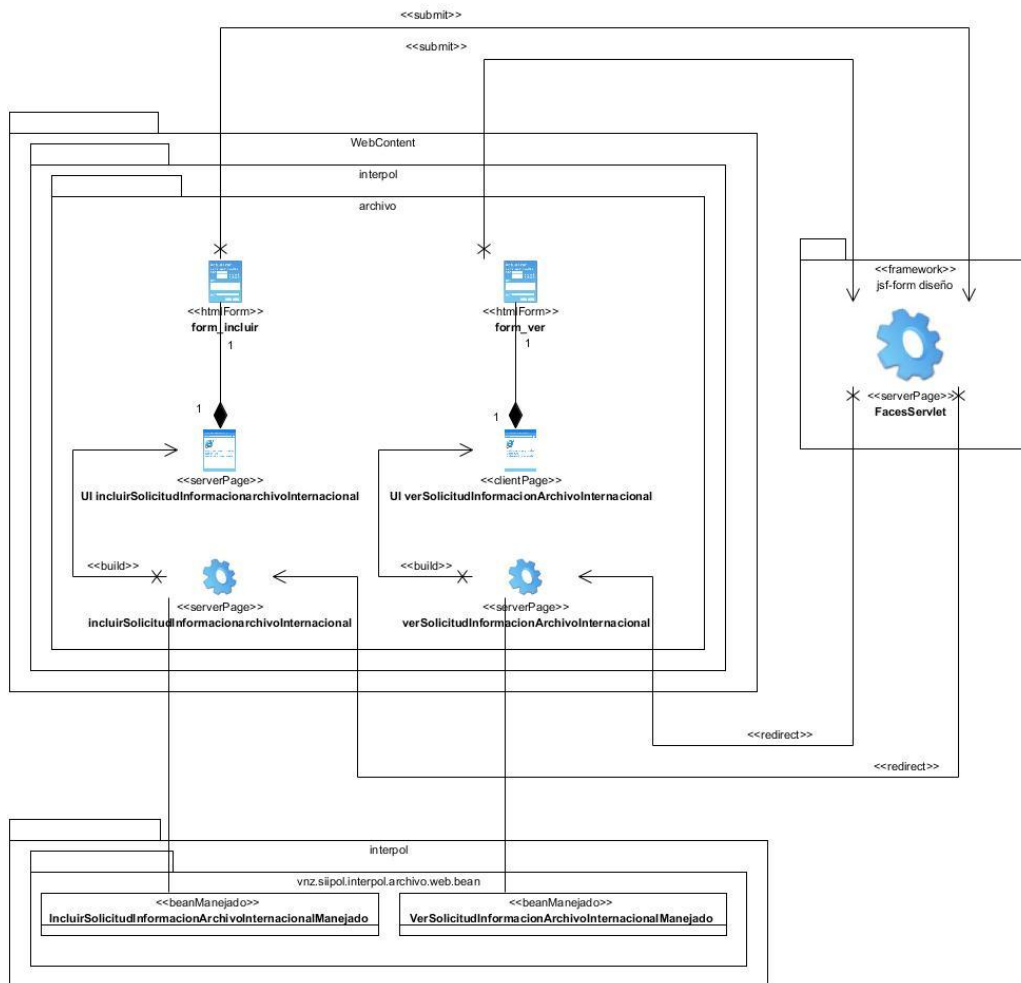


Fig. 11. Vista de Presentación. CU Gestionar Solicitud de Información de Archivo Internacional.

En las Fig. 18, Fig. 19 y la Fig. 20 se muestran los elementos más relevantes involucrados en el negocio de los casos de uso Gestionar Solicitud de Información de Archivo Internacional, Gestionar Orden de Detención con fines de Extradición y Gestionar Solicitud de Notificación Internacional. En estas se ilustran las relaciones que existen entre estos casos de uso y otros subsistemas del SIIPOL.

A continuación las vistas de negocio de los casos de uso mencionados.

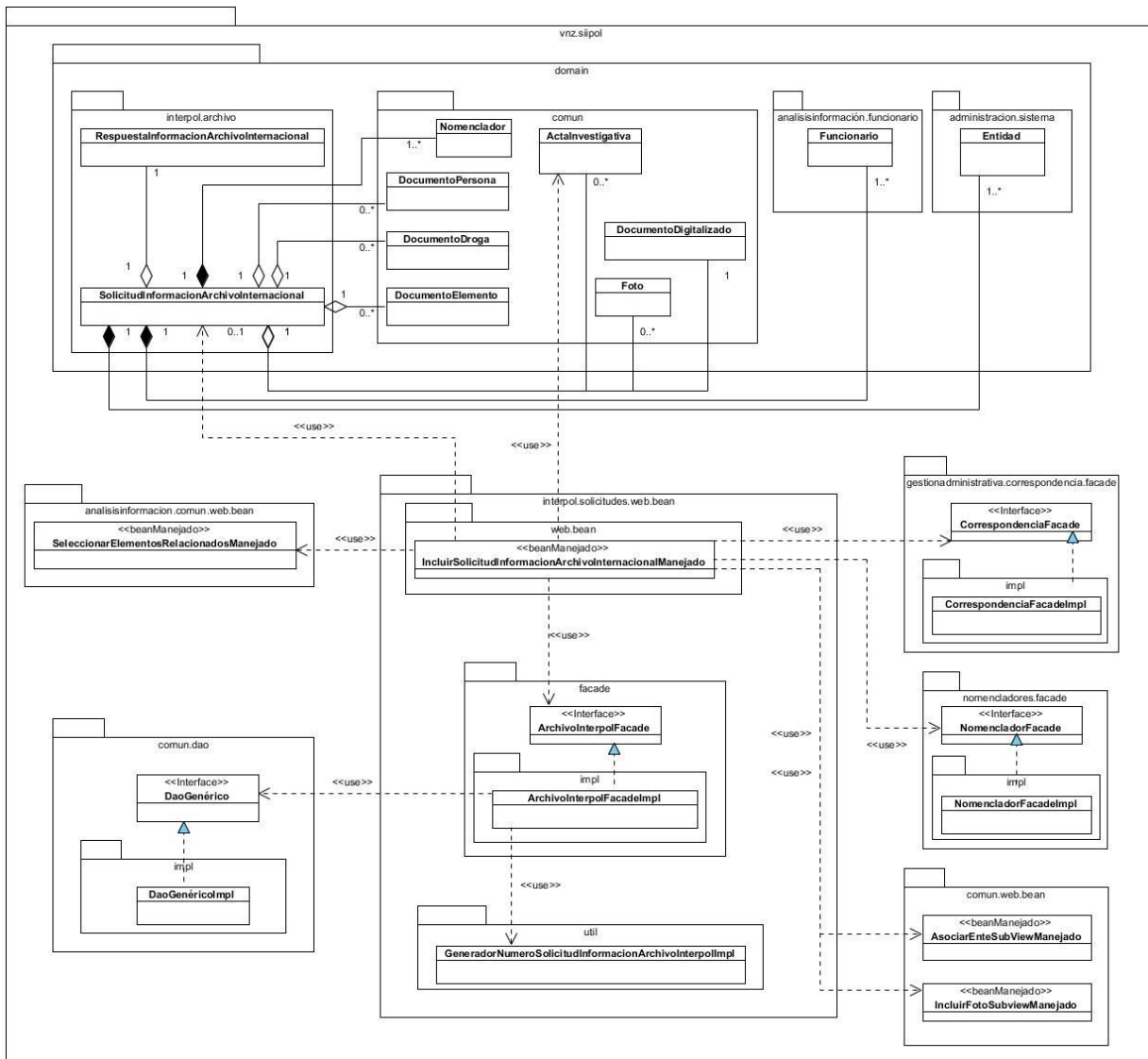


Fig. 12 Vista de Negocio. CU Gestionar Solicitud de Información de Archivo Internacional.

2.3.2.2 Clases significativas propias de la solución.

Como parte de la solución se definieron un grupo de clases que dan respuesta a las necesidades de los submódulos a desarrollar. A continuación se describen algunas de ellas:

Nombre: SolicitudInformacionArchivoInternacional	
Tipo de Clase: Entidad Persistente	
Atributo	Tipo
basamentoLegal	String
tipoInformacionSolicitadaOtro	String
tipoInformacionSolicitada	Nomenclador
Responsabilidades	
isTipoInformacionOtro	Determina si la información solicitada en el documento es del tipo "Otro".
isProcesableObligatorio	Determina si él se puede archivar o no antes de ser procesado.
isProcesableConOtroDocumento	Determina si el documento es procesable con otro documento para después saber a qué *.jsp redireccionar para incluir su respuesta.
getFormatoReporte	Devuelve el formato de la plantilla para el reporte del documento.

Nombre: GeneradorNumeroSolicitudInformacionArchivoInterpollmpl	
Tipo de Clase: Útil	
Responsabilidades	
generarNumero	Genera el Nro. de las Solicitud Información Archivo Internacional de una dependencia especificada con su formato específico
getCadenaCriterio	Convierte a String el id del criterio especificado.
getIdObjetivo	Devuelve un String con el tipo de documento al que se le genera el número.

Nombre: ArchivoInterpolFacadelmpl	
Tipo de Clase: Control	
Atributos de la Clase.	
Atributo: generadorNumeroRespuestaInformacionArchivoInterpol	
Tipo: GeneradorNumeroRespuestaInformacionArchivoInterpollmpl	
Atributo: generadorNumeroSolicitudInformacionArchivoInterpol	
Tipo: GeneradorNumeroSolicitudInformacionArchivoInterpollmpl	
Atributo: generadorNumeroInformacionDocumentallInterpollmpl	
Tipo: GeneradorNumeroInformacionDocumentallInterpollmpl	
Responsabilidades	

Nombre: salvarSolicitudInformacionArchivoInternacional
Descripción: Salva una Solicitud de Información de Archivo Internacional.
Nombre: salvarRespuestaInformacionArchivoInternacional
Descripción: Salva una Respuesta de Información Interpol.
Nombre: salvarInformacionDocumental
Descripción: Salvar una Información Documental

Nombre: RegistrarSolicitudInformacionArchivoInternacionalManejado
Tipo de Clase: Control
Atributos de la Clase.
Atributo: archivoInterpolFacade
Tipo: ArchivoInterpolFacade
Atributo: nomencladorFacade
Tipo: NomencladorFacade
Atributo: incluirFotoSubviewManejado
Tipo: IncluirFotoSubviewManejado
Atributo: relacionarElementosManejado
Tipo: RelacionarElementosManejado

Atributo: SeleccionarElementosRelacionadosManejado
Tipo: seleccionarElementosRelacionadosManejado
Atributo: solicitudInformacionArchivoInternacional
Tipo: SolicitudInformacionArchivoInternacional
Responsabilidades
Nombre: registrar
Descripción: Registra una Solicitud de Información de Archivo Internacional
Nombre: setSolicitudInformacionArchivoInternacional
Descripción: Inicializa los datos necesarios en el Bean para poder utilizarlo.
Nombre: activarDependencias
Descripción: Pone los beans especificados activos para la sesión.
Nombre: verArchivoDigital
Descripción: Permite visualizar el archivo digital asociado al documento.
Nombre: cancelar
Descripción: Cancela la operación que se está realizando.

2.3.2.3 Realización casos de uso del diseño.

La secuencia de acciones en un caso de uso tiene inicio cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje. Un objeto de diseño recibe el mensaje del actor y llama a algún otro objeto, y de esa manera los objetos implicados interactúan para realizar y llevar a cabo el caso de uso. La

mejor manera de representar estas interacciones es a través de un diagrama de secuencia, pues lo que se desea representar es una secuencia ordenada de interacciones en el tiempo (12).

Una realización de caso de uso del diseño, describe como se realiza un caso de uso específico, y como se ejecuta en términos de clases del diseño y sus objetos. Es la entrada principal para la implementación del caso de uso. Es de alguna manera el comportamiento dinámico de los objetos del sistema para obtener un resultado.

Para los casos de uso del SIIPOL resulta bastante complejo realizar este tipo de diagramas por la gran interrelación existente entre los objetos. Por tal motivo y al ser conocida la importancia de los mismos, se desarrolló una alternativa. Los “Diagramas de Contrato entre Paquetes” son un artefacto creado por el proyecto con características similares a los Diagramas de Secuencia propuestos por RUP, solo que el envío de los mensajes se realiza entre los subsistemas implicados en la solución de un caso de uso.

Estos diagramas tienen como características específicas:

- La línea de vida denota subsistemas en vez de objetos del diseño.
- Cada subsistema identificado debería tener al menos una línea de vida que lo denote en el diagrama de secuencia.

En las *Fig. 21*, *Fig. 22* y *Fig. 23* se muestran los diagramas de “Contrato entre Paquetes” asociados a una sesión de los casos de uso Gestionar Solicitud de Información de Archivo Internacional, Gestionar Orden de Detención con fines de Extradición y Gestionar Solicitud de Notificación Internacional.

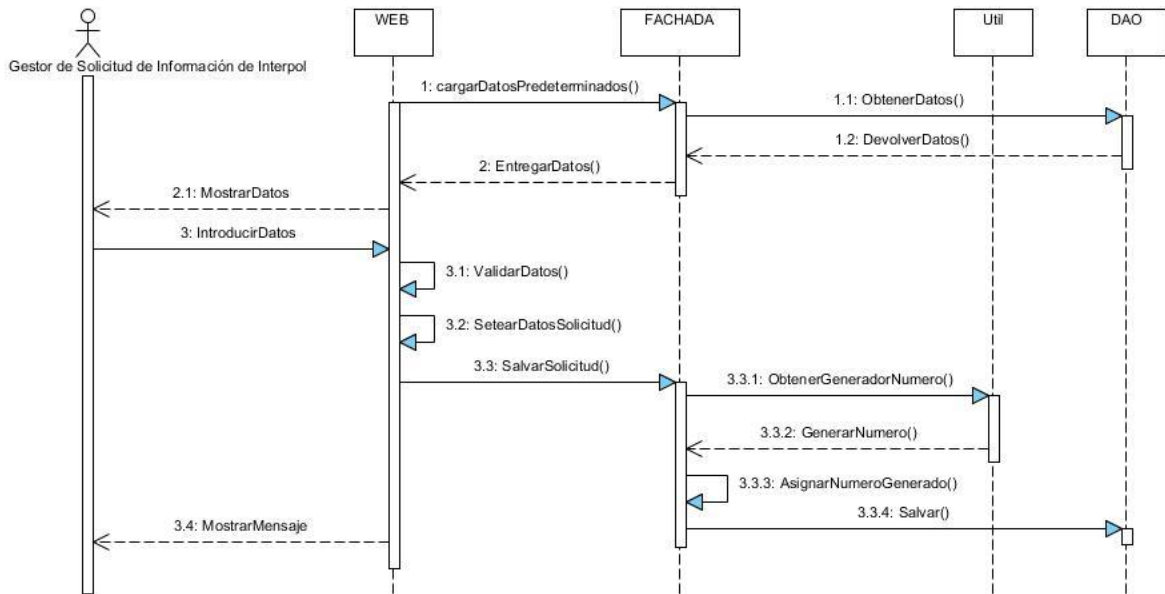


Fig. 13 Diagrama de Secuencia. CU Gestionar Solicitud de Información de Archivo Internacional.

2.3.3 Modelo de datos

La persistencia de la información en bases de datos nunca ha dejado de ser una tarea difícil. De su correcto diseño y elaboración depende mucho la flexibilidad del sistema. Este artefacto describe las representaciones lógicas y físicas de datos persistentes utilizados por la aplicación. Un modelo de datos es específicamente necesario donde la estructura de datos persistente no se puede derivar automáticamente ni mecánicamente a partir de la estructura de clases persistentes del modelo de diseño. Se utiliza para definir la correlación entre las clases de diseño persistentes y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes. El modelo de datos muestra las relaciones creadas inicialmente desde las clases del diseño persistente y sus relaciones.

2.3.3.1 Diagramas de clases del dominio.

En el SIIPOL existe una jerarquía de clases asociada a los documentos que se elaboran en el CICPC, y que responde a las necesidades del proceso de negocio en general. Esta jerarquía está constituida por la clase Documento, que permite agrupar todas las características de todos los documentos elaborados en el CICPC y demás elementos necesarios para su trabajo en el sistema. De la entidad Documento heredan

DocumentoSalida, que recoge el comportamiento específico de los documentos que viajan de una dependencia a otra del CICPC como parte de una investigación y que permite su control en el sistema. Existe además DocumentoSustancial que hereda de Documento y especializa el comportamiento de los documentos elaborados que sustancian un expediente, y que no viajan entre los despachos del CICPC. En la Dirección de Policía Internacional, se elaboran un grupo de documentos que reflejan las labores de investigación que en esta dirección se desarrollan, y que fueron debidamente distribuidos por los dos submódulos a implementar. Los documentos identificados y que posteriormente se convirtieron en clases del dominio se clasifican en DocumentoSalida o DocumentoSustancial en dependencia de sus características. En las Fig. 25 se muestran resaltadas de color azul las clases del dominio para el módulo Archivo y en la Fig. 26 para el módulo Solicitudes, además de sus relaciones.

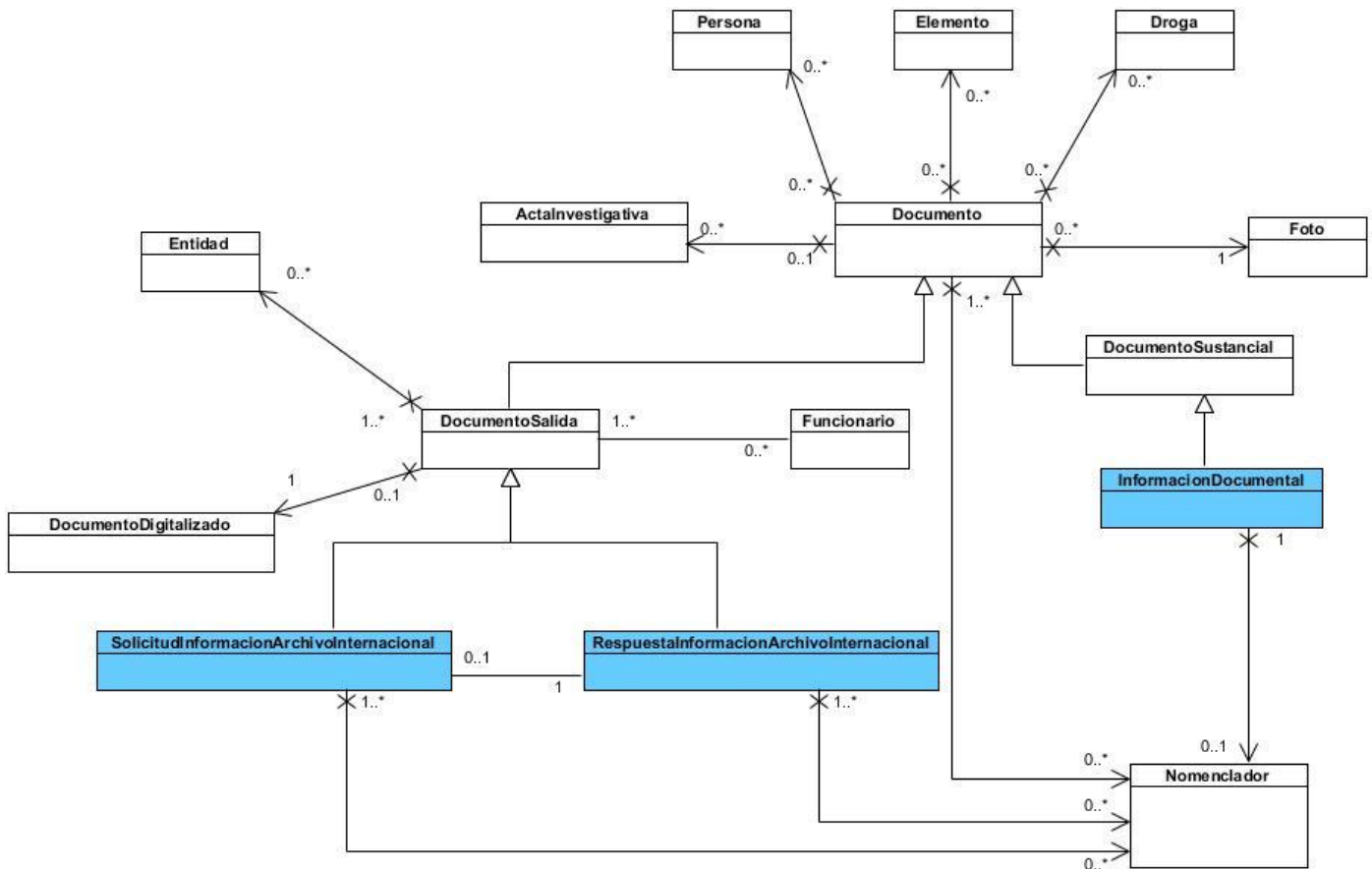


Fig. 14 Diagrama de Clases del Dominio Módulo Archivo

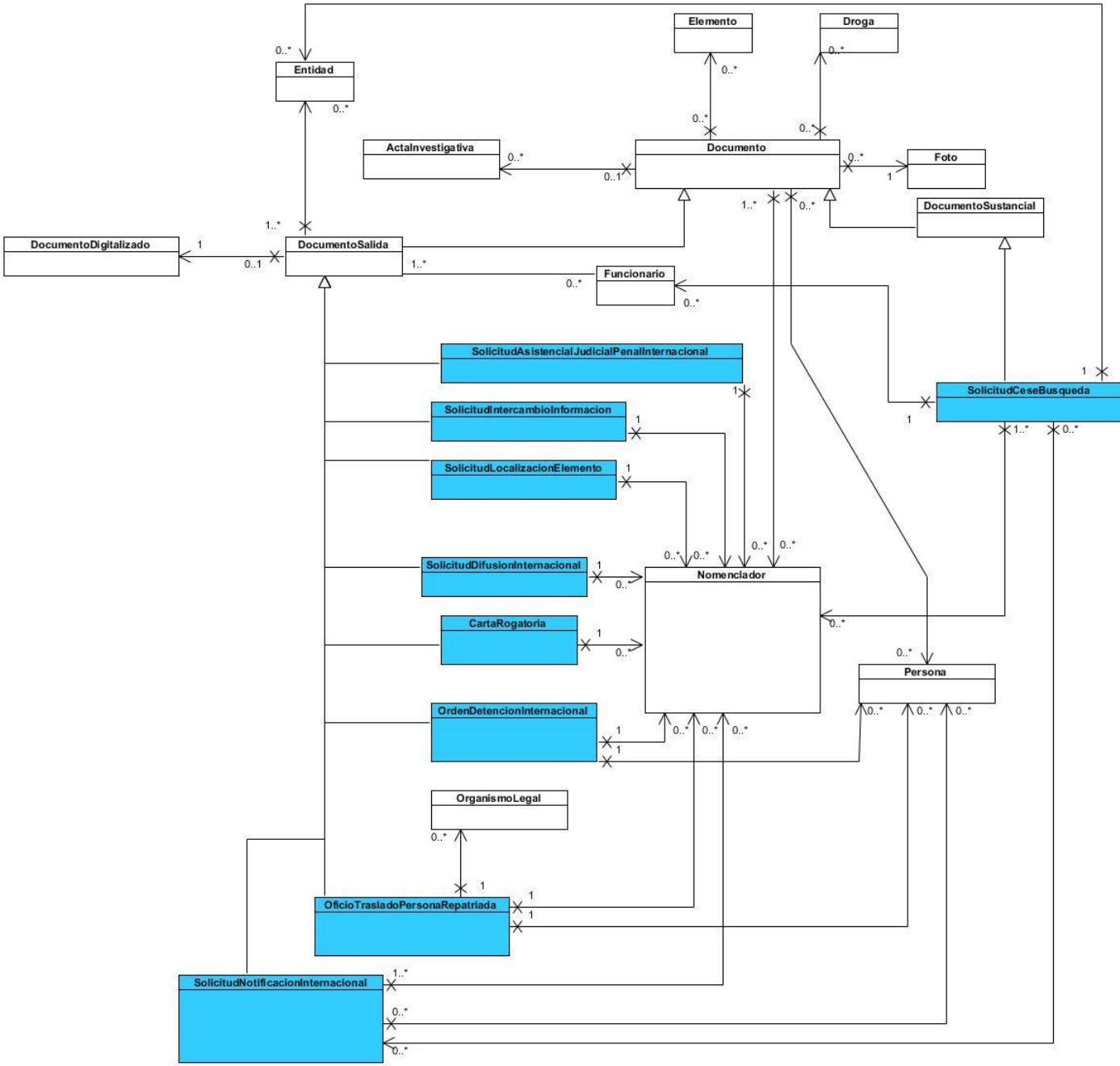


Fig. 15 Diagrama de Clases del Dominio Módulo Solicitudes.

2.3.3.2 *Modelo Relacional*

Por la necesidad de ofrecer claridad en el diagrama del modelo relacional (Fig. 27) fue necesario ocultar los campos de cada una de las tablas y mostrar solo el nombre, en los Anexos 1 y 2 se muestran los campos de cada una estas tablas.

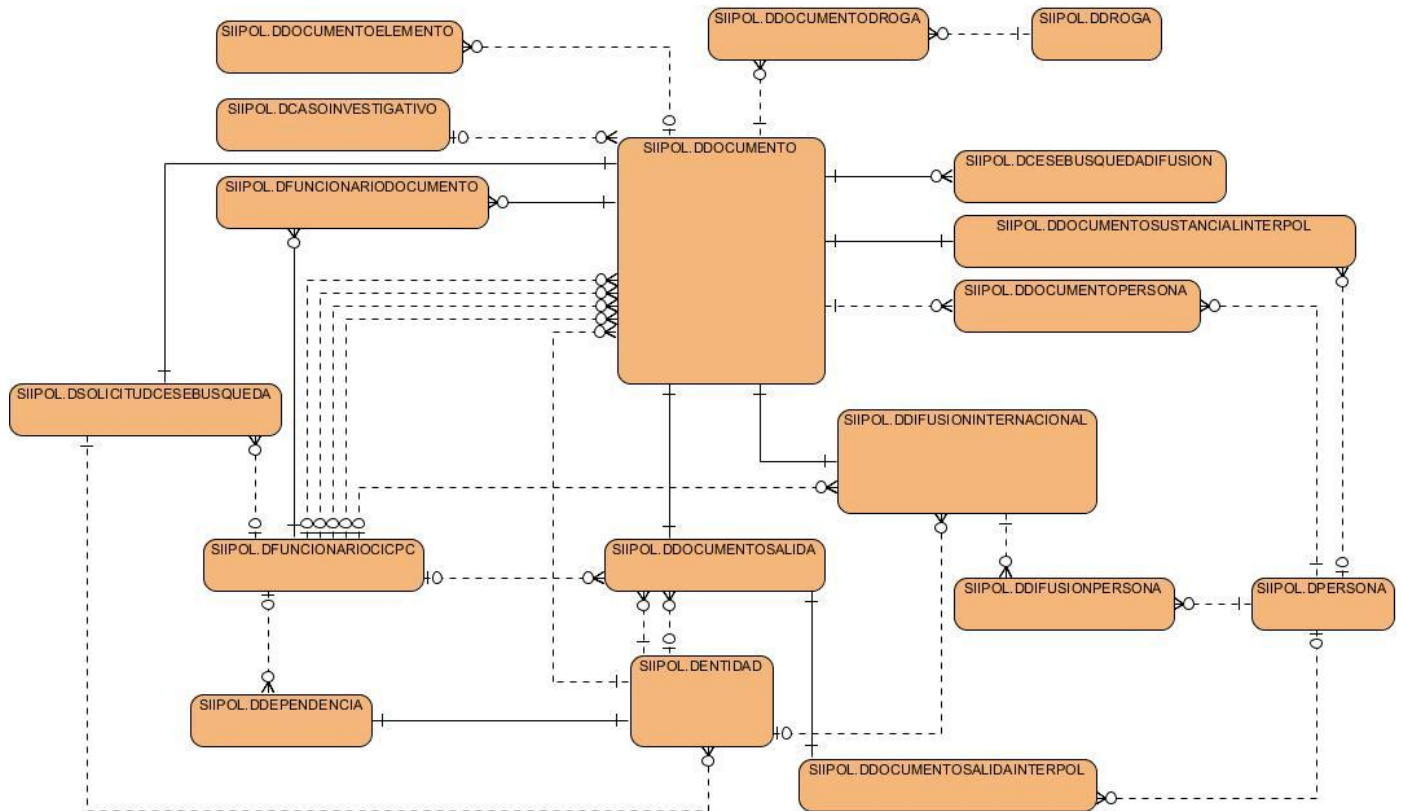


Fig. 16 Modelo Relacional.

2.4 Implementación

En implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes²² (12). En la Implementación se desarrolla la arquitectura y el sistema como un todo. Se implementan las clases y subsistemas encontrados durante el diseño. Las clases particularmente se implementan como componentes de fichero que contienen código fuente.

2.4.1 Modelo de Implementación.

El modelo de implementación describe como los elementos del modelo de diseño, como las clases, se implementan en términos de componentes. El modelo de implementación describe también cómo se organizan los componentes de acuerdo a los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros (12).

2.4.1.1 Subsistema de implementación.

Los subsistemas de implementación proporcionan una forma de organizar los artefactos del modelo de implementación en piezas más manejables. Un subsistema puede estar formado por componentes, interfaces y otros subsistemas (recursivamente) (12). Es importante entender que un subsistema de implementación se manifiesta a través de un “mecanismo de empaquetamiento” concreto en un entorno de implementación determinado (12), en el caso de Java como un paquete. Los subsistemas de implementación deben seguir una traza uno a uno de sus subsistemas de diseño correspondientes.

En la *Fig. 28* se muestra el subsistema de implementación asociado a la los submódulos Solicitudes y Archivo Internacional, con los subsistemas más importantes con los que se relacionan en el SIIPOL.

²² Según RUP es el encapsulamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño.

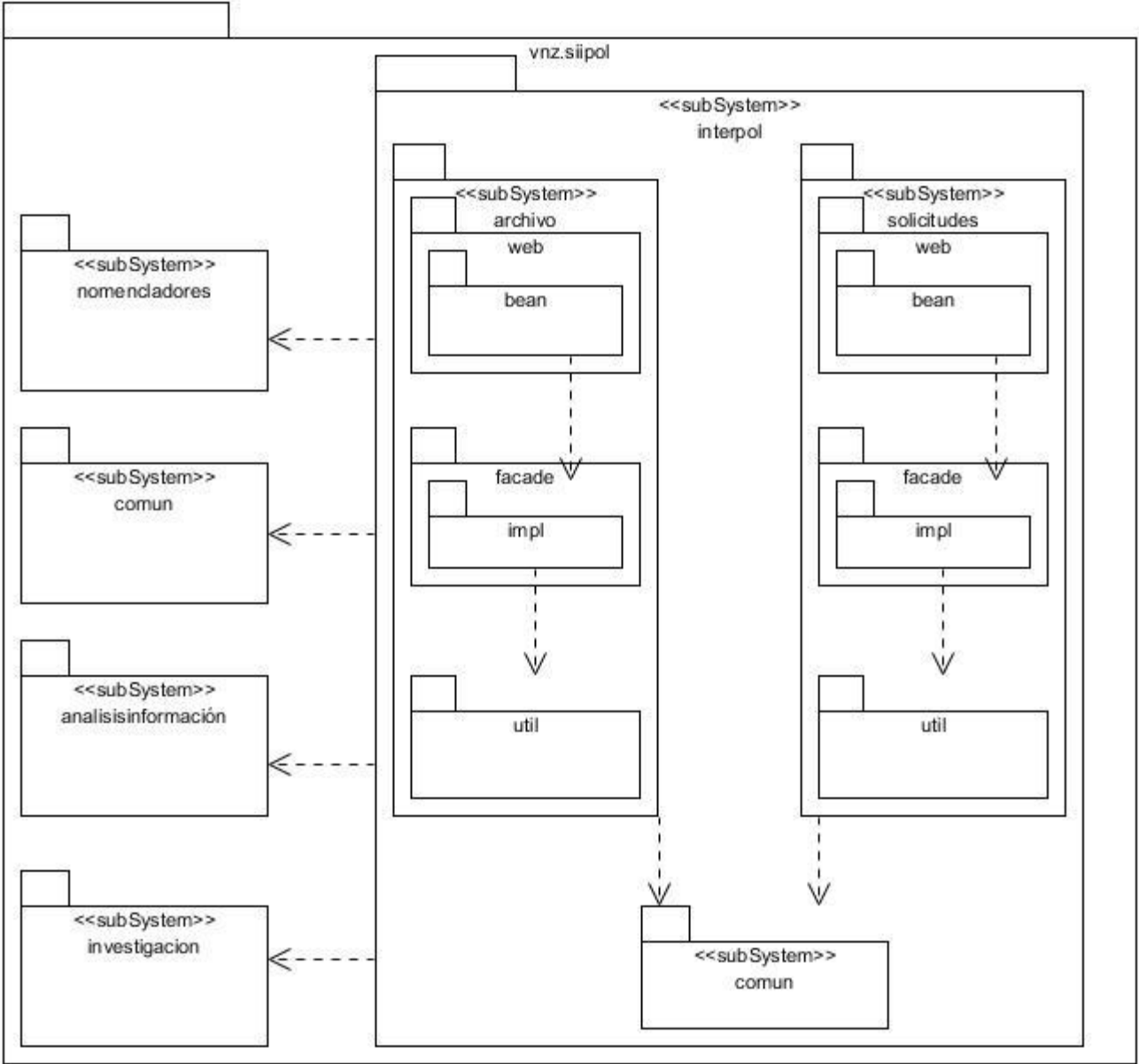


Fig. 17 Subsistema de Implementación.

2.4.1.2 Diagrama de Componentes

Un diagrama de componentes muestra la estructura de los componentes, incluyendo clasificadores que especifican componentes, y artefactos que los implementan. Muestran los subsistemas de implementación organizados en capas, los elementos de implementación significativos, y sus relaciones.

A continuación se muestran los diagramas de componentes de los submódulos Archivo y Solicitudes Internacionales dividido en tres vistas, para ofrecer claridad y entendimiento.

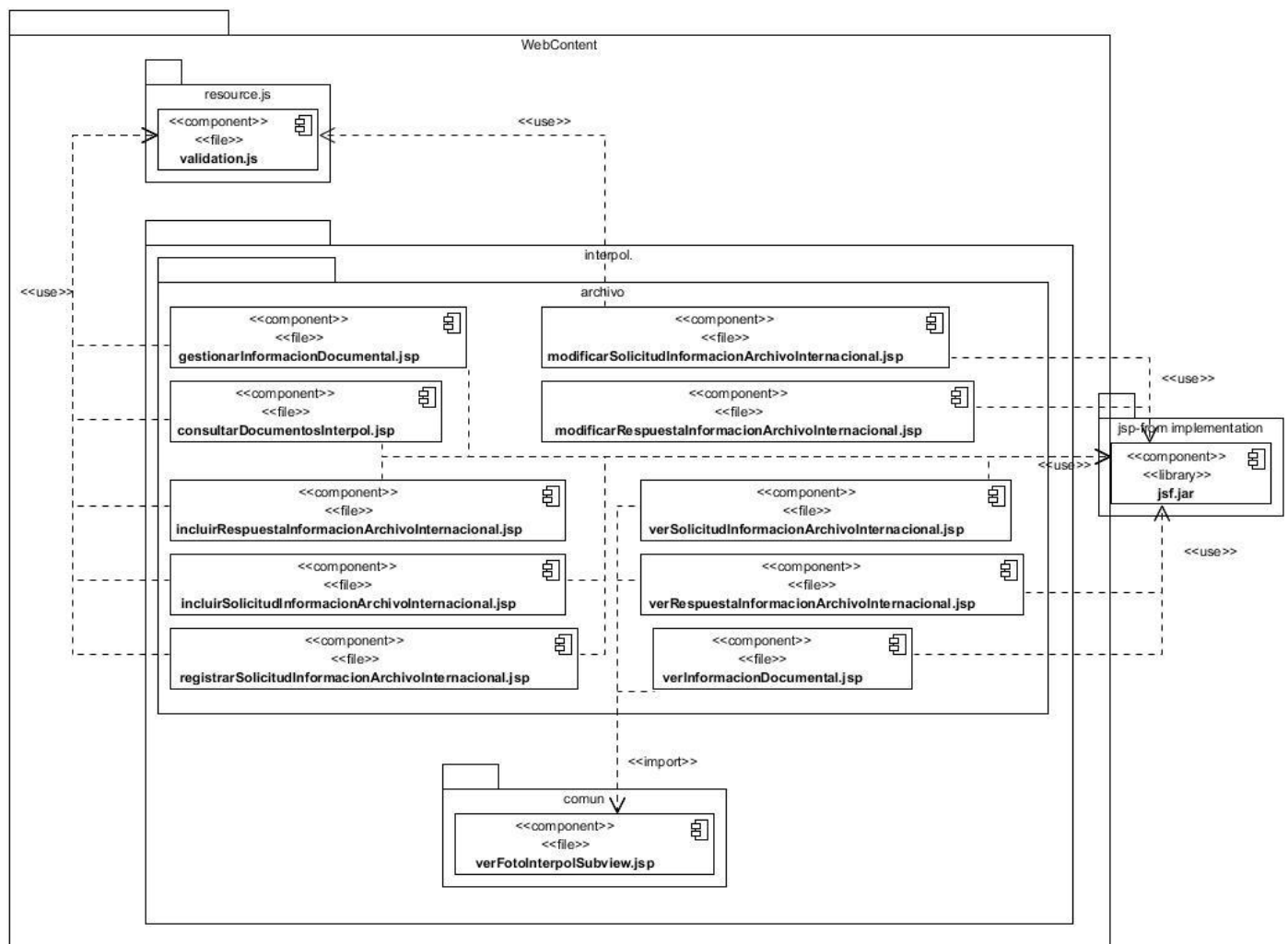


Fig. 18 Vista Presentación módulo Archivo. Páginas jsp

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

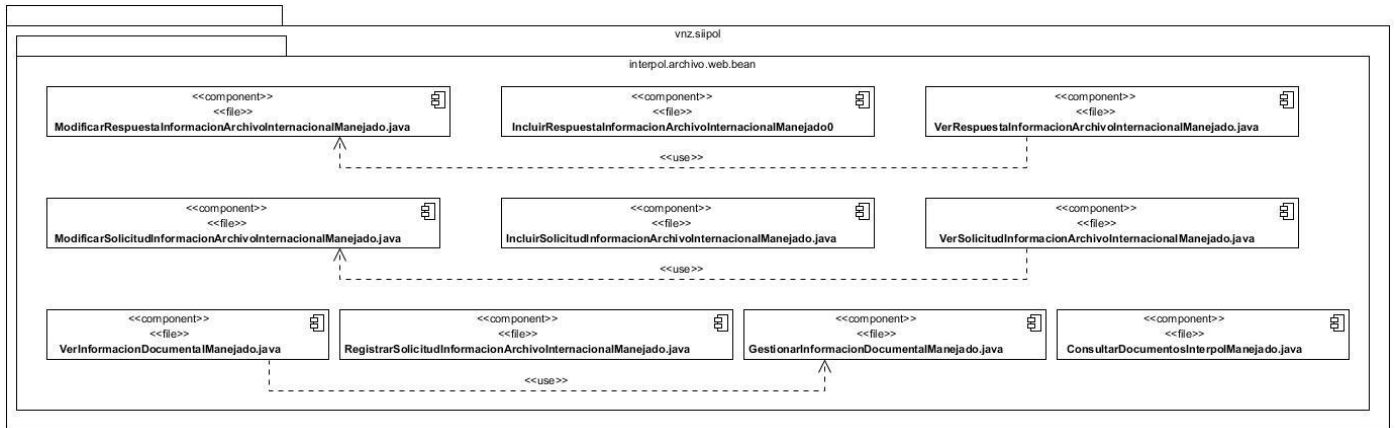


Fig. 19 Vista Presentación módulo Archivo. Beans manejados

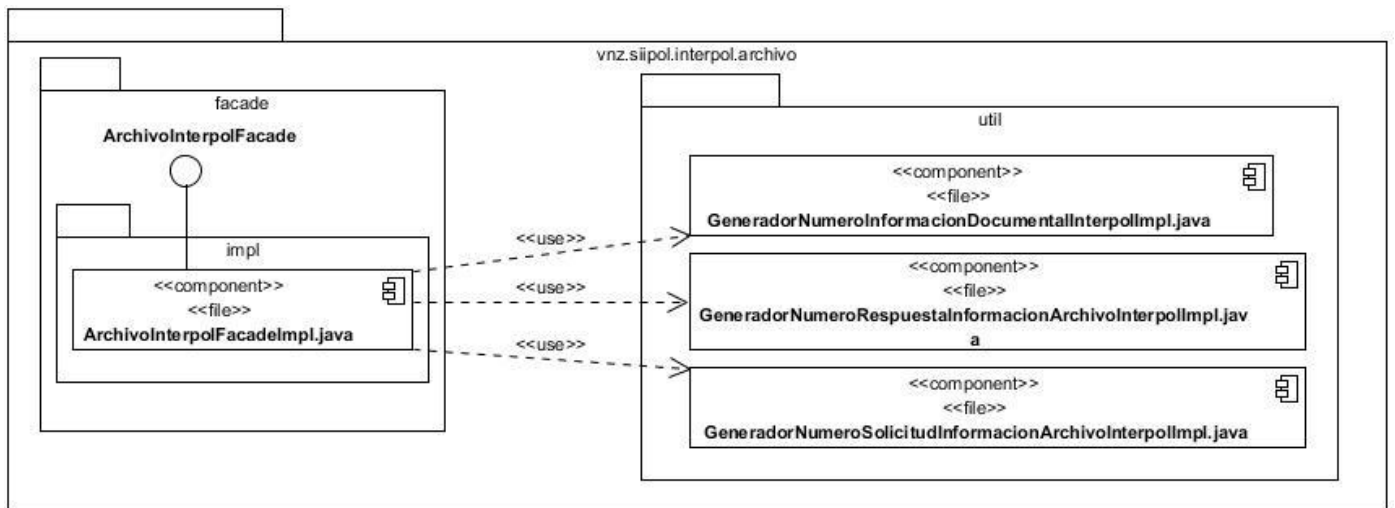


Fig. 20 Vista Negocio módulo Archivo

2.5 Conclusiones

En el presente capítulo se dio cumplimiento a los objetivos de análisis, diseño e implementación de los submódulos Solicitudes y Archivo Internacional. Para lograr el cumplimiento de los objetivos trazados fue decisivo estudiar los casos de uso asociados a los submódulos a implementar. Se elaboraron los Modelos de Análisis, Diseño, Datos e Implementación lo que permitió tomar decisiones antes de implementar. Al realizar un estudio de la arquitectura definida para el SIIPOL, se conocieron los patrones de diseño más utilizados por el equipo de desarrollo. El uso de los patrones como Facade, Experto y Dao permiten la existencia de un lenguaje común entre todos los desarrolladores del proyecto SIIPOL, lo que hace posible que los desarrolladores puedan trabajar en cualquier módulo del sistema en caso de ser necesario. De manera general se implementó e integró la solución al SIIPOL, lo que permite dar comienzo a la verificación del correcto funcionamiento de lo implementado.

Capítulo 3. Validación de la implementación de la propuesta de solución.

3.1 Introducción

Las pruebas de software son un elemento crítico para la garantía de la calidad²³ de software y representan una revisión final de las especificaciones, del diseño y de la codificación (14). Este capítulo está enfocado en la verificación del correcto funcionamiento de la solución implementada y se tratan los conceptos referentes a métodos de pruebas y niveles de pruebas. Se realiza una descripción de las pruebas realizadas a los submódulos implementados, lo que permite tener un espectro mucho más amplio de la seriedad e importancia de las pruebas de software. Todos los elementos de prueba de software que se describen en el capítulo, contribuyen a entregar un producto con calidad.

3.2 Prueba

Los desarrolladores de software intentan construir un producto a partir de un concepto abstracto, que posteriormente se convierte en un elemento tangible. Durante esta transformación que experimenta lo que el cliente desea, se introducen un gran número de errores que atentan contra la calidad del software desarrollado o en desarrollo. Por tal motivo y en busca de satisfacer las necesidades del cliente, es necesario probar el sistema en una variedad de niveles diferentes, en un esfuerzo por descubrir errores que surgen cuando los elementos del sistema como las clases o subsistemas se comunican entre sí. El objetivo de las pruebas, expresado de forma sencilla, es encontrar el mayor número posible de errores con una cantidad de tiempo razonable de esfuerzo, en un lapso de tiempo realista y manejar los resultados de cada prueba sistemáticamente. Se verifica el resultado de la implementación probando cada construcción antes de que las versiones finales sean entregadas a terceros. Según Glenford Myers²⁴ la prueba es el proceso de ejecución de un programa con la intención de descubrir un error, un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces y que una prueba tiene éxito si descubre un error no detectado hasta entonces (12) (14). De manera general, las

²³ Según Pressman: “Concordancia del software producido con los requisitos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requisitos implícitos no establecidos formalmente, que desea el usuario”.

²⁴ Glenford Myers (12 de diciembre de 1946) es un científico informático estadounidense con una gran cantidad de publicaciones de documentos técnicos y ocho libros, entre los que destaca “The art of Software Testing”.

pruebas no aseguran la ausencia de errores en el sistema desarrollado, solo permiten tener una idea de que tan correcto es el funcionamiento del producto. Las pruebas de software deben ayudar a detectar errores existentes y no a introducir otros.

3.3 Métodos de Prueba

Con el objetivo de detectar la mayor cantidad de errores en un producto de software se han definido dos métodos de pruebas que responden a perspectivas diferentes. El primero, responde a la perspectiva de que una vez conocida la funcionalidad específica para la que fue diseñado el producto, se puedan llevar a cabo pruebas que demuestren que cada funcionalidad es operativa y se conoce como Prueba de Caja Negra. El segundo, responde a la perspectiva de asegurar que todas las piezas internas del software se ajustan correctamente y se conoce como Prueba de Caja Blanca o Caja de Cristal (14). Los métodos de prueba de software dictan de qué manera probar, pero no qué es lo que se debe probar.

3.3.1 Pruebas de Caja Blanca

Las pruebas de Caja Blanca, denominada a veces prueba de Caja de Cristal es un método de diseño que tiene como objetivo determinar errores lógicos, es decir, determinar que tan bien está el código de un punto de vista lógico, para obtener el resultado esperado. Permite verificar si en el momento de implementar una determinada funcionalidad, se tuvo en cuenta las diferentes situaciones a las que puede estar expuesta a la hora de ser utilizada. Al emplear las Pruebas de Caja Blanca el ingeniero de software podrá garantizar que todas las rutas independientes dentro del módulo se han ejercitado por lo menos una vez, que se ejerciten los lados verdaderos y falsos de todas las decisiones lógicas, que se ejecuten todos los bucles en sus límites operacionales y que se ejerciten las estructuras de datos internos para asegurar su validez (14).

3.3.2 Pruebas de Caja Negra

Las pruebas de Caja Negra también conocidas como pruebas de Comportamiento, se centran en los requisitos funcionales del software, con el objetivo de demostrar que cada función es completamente operativa. Las pruebas de Caja Negra permiten detectar errores de funciones incorrectas o ausentes,

errores de interfaz, errores de rendimiento, errores en estructuras de datos o en accesos a bases de datos externas (14).

3.4 Niveles de Prueba

Un producto de software están compuesto por piezas (componentes) más pequeñas que interactúan entre sí, y que son diseñadas e implementadas para realizar una tarea específica dentro del producto de software. Si un componente falla todo lo relacionado con él también lo hará, sin embargo que un componente esté correctamente implementado no es garantía de que su interacción con otros elementos del sistema se realizará correctamente. Por tal motivo es necesario probar los componentes individuales del sistema, los componentes interactuando entre sí y el sistema como un todo, para poder valorar los resultados obtenidos y trazar nuevas metas. En pocas palabras las pruebas de software deben ser ejecutadas a diferentes niveles, para determinar la robustez del software. A continuación se describen los niveles de prueba realizados a la aplicación.

3.4.1 Pruebas de Unidad

Las pruebas de Unidad centran el proceso de verificación en la menor unidad del diseño del software: el componente software o módulo (14). En este nivel de prueba llevado a cabo solo por el desarrollador, se hace un uso continuo de las pruebas de Caja Blanca. Durante la aplicación de este nivel de prueba se realizó un debugging de los elementos asociados a un caso de uso concreto. Se examinó que los datos introducidos llegaran correctamente a el bean de respaldo de la página *.jsp. Se verificó en la medida de lo posible que los métodos implementados en los servicios se ejecutaban correctamente en dependencia de los datos introducidos y que se tuviesen en cuenta las posibles condicionales para diferentes escenarios. Se verificó que no existiera pérdida de la referencia de las colecciones de datos almacenados en las estructuras de datos locales, verificando la integridad de los mismos durante la ejecución del algoritmo. Los resultados arrojados como parte de esta prueba, fue la mejora del funcionamiento y calidad de lo implementando, gracias a la aplicación continua de estas pruebas cada vez que se terminaba de implementar un caso de uso se agregaba como un nuevo elemento al sistema.

3.4.2 Pruebas de Integración

Las pruebas de Integración tienen como objetivo probar el correcto funcionamiento de la interacción entre los módulos. Lo que de una manera aislada funcionó correctamente, no tiene necesariamente que hacerlo cuando interactúa con otros módulos del sistema. Los datos se pueden perder, las funciones asociadas a un módulo pueden tener un efecto contrario al esperado, cuando se combinan con otras de un módulo diferente. Las pruebas de Integración es una técnica sistemática para construir un software lo más correcto posible (14). Para desarrollar estas pruebas el desarrollador probó los componentes implementados y su integración con otros componentes que forman parte de otros módulos del SIIPOL. Algunos de los puntos de integración que fueron sometidos a pruebas son las funcionalidades de asociar elementos del módulo Análisis de Información, el componente de asociar imágenes del módulo de Investigación Criminalística.

Los submódulos Solicitudes y Archivo Internacional poseen además otros puntos de integración con el SIIPOL como son: Gestión Administrativa, donde era necesario realizar una configuración a nivel de base de datos para su correcta integración con el sistema y con la Agenda de Trabajo.

3.4.3 Pruebas de Sistema

Las pruebas de Sistema tiene como objetivo principal probar el sistema como un todo, cada uno de los módulos del sistema son puestos a prueba. Las pruebas de Sistema están constituidas por una serie de pruebas diferentes, cuyo propósito primordial es ejercitar profundamente el sistema en su totalidad y determinar si todos los componentes del software funcionan correctamente en su conjunto.

Caso de Prueba

Un caso de prueba específica como probar un caso de uso o un escenario específico de un caso de uso. Incluye la verificación del resultado de la interacción entre los actores y el sistema, que se satisfacen las precondiciones y poscondiciones especificadas por el caso de uso y sigue las acciones especificadas en el mismo (12). En la *Fig. 35* se muestra una porción de un caso de prueba para pruebas de Caja Negra en las pruebas de Sistema, asociado al caso de uso Gestionar Respuesta de Información de Archivo Internacional. En el describen los elementos involucrados a cada escenario, como son: descripción, valor

de las variables implicadas en el escenario, la respuesta del sistema y la ruta en el sistema para poder ejecutarlo.

Escenario	Descripción	Variable 6 Fecha de Captura	Respuesta del sistema	Flujo central
EC 1.1: Incluir una nueva Respuesta de Información de Archivo Internacional.	Permite incluir una nueva Respuesta de Información de Archivo Internacional.	N/A	Muestra un mensaje de información: "Se ha creado un elemento".	Agenda de Trabajo / Asignaciones en Curso Se muestra en la tabla "Listado de Resultados" los datos de las asignaciones en curso. 1. Marcar la Solicitud de Información de Archivo Internacional al que se le quiere
EC 1.2: Cancelar.	Permite cancelar la operación.	N/A	Elimina los datos creados. Muestra el Escritorio de Trabajo del usuario autenticado.	Agenda de Trabajo / Asignaciones en Curso Se muestra en la tabla "Listado de Resultados" los datos de las asignaciones en curso.
EC 1.3: Insertar Imágenes.	Permite insertar Imágenes.	V	Inserta tantas imágenes como necesite.	Agenda de Trabajo / Asignaciones en Curso Se muestra en la tabla "Listado de Resultados" los datos de las
EC 1.4: Personalizar la comunicación.	Permite personalizar comunicación.	N/A	Permite personalizar la comunicación.	Agenda de Trabajo / Asignaciones en Curso Se muestra en la tabla "Listado de Resultados" los datos de las
EC 1.5: Existen datos incompletos.	Indica que existen datos incompletos.	V	Muestra el mensaje de error: "Existen campos vacíos que son obligatorios, por favor, complete estos datos". Muestra un indicador sobre los campos vacíos.	Agenda de Trabajo / Asignaciones en Curso Se muestra en la tabla "Listado de Resultados" los datos de las asignaciones en curso. 1. Marcar en el documento al que se le quiere dar respuesta. 2. Presionar el botón "Procesar"

Fig. 21 Caso de Prueba Gestionar Respuesta Información Archivo Internacional.

Un escenario de un caso de prueba, responde a una acción dentro del sistema realizada por el usuario. Para poder probar correctamente un escenario, el probador debe guiarse por la descripción de cada una de las columnas asociadas al escenario que se está probando. Si se toma como ejemplo el escenario (EC) 1.1 de la Fig. 35, es apreciable como existe una columna "Descripción" que contiene la información de lo que el escenario permite hacer. Seguidamente, existen una o varias columnas con todas las variables²⁵ implicadas en la sesión²⁶ del caso de prueba y se especifica si son válidas (V) o no (N/A) para probar el escenario en cuestión. En el caso de la columna "Respuesta del sistema" se muestra la respuesta que debe dar el sistema para el escenario que se está probando. Por último, existe una columna "Flujo central" en la que se muestra la ruta en el sistema para poder probar el escenario que se desea.

Todas las pruebas de sistemas se realizaron guiadas por los casos de prueba elaborados para los submódulos.

²⁵ Son los valores que pueden tomar los campos de entrada de datos, para probar determinada funcionalidad.

²⁶ Agrupa un número significativo de escenarios que conducen a un resultado de valor para el usuario. Puede ser de un caso de uso Gestionar las sesiones Incluir, Ver o Modificar.

3.4.3.1 Pruebas Cruzadas

Las pruebas Cruzadas se realizaron entre los diferentes equipos de desarrollo del proyecto. Los desarrolladores realizaron el papel de probadores agregando rigurosidad a las pruebas por el dominio del negocio, de la estructura interna del software y por conocer cuáles son los errores más comunes durante la implementación. El objetivo principal es detectar la mayor cantidad posible de errores en términos de validaciones y funcionalidades. En la gráfica de la Fig. 36 se muestran los resultados de las 3 iteraciones realizadas de este tipo de prueba, donde las no conformidades se clasifican según su prioridad: en Alta por errores en la interpretación de los procesos de la entidad y de funcionalidad, en Media por errores de terminología y de diseño de interface y en Baja por errores de redacción y ortografía. Según la acción realizada para corregirlas se clasifican en: Corregidas para las resueltas por el equipo de desarrollo y en No Procede, a las detectadas por el desconocimiento de elementos del negocio por parte de los probadores, por cambios ocurridos por restricciones de implementación o ausencia de datos en la aplicación.

Las clasificaciones utilizadas para la prioridad y para las acciones de corrección son aplicables a todas las Pruebas de Sistema que se describen más adelante.

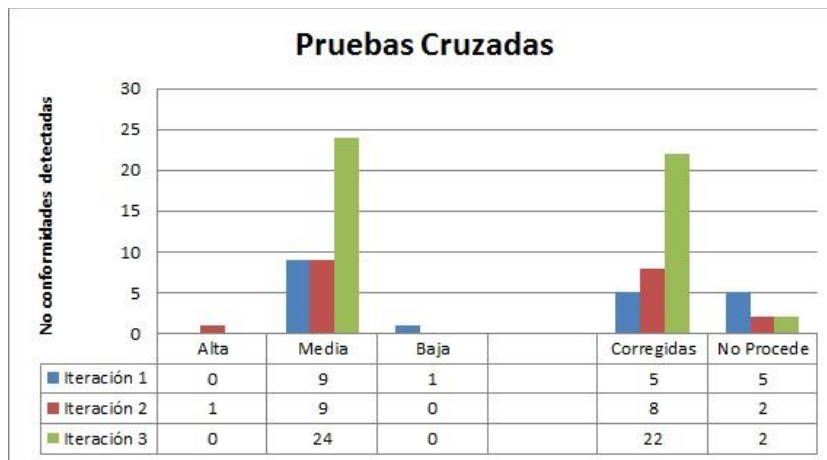


Fig. 22 Resultados Pruebas Cruzadas

Si se analiza la gráfica entre las no conformidades detectadas y las corregidas, es evidente como en cada iteración realizada los errores van en aumento de una iteración a la siguiente. Dando al traste con el pensamiento, de que entre más pruebas sean realizadas, el sistema debería estar más libre de errores.

Sin embargo el equipo de desarrollo no valora estos resultados de esa manera. Para ellos es la evidencia de que pequeños cambios en la aplicación, pueden influir en el funcionamiento de otras funcionalidades del sistema. Por tal motivo es necesario probar con mayor rigurosidad. La iteración 3 fue considerada como la más fructífera en este período de pruebas, por ser la iteración donde se detectaron la mayor cantidad de no conformidades.

3.4.3.2 Pruebas de Calidad Interna

Estas pruebas fueron realizadas por el equipo de Calidad Interna del proyecto, centrándose esencialmente en lo descrito en los requisitos y auxiliándose de los casos de pruebas. El objetivo es entregar un producto con mayor calidad a un equipo de probadores externo al proyecto, y luego entregar la nueva versión al cliente. En la gráfica de la Fig. 26 se muestran los resultados de las pruebas de Calidad Interna, donde las no conformidades se clasificaron en Alta, Media y Baja. Las acciones de corrección se dividieron en Corregido y en No Procede.



Fig. 23 Resultados Prueba Calidad Interna

En el análisis de los resultados de la gráfica entre las no conformidades encontradas y las corregidas, se denota una fluctuación de una iteración a la siguiente. La diferencia visible en las barras que representan cada iteración, dan al traste con los resultados que esperaba el equipo de desarrollo en cuanto a las no conformidades encontradas. El equipo de desarrollo determinó que era necesario realizar una comparación entre los resultados obtenidos en las pruebas internas del proyecto Fig. 38, con el objetivo

de conocer que tan exitosa han sido las pruebas realizadas y si el sistema se encuentra más cerca de cumplir con los requisitos que desea el cliente.

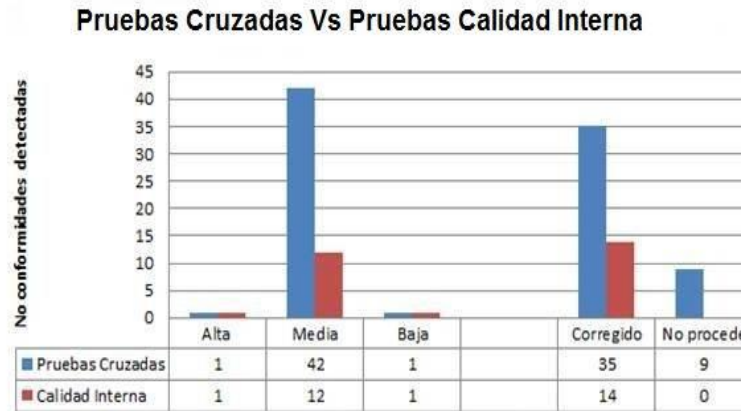


Fig. 24 Resultados Comparación Pruebas Internas

En el análisis de los resultados entre las pruebas internas del proyecto, se determinó que existe una disminución 3 veces menor casi de manera uniforme, entre las pruebas de Calidad Interna y las pruebas Cruzadas. Los resultados de la comparación sirven como argumento para catalogar de muy satisfactorio el proceso de detección y corrección de las no conformidades de las Pruebas Cruzadas, pues hicieron posible que el software llegara a las pruebas de Calidad Interna con la menor cantidad de no conformidades posibles.

3.4.3.3 Pruebas Calisoft

Las pruebas de Calisoft agrupan los objetivos que las pruebas Cruzadas y las pruebas de Calidad Interna persiguen de manera general, probar el sistema como un todo: validaciones, funcionalidades, conexiones a la base de datos, concurrencia, usabilidad. Las ventajas de estas pruebas radican en que son realizadas por un equipo especializado en pruebas de software, donde cada uno de los probadores no tiene que conocer el sistema, por lo que se crea el ambiente propicio para ejecutar acciones en muchos casos no permitidas o no tenidas en cuenta. En la gráfica de la Fig. 39 se muestran los resultados de las pruebas realizadas por Calisoft, en las cuales se utilizaron las mismas clasificaciones para la prioridad de las no conformidades detectadas y para las acciones de corrección, que son empleados en las dos pruebas de sistema descritas con anterioridad.

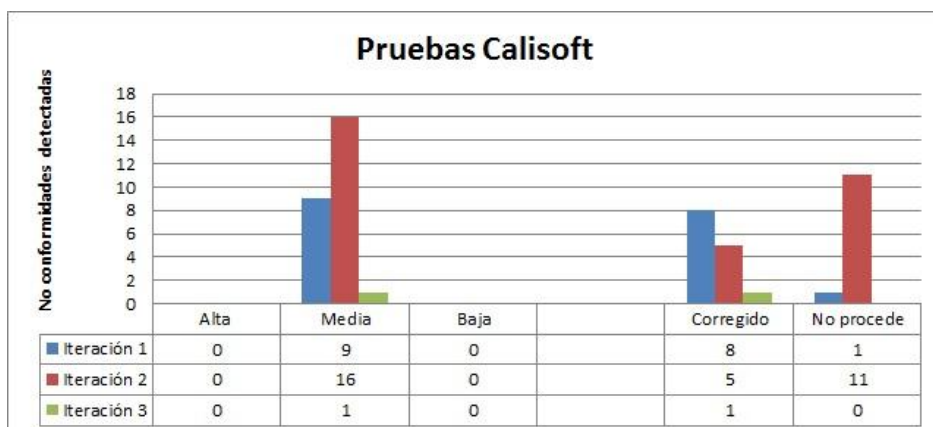


Fig. 25 Resultados Pruebas de Calisoft

En el análisis de los resultados de la gráfica entre las no conformidades encontradas y las corregidas, se observa una disminución de errores detectados de una iteración a la siguiente, se destaca una sola no conformidad detectada. Se arriba a la conclusión de que el sistema está en condiciones de ser presentado al cliente para su aceptación.

3.4.3.4 Pruebas de Aceptación

Las pruebas de aceptación tienen como objetivo la evaluación del producto y la revisión de la documentación final en conjunto con el cliente. Estas pruebas no se realizan durante la etapa de desarrollo, el objetivo es que el cliente pruebe el sistema en un ambiente muy parecido al de producción. En las pruebas de aceptación el cliente puede aceptar lo implementado o realizar pedidos de cambios²⁷. Durante la realización de las pruebas de aceptación al módulo Interpol, se detectaron 2 no conformidades que se clasificaron según su complejidad de igual manera que las pruebas anteriores. El cliente solicitó 17 pedidos de cambios, que fueron atendidos satisfactoriamente. Después de culminadas las pruebas de aceptación, resueltas todas las no conformidades y algunos pedidos de cambio, los submódulos quedaron listo para entrar en las pruebas piloto. En las pruebas piloto el cliente quedó satisfecho con los resultados obtenidos y no se creyó necesario reflejar dichos resultados en el documento. Es importante destacar que después de las pruebas piloto los submódulos quedaron puestos en producción, aguardando solo por la creación de los usuarios y sus perfiles.

²⁷ Petición realizada por el cliente para cambiar algunos elementos del software.

3.5 Conclusión

En este capítulo se dio cumplimiento a los objetivos de validación de la implementación de la propuesta de solución. Se aplicó el método de Caja Blanca para las pruebas de Unidad e Integración y el método de Caja Negra para las pruebas de Sistema. Se elaboraron gráficas con los resultados obtenidos en cada una de las pruebas, lo que ofreció la posibilidad al equipo de desarrollo de analizarlas, valorarlas y poder trazar estrategias para mejorar la calidad del producto. Durante todo el período de pruebas de Sistema, se detectaron 2 no conformidades de complejidad alta, 82 de complejidad media y 2 de complejidad baja para un total de 86 no conformidades y se realizaron por el cliente un total de 17 pedidos de cambios. De las no conformidades encontradas 61 fueron corregidas y 21 no procedían concentrándose el mayor número, en errores de terminología y de diseño de interfaz de usuario. Después de corregidas todas las no conformidades detectadas y atendidos los pedidos de cambio, se está en poder de un producto con un nivel de calidad aceptable, que se ajusta a las especificaciones documentadas y listo para entrar en producción.

Conclusiones Generales

En el presente documento se describe un largo período de investigación para darle solución al problema existente en la Dirección de Policía Internacional. Fue necesario definir los objetivos específicos para delimitar las tareas a realizar en aras de darle cumplimiento al objetivo general de este trabajo. Como parte de las tareas realizadas en busca de cumplir con los objetivos trazados, se estudiaron otros sistemas de gestión de información policial. Con el estudio de estos sistemas se pretendía obtener elementos que ayudaran a desarrollar la solución. Se buscó información de cómo se realizaban las solicitudes internacionales, qué formato o nomenclatura utilizaban para almacenar la información, si tenían o no comunicación con otros sistemas. Aunque la búsqueda no fue lo suficientemente satisfactoria, porque los sistemas estudiados son software hecho a la medida de las necesidades y condiciones del entorno donde se utilizan, sí fue posible conocer que todos de una manera u otra manejan información con similitud a la que se gestiona en la Dirección de Policía Internacional. Se realizó un estudio de las herramientas de desarrollo utilizadas para la implementación del SIIPOL, donde fue posible conocer que son muy reconocidas en el mundo y favorecen el desarrollo de software. Se elaboraron los artefactos propuestos por RUP para los flujos de trabajo Análisis, Diseño e Implementación, lo que posibilitó obtener una visión de lo que se estaba desarrollando, en cuanto complejidad e implicación con otros subsistemas. La documentación generada se agregó a la existente en el SIIPOL.

Durante la realización del diseño de la solución, se estudiaron los elementos más significativos de la arquitectura del SIIPOL, lo que hizo posible que la implementación de los submódulos a desarrollar se realizara bajo la rectoría de la misma sin ningún problema. Se realizaron las pruebas necesarias que verificaron el correcto funcionamiento del sistema y que aportaron un nivel de calidad aceptable, para presentar el producto al cliente y ser puesto en producción.

De manera general se dio cumplimiento al objetivo general de la investigación y se documentó todo el proceso para alcanzarlo.

Recomendaciones

- Realizar la refactorización de los submódulos implementados para hacer más legible el código, alcanzar mayores niveles de mantenimiento del producto, un mayor nivel de cohesión y bajo acoplamiento.
- Analizar la posibilidad de ofrecer nuevas funcionalidades a los usuarios para mejorar la gestión de toda la información en la Dirección de Policía Internacional.
- Tener en cuenta este trabajo para proyectos futuros, siempre y cuando se cumpla con las normas de confidencialidad establecidas.

Referencias Bibliográficas

1. COMISIÓN INTERAMERICANA DE DERECHOS HUMANOS.

INFORME SOBRE SEGURIDAD CIUDADANA . s.l. : OEA, 31 diciembre 2009.

2. **Ponjuan, Yamil and de la Cruz, Agner.** *Análisis, Diseño e Implementación del submódulo Investigación Internacional del Sistema de Investigación e Información Policial.* 2011. 2.

3. <http://www.cicpc.gov.ve>. <http://www.cicpc.gov.ve>. [Online] [Cited: 07 08, 2011.]
<http://www.cicpc.gov.ve/policiainternacional>. 3.

4. <http://www.radiomundial.com.ve>. <http://www.radiomundial.com.ve>. [Online] [Cited: 07 06, 2011.]
<http://www.radiomundial.com.ve/yvke/noticia.php?41837>. 4.

5. **Almeida Hernández, Olivia and Almaguer Ricardo, Maité.** *Ingeniería de Requisitos para el Módulo Interpol del SIIPOL.* 2010. 5.

6. <http://www.bsigroup.com.mx>. <http://www.bsigroup.com.mx>. [Online] [Cited: 07 08, 2011.]
<http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.

7. **INTERPOL.** www.interpol.int. www.interpol.int. [Online] [Cited: 07 08, 2011.] www.interpol.int. 6.

8. <http://www.informador.com.mx>. <http://www.informador.com.mx>. [Online] [Cited: 07 07, 2011.]
<http://www.informador.com.mx/mexico/2009/92207/6/mexico-sera-plataforma-como-principal-fuente-de-informacion-policia.html>. 7.

9. **INTERPOL.** <http://www.interpol.int>. <http://www.interpol.int>. [Online] [Cited: Julio 06, 2011.]
<http://www.interpol.int/es/Centro-de-prensa/Noticias-y-comunicados-de-prensa/2008/N20080603>.

10. **www.mir.es.** <http://www.mir.es>. [Online] [Cited: 07 08, 2011.]
http://www.mir.es/SGACAVT/extranje/extschengen/sistema_informacion.html. 8.

11. <http://www.gobierno.pr>. [Online] [Cited: 07 08, 2011.]
<http://www.gobierno.pr/SIJC/InteresPublico/AnuncioNuevoSistema.htm>. 9.
12. **Grady Booch, Ivar Jacobson, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Pearson Educación S.A, 2000. 11.
13. **Larman, Craig.** *UML y Patrones 1ra Edición*. Mexico : PRENTICE HALL, 1999. 12.
14. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico. Quinta Edición*. Febrero 2002. 13.
15. <http://www.visual-paradigm.com>. <http://www.visual-paradigm.com>. <http://www.visual-paradigm.com>. [Online] [Cited: 01 30, 2012.] <http://www.visual-paradigm.com/product/vpum/>. 14.
16. **ORACLE.** <http://download.oracle.com>. <http://download.oracle.com>. [Online] [Cited: 10 01, 2011.]
<http://download.oracle.com/javaee/5/tutorial/doc/>. 15.
17. **Ramos Rodríguez, Msc. Yadiel .** *Descripción de la Arquitectura de Software*. 2009. 18.
18. **ALBET, S.A.** *MANUAL DE TRANSFERENCIA TECNOLÓGICA. ARQUITECTURA SISTEMA SIIPOL*. Ciudad de la Habana : s.n.
19. **Oracle Corporation.** *The Java EE 6 Tutorial*. Redwood City : s.n., 2011. Oracle.
20. **Mann, Kito D.** *Java Server Faces in Action*. 2004.
21. **Red Hat.** *RichFaces Developer Guide*. 2007.
22. **springsource.org.** <http://static.springsource.org>. <http://static.springsource.org>. [Online]
<http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/html/>.
23. **Red Hat, Inc.** <http://docs.jboss.org>. *HIBERNATE - Relational Persistence for Idiomatic Java*. [Online] 05 02, 2012. http://docs.jboss.org/hibernate/orm/4.1/manual/en-US/html_single/.

24. **López Espinosa, Ing. Kenny and Díaz Cabrera, Ing. Maylin.** *Análisis, Diseño e Implementación de la Agenda de Trabajo del Sistema de Investigación e Información Policial.* Ciudad de la Habana : s.n., 2009. 19.
25. **Serrano Hernández, Arniel and Rodríguez Hernández, Armando Alejandro.** *Software de Migración de ADABAS a Oracle.* 2008.
26. **Sun Microsystems, Inc.** <http://java.sun.com>. [Online] 2002. [Cited: 02 01, 2012.] <http://java.sun.com/blueprints/patterns/CompositeView.html>.

Bibliografía Consultada

27. **Monagas Reyes, Ing. Miguel Angel.** *Documento Procesos elementales del negocio Etapa V.* 10.
28. **Gamma, Erich, y otros, y otros.** *Elements of Reuseable Object-Oriented Software.* 21.
29. **Velazco Elizondo, Perla Inés.** *Pruebas de componentes de software basada en el modelo de JavaBeans.* s.l. : Apizaco, Tlax, Abril 2001.
30. **Vladés Jiménez, Ing. Sasha.** *Glosario de Términos.* 2007.
31. **Ramírez Luján, Ing. Dariena.** *Arquitectura de la Información V2.0 SIIPOL.*
32. **García Córdoba, Fernando.** *La Tesis y el trabajo de tesis.* Mexico : Limusa, 2004.
33. **Pérez Martinto, Msc. Pedro Carlos.** *El Problema Científico.*
34. **Blanco Pérez, Dr. Antonio y Addine Hernández, Dra. Fátima.** *Acerca de la orientación y asesoría del trabajo científico de maestrantes y doctorantes.*
35. **Hernández Leon, Rolando Alfredo y Coello González, Sayda.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA.* Ciudad de la Habana : Editorial Universitaria Cubana, 2011.

36. **Ministerio del Poder Popular para Relaciones Interiores y Justicia.** www.saime.gob.ve. [En línea]
[Citado el: 20 de 09 de 2011.]

Glosario de términos

Ajax: Asynchronous JavaScript And XML. Conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que nos permiten hacer páginas de internet más interactivas.

API: Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Backing Beans: Bean Administrado de JSF que está asociado a un componente de interfaz de usuario en una determinada página.

Bean Administrados (Managed Beans): Contenedor ligero para administrar objetos POJOs con los requerimientos mínimos. Suponen una generalización de los beans manejados para la tecnología JSF.

CSS: Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML y separar la estructura de un documento de su presentación.

Diligencias: En el marco del CICPC se le conoce a las acciones encaminadas a realizarse para el esclarecimiento de un hecho delictivo y que poseen respaldo legal.

Experticias: En el marco del CICPC se le conoce como un conjunto de pruebas que pueden ser llevadas a cabo para el esclarecimiento de un hecho delictivo. Pueden ser criminalísticas, forenses y químicas.

JavaScript: Lenguaje de programación interpretado que permite mejoras en la interfaz de usuario y páginas web dinámicas.

Java Server Pages: Proporciona una forma simplificada y rápida de crear contenido web dinámico. La tecnología JSP permite el desarrollo rápido de aplicaciones web con tecnología java del lado del servidor.

HTTP Request: Petición que envía el navegador a un servidor web para solicitar algún recurso.

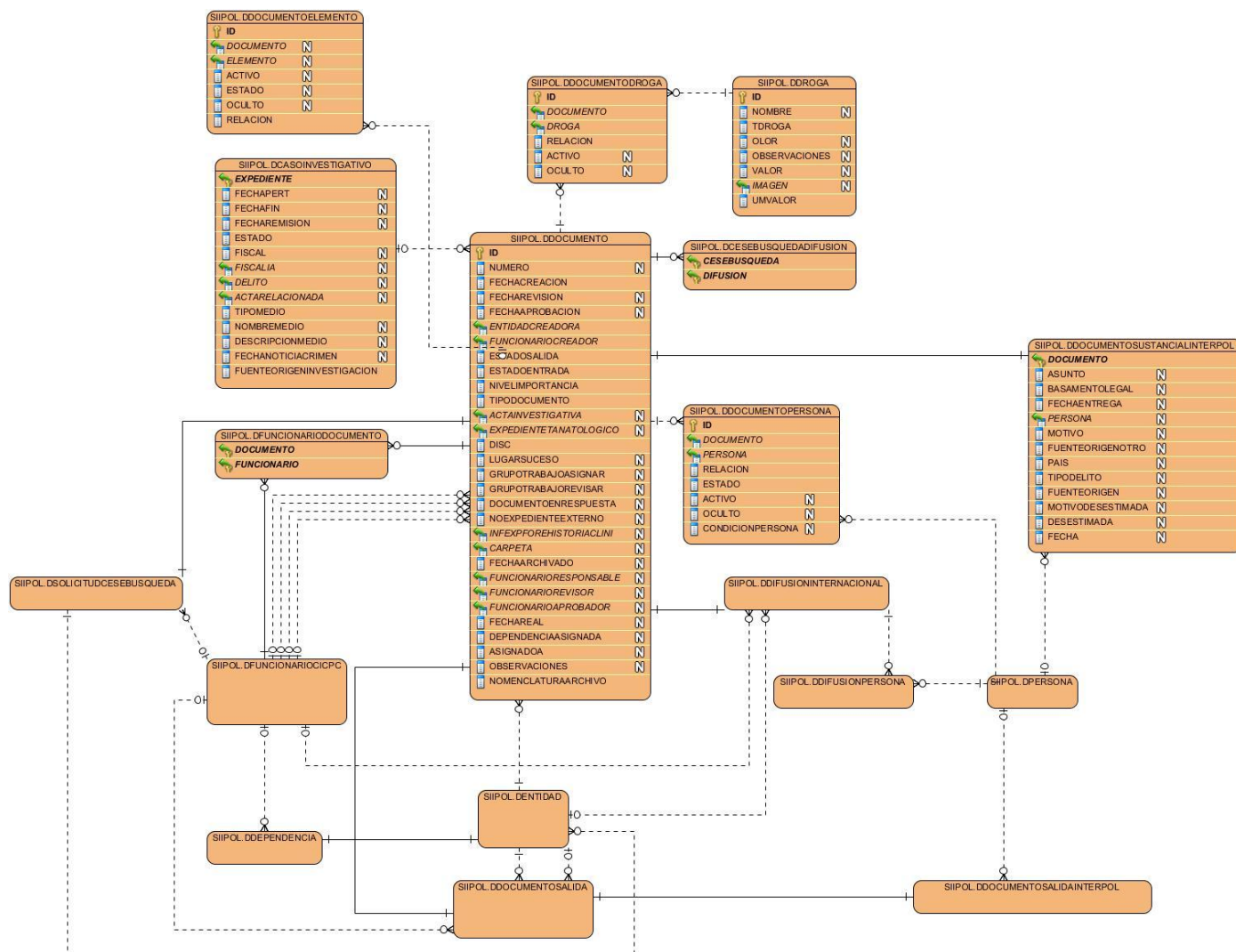
HTTP Response: Respuesta que se envía desde un servidor web hacia el navegador, en respuesta a una solicitud.

Plugins (Complemento): Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

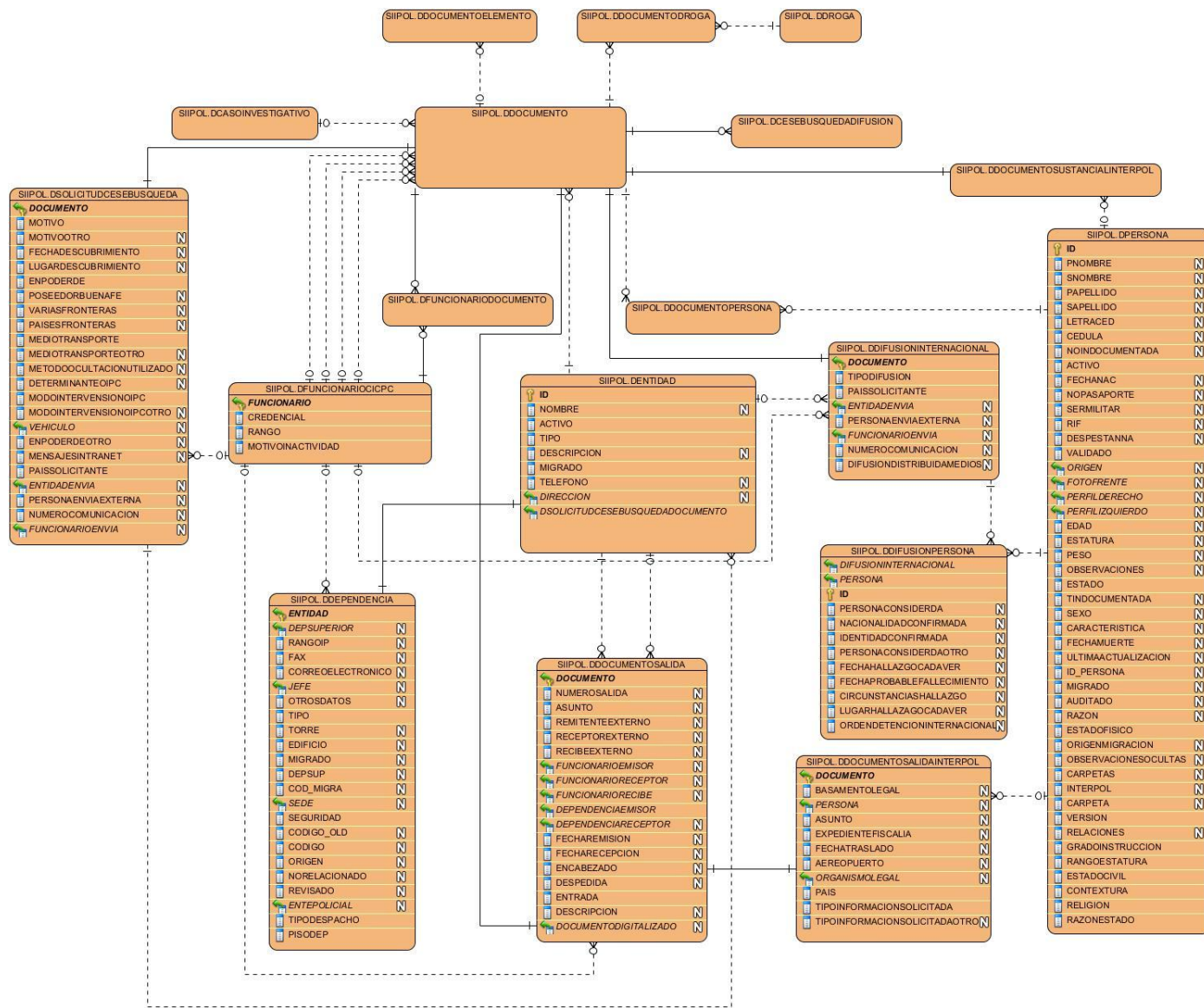
Rich Internet Application: Las aplicaciones RIA son un nuevo tipo de aplicaciones con más ventajas que las tradicionales aplicaciones web. Surgen como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales.

Scripts: Es una secuencia de instrucciones que pueden ser interpretadas por un programa.

Anexos



Anexo 1 Modelo Relacional principales entidades



Anexo 2 Modelo Relacional principales entidades.