



Documento de Arquitectura de la
Solución
V 0.2



DISEÑO, DESARROLLO E IMPLANTACIÓN DEL SISTEMA DE INFORMACIÓN MISIONAL (SIM) DE LA PROCURADURÍA GENERAL DE LA NACIÓN

DOCUMENTO DE ARQUITECTURA DE LA SOLUCIÓN

Código: SAD 0.2



 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <small>Soluciones y Servicios IT</small>
---	---	--

1. Control del Documento

1.1 Cláusula de confidencialidad

La información contenida en este documento es generada en el marco del Contrato 054 celebrado entre *LA PROCURADURÍA GENERAL DE LA NACION* y *SYNAPSIS LTDA*. Dado que la información describe procesos sensibles de naturaleza competitiva, puede resultar en perjuicio de las dos partes del contrato, que sea conocido por personas distintas a aquellas a las que está dirigida. Por tales razones, no podrá ser reproducida, mostrada o divulgada sin el correspondiente permiso escrito de *LA PROCURADURÍA GENERAL DE LA NACION* y *SYNAPSIS LTDA*.

1.2 Control de versiones

Fecha de Actualización	Versión	Revisado por	Cambio / Comentarios
15/03/2007	0.1	Cesar Ariza	Creación del Documento
10/04/2007	0.2	Cesar Ariza	Modificación del Documento
10/04/2007	0.2	Mario Ríos	Revisión del Documento

1.3 Aprobación del Documento

Rol	Nombre	Firma	Fecha
Gerente de Proyecto SIM – PGN	Evelin Julio Estrada		
Supervisor del contrato SIM – PGN	Javier Salazar Cedeño		
Asesor de calidad proyecto SIM - PGN	Rigoberto Rodríguez Peralta		
Gerente Proyecto SIM – Synapsis	Germán Díaz Montoya		

TABLA DE CONTENIDO

1. CONTROL DEL DOCUMENTO	2
1.1 CLÁUSULA DE CONFIDENCIALIDAD.....	2
1.2 CONTROL DE VERSIONES.....	2
1.3 APROBACIÓN DEL DOCUMENTO	2
Lista de figuras.....	6
Lista de Tablas	8
2. GENERALIDADES.....	9
2.1 <i>OBJETIVO</i>	9
2.2 <i>ALCANCE</i>	9
2.3 <i>ORGANIZACIÓN</i>	9
2.4 <i>REPRESENTACIÓN DE LOS STAKEHOLDERS</i>	10
2.5 <i>DEFINICIÓN DE LOS PUNTOS DE VISTA DE LA ARQUITECTURA</i>	10
2.6 MARCO TEÓRICO	12
2.6.1 <i>Edición empresarial de la plataforma Java (Java Enterprise Edition -J2EE-)</i>	12
2.6.2 <i>SOA</i>	14
2.6.3 <i>Patrones y Frameworks</i>	16
2.6.3.1 Patrones	16
2.6.3.2 Frameworks.....	17
2.6.4 <i>Procesos de Negocio & BPEL</i>	17
2.6.4.1 BPEL	18
3. BASES DE LA ARQUITECTURA.....	22
3.1 <i>PROBLEMÁTICA</i>	22
3.1.1 <i>Descripción del Sistema</i>	22
3.1.2 <i>Objetivos y Contexto</i>	23
3.1.3 <i>Requerimientos Significativos</i>	24
3.2 <i>SOLUCIÓN PROPUESTA</i>	26
3.2.1 <i>Enfoque y descripción de la Arquitectura</i>	26
3.2.1.1 Decisión Principal 1: Selección de las herramientas base Oracle y J2EE.....	27
3.2.1.2 Decisión Principal 2: Estilo de arquitectura SOA.....	28
3.2.1.3 Decisión Principal 3: Selección de Frameworks dentro del SIM	30
3.2.1.4 Decisión Principal 4: División física de grandes funcionalidades y contingencia.....	31
3.2.2 <i>Cobertura de los Requerimientos en la Arquitectura</i>	32
4. VISTAS	36
4.1 <i>VISTA CONCEPTUAL</i>	36
4.1.1 <i>Descripción</i>	36
4.1.2 <i>Estructura General</i>	37
4.1.3 <i>Paquetes de la vista</i>	38
4.1.3.1 Paquete “Administración”	38
4.1.3.2 Paquete “Gestión Documental”	39
4.1.3.3 Paquete “Relatoría”	40

4.1.3.4	Paquete “Integración”	41
4.1.3.5	Paquete “Gestión de procesos”	42
4.1.3.6	Paquete “Operaciones”	43
4.1.3.7	Paquete “Consultas y reportes”	45
4.2	VISTA LÓGICA	46
4.2.1	<i>Descripción</i>	46
4.2.2	<i>Estructura General</i>	46
4.2.3	<i>Modelo Arquitectónico</i>	47
4.2.4	<i>Paquetes de la vista</i>	47
4.2.4.1	Paquete “web”	48
4.2.4.2	Paquete “bpel”	49
4.2.4.3	Paquete “Servicios”	51
4.2.4.4	Paquete “datos”	52
4.2.4.5	Frameworks utilizados transversalmente en la aplicación	54
4.3	VISTA FÍSICA	58
5.	RELACIONES ENTRE LAS VISTAS	60
5.1	<i>RELACIONES GENERALES</i>	60
5.2	<i>RELACIONES VISTA CONCEPTUAL Y LÓGICA</i>	60
5.2.1	<i>Relación del Paquete Administración de de vista Conceptual</i>	61
5.2.2	<i>Relaciones del Paquete Gestión Documental de la vista Conceptual</i>	62
5.2.3	<i>Relaciones del Paquete Relatoría de la vista Conceptual</i>	63
5.2.4	<i>Relaciones del Paquete Integración de la vista Conceptual</i>	64
5.2.5	<i>Relaciones del Paquete Gestión de la vista Conceptual</i>	65
6.	MODELO DE DOMINIO	66
6.1	DESCRIPCIÓN	66
6.2	ELEMENTOS	66
6.2.1	<i>Procesos</i>	67
6.2.2	<i>Personas</i>	68
6.2.3	<i>Estructura Organizacional</i>	69
6.2.4	<i>Relatoría</i>	70
6.2.5	<i>Instrumentos</i>	71
6.2.6	<i>Operativo</i>	72
6.2.7	<i>Seguridad</i>	73
6.2.8	<i>Gestión Documental</i>	74
7.	AMBIENTE DE DESARROLLO	75
7.1	APLICACIONES DENTRO DEL AMBIENTE DE DESARROLLO	75
7.2	LISTA PRINCIPAL FRAMEWORKS UTILIZADOS	79
7.3	INTERFAZ DE USUARIO	80
7.3.1	<i>Elementos</i>	81
7.3.1.1	Formas Portal	82
7.3.1.2	Formas Primarias	82
7.3.1.3	Formas Administrativas	85
7.3.1.4	Formas de Ayuda	87
8.	MATERIAL DE REFERENCIA	89



9.	DIRECTORIO	90
9.1	<i>ÍNDICE</i>	90
9.2	<i>GLOSARIO.....</i>	90
9.3	<i>LISTADO DE ACRÓNIMOS.....</i>	91



Lista de figuras	Figura 1. Arquitectura genérica J2EE	13
	Figura 2. Ejemplo de un proceso BPEL	21
	Figura 3. Clasificación modular de las tareas que soportaría e SIM	26
	Figura 4. Vista lógica simplificada de la Arquitectura	27
	Figura 5. Vista Conceptual	37
	Figura 6. Paquete de Administración	38
	Figura 7. Gestión documental	39
	Figura 8. Vista Conceptual Paquete de Relatoría	41
	Figura 9. Vista Conceptual Paquete de Integración	41
	Figura 10. Vista conceptual Gestión de Procesos.....	42
	Figura 11. Vista conceptual Paquete Operaciones	44
	Figura 12. Vista Conceptual Paquete Consultas y Reportes	45
	Figura 13. Vista Lógica SIM (simplificada)	46
	Figura 14. Vista Lógica Paquete web.....	48
	Figura 15. Vista Lógica del Paquete bpel.....	50
	Figura 16. Vista Lógica Paquete servicios	51
	Figura 17. Vista Lógica Paquete datos.....	53
	Figura 18. Frameworks Utilizados	54
	Figura 19. Arquitectura Física	58
	Figura 20. Vista Conceptual paquete de Administración.....	61
	Figura 21. Relaciones con la Vista Lógica del paquete de Administración-Conceptual.	61
	Figura 22. Vista Conceptual paquete de Gestión documental	62
	Figura 23. Relaciones del paquete de Gestión docuemental en la Vista Lógica	62
	Figura 24. Paquetes relacionados en la vista Lógica del paquete relatoria	63
	Figura 25. Vista Conceptual paquete Gestión procesos	65
	Figura 26. Relaciones con la vista lógica con el paquete Gestión procesos-conceptual	65
	Figura 27. Paquete Modelo de Dominio	66
	Figura 28. Modelo de dominio – Procesos	67
	Figura 29. Modelo de Dominio - Personas	68
	Figura 30. Modelo de dominio - Estructura Organizacional	69

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

Figura 31. Modelo de Dominio – Relatoría	70
Figura 32. Modelo de dominio - Instrumentos	71
Figura 33. Modelo de Dominio - Operativo.....	72
Figura 34. Modelo de dominio - Seguridad	73
Figura 35. Modelo de domino - Gestión Documental	74
Figura 36. Interfaz de usuario.....	81
Figura 37. Forma de entrada al sistema.....	82
Figura 38. Forma de Lista de Tareas	83
Figura 39. Forma específica de una tarea.....	84
Figura 40. Forma de información BPEL.	85
Figura 41. Forma genérica de búsqueda.	86
Figura 42. Ejemplo de detalle de forma administrativa	87
Figura 43. Modelo de Navegación de la Ayuda.....	88



Lista de Tablas

Tabla 1. Stakeholders y puntos de vista de su interés	11
Tabla 2. Actividades principales en BPEL.....	18
Tabla 3. Actividades estructuradas en BPEL	19
Tabla 4. Otros elementos en BPEL.....	20
Tabla 5. Cobertura de los principales requerimientos no funcionales.....	32

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <small>Soluciones y Servicios IT</small>
---	---	--

2. GENERALIDADES

2.1 *Objetivo*

Este documento provee una visión comprensible de la arquitectura del **SIM**, usando un grupo de vistas para representar los diferentes aspectos del mismo. Su intención es capturar y comunicar las decisiones significativas de arquitectura que se tomaron sobre el sistema.

2.2 *Alcance*

Este documento es de interés a todos los stakeholders del proyecto **SIM** que desean conocer la arquitectura de la solución propuesta, para lo cual se proveen vistas que intentan brindar la información requerida desde diferentes ángulos.

Este documento comprime gran parte de los diagramas levantados en las fases de análisis y diseño detallado, brindándoles un contexto y una explicación a sus relaciones.

En el marco de PROCURADURÍA GENERAL DE LA NACIÓN, este documento es un insumo para el flujo de evaluación arquitectónica de proyectos en la fase de diseño, ya que provee una correspondencia entre las funcionalidades de negocio y las opciones tecnológicas que las atacan.

2.3 *Organización*

Este documento de arquitectura está organizado en las siguientes secciones:

Sección 1. Provee información acerca del documento y su audiencia esperada. Todos los lectores que deseen encontrar información relevante de la arquitectura de software descrita en este documento deberían comenzar leyendo la sección 1, que describe cómo está organizado el documento, cómo está representada la arquitectura y qué información puede ser encontrada.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

Sección 2. Provee información acerca de la arquitectura de software. Describe las bases y el raciocinio usados para concebir la arquitectura propuesta del software. Explica las restricciones e influencias que llevaron a la definición de la arquitectura, describiendo las alternativas más importantes que se descartaron en el proceso.

Secciones 3 y 4. Especifican la arquitectura de software propuesta.

Secciones 5 y 6. Proveen material de referencia. La sección 5 provee información acerca de los documentos utilizados como referencia en la elaboración del presente documento. La sección 6 provee un directorio, que es un índice de los elementos y relaciones de arquitectura, indicando dónde se define y dónde se usa dentro del documento. Así mismo, provee un glosario y una lista de acrónimos con el fin de aclarar términos usados dentro del documento.

2.4 *Representación de los Stakeholders*

Los interesados en el proyecto están detallados en el documento de Visión. A continuación se presenta una lista de los mismos:

- Desarrolladores de software
- Usuarios finales
- Ingenieros de infraestructura de hardware y software
- Ingenieros de seguridad
- Ingenieros de comunicaciones
- Comité de arquitectura
- Entidades reguladoras externas
- Integradores e ingenieros de pruebas
- Administradores de sistemas en producción
- Entrenadores
- Ingenieros de mantenimiento de aplicaciones
- Auditores

2.5 *Definición de los Puntos de Vista de la Arquitectura*

Los puntos de vista de arquitectura permiten observar un sistema desde diferentes ángulos que unidos brindan la descripción completa del mismo. Los puntos de vista tratados son: Conceptual, Lógico y Físico.

Conceptual. Acá, el elemento primario es el módulo: una unidad que implementa una serie de responsabilidades. Los módulos representan una forma de considerar el sistema basada en código, puede ser una clase, colección de clases, capa, o cualquier descomposición de una unidad de código. Los módulos son áreas de responsabilidad funcional, asignadas a equipos para su implementación. Existe menos énfasis en cómo el software resultante se manifiesta a sí mismo en tiempo de ejecución. La estructura de los módulos permiten resolver preguntas como: ¿Cuál es la responsabilidad funcional primaria asignada a cada módulo?, ¿Qué otros elementos de software está en capacidad de usar?, ¿Qué otros elementos de software en realidad usa? ¿Qué relaciones existen entre los módulos (Descomposición, Herencia, Capas)?

Lógico. Acá, los elementos son componentes de tiempo de ejecución (que son unidades principales de procesamiento) y conectores (que son los mecanismos de interacción entre componentes). Objetos, procesos o colecciones de objetos pueden ser componentes. Los conectores incluyen pipes, repositorios, colas y sockets. Las estructuras de componentes y conectores ayudan a contestar preguntas como: ¿Cuáles son los principales componentes de ejecución y cómo interactúan? ¿Cuáles son los principales depósitos de datos? ¿Qué partes del sistema están replicadas?

Físico. Acá, se muestra la relación entre los elementos de software con otros elementos de su entorno. Las estructuras físicas responden a preguntas como: ¿En qué ambiente se ejecuta cada elemento de software? ¿En qué repositorios se guarda cada elemento durante desarrollo, pruebas y producción? ¿Cuál es la asignación de los elementos de software a los equipos de desarrollo?

En la siguiente tabla se muestra la correspondencia entre las vistas de arquitectura con el posible público interesado:

Tabla 1. Stakeholders y puntos de vista de su interés

Stakeholder	Punto de vista
Desarrolladores de software	Conceptual, Lógico, Físico
Usuarios finales	Conceptual
Ingenieros de infraestructura de hardware y software	Lógico, Físico

Stakeholder	Punto de vista
Ingenieros de seguridad	Lógico, Físico
Ingenieros de comunicaciones	Físico
Comité de arquitectura	Conceptual, Lógico, Físico
Integradores e ingenieros de pruebas	Lógico, Físico
Administradores de sistemas en producción	Lógico, Físico
Entrenadores	Lógico, Físico
Ingenieros de mantenimiento de aplicaciones	Conceptual, Lógico, Físico
Auditores	Cualquiera

2.6 Marco Teórico

En esta sección se introducen brevemente tecnologías utilizadas dentro del proyecto.

2.6.1 Edición empresarial de la plataforma Java (Java Enterprise Edition - J2EE-)

J2EE es una plataforma Java diseñada para centros de cómputo de gran tamaño (mainframe-scale) de empresas grandes. J2EE fue diseñado por Sun Microsystems (junto con otras empresas como IBM) para simplificar el desarrollo de aplicaciones en ambientes multicapa para clientes delgados. J2EE simplifica el desarrollo de aplicaciones y disminuye la necesidad de programación y el entrenamiento de los programadores por medio de la creación de componentes estandarizados y modulares que componen las capas que permiten el manejo automático de muchos aspectos de la programación.

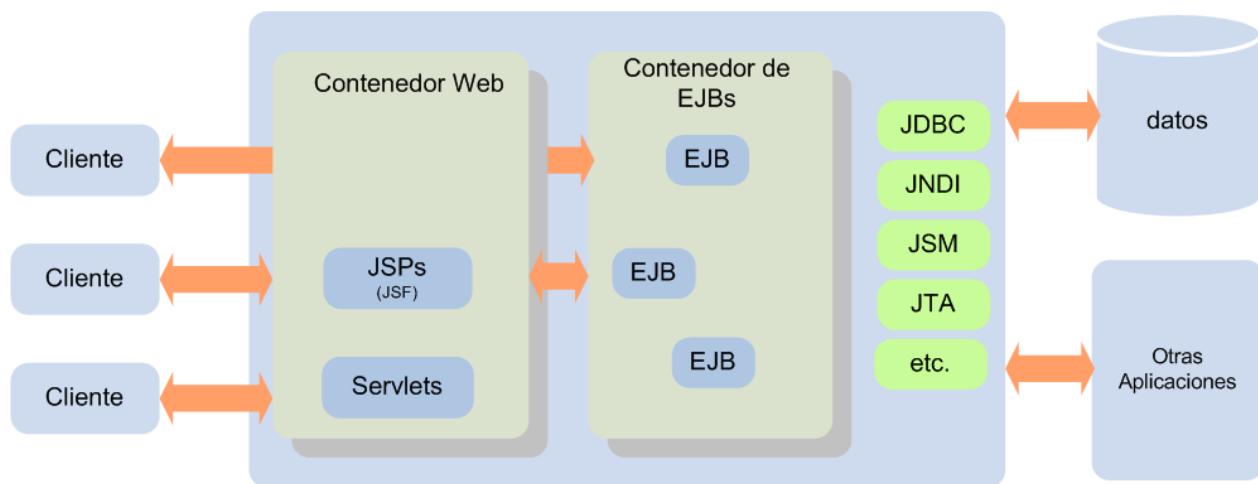


Figura 1. Arquitectura genérica J2EE

En la plataforma J2EE, se pueden identificar varias capas (ver Figura 1). Las capas son:

- La capa de presentación, compuesta por los clientes Web y el Contenedor Web.
- La capa de negocios, compuesta principalmente por el Contenedor de EJB
- La capa de datos, compuesta por las otras aplicaciones y/o datos.

La plataforma J2EE incluye varios componentes comunes a otras plataformas (i.e. J2SE) a saber:

- El JDK (Java Development Kit) es incluido como el lenguaje principal.
- La tecnología WORA (Write Once Run Anywhere) es incluida para asegurar portabilidad.
- La plataforma soporta objetos CORBA (Common Object Request Broker Architecture), un predecesor de EJB (Enterprise JavaBeans). Gracias a esto, los objetos Java se pueden comunicar con objetos CORBA (localmente y sobre una red a través de la interfaz del broker).
- Para la comunicación estándar con bases de datos es ofrecido JDBC (Java Database Connectivity 2.0), que es el equivalente a ODBC (Open Database Connectivity).
- Un modelo de seguridad es incluido para proteger los datos locales y las aplicaciones Web.

La plataforma J2EE también incluye otros componentes específicos:

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <small>Soluciones y Servicios IT</small>
---	---	---

- Soporte a EJB. EJB es una tecnología basada en servidores de aplicaciones para proveer componente de programas en un ambiente empresarial. EJB soporta XML (Extensible Markup Language) y tiene características mejoradas para despliegue y seguridad.
- La API para servlets de Java, que mejora la consistencia de las aplicaciones independiente de la interfaz gráfica.
- JSP (Java Server Pages) es el equivalente Java de ASP (Las páginas activas de Microsoft - Active Server Pages) y es utilizado para manipulación y acceso dinámico de datos desde páginas Web.

La arquitectura J2EE consta de cuatro elementos principales:

- El modelo de programación de aplicaciones J2EE que es un estándar utilizado para facilitar el desarrollo de aplicaciones de múltiples capas y clientes delgados (e.g. navegadores de Internet).
- La plataforma J2EE incluye las políticas necesarias y APIs como lo son los servlets de Java y el servicio de mensajería de Java (Java Message Service – JMS-).
- Existe un conjunto de pruebas de compatibilidad para aplicaciones J2EE (J2EE Compatibility Test Suite) que asegura que los productos son compatibles con los estándares de la plataforma.
- Existe un marco de referencia para la implementación de aplicaciones (J2EE Reference Implementation) que explica las capacidades de J2EE y provee una definición operacional del mismo.

2.6.2 SOA

SOA es el acrónimo de arquitectura orientada a servicios o en Inglés Service Oriented Architecture. El estilo SOA es una arquitectura de software que usa servicios de software desacoplados para soportar los requerimientos de procesos de negocios y de los usuarios de software. En SOA, los recursos de red (entiéndase cualquier recurso de software o hardware) están disponibles como servicios independientes que pueden ser accedidos sin saber como fueron implementados.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

SOA también puede ser interpretada como un estilo de arquitectura de sistemas de información que permite la creación de aplicaciones que son construidas combinando servicios desacoplados e inter-operables. Estos servicios, pueden ser invocados gracias a una definición formal de su interfaz (e.g. usando WSDL) que es independiente de la otra plataforma y el lenguaje de programación del servicio. La definición de la interfaz esconde la implementación del servicio. Los sistemas basados en SOA además son independientes de las plataformas (e.g. Java, .NET etc). Por ejemplo servicios escritos en C# ejecutándose en plataformas .NET y servicios escritos en Java ejecutándose en plataformas Java EE, los dos pueden ser consumidos por la misma aplicación. Aplicaciones ejecutándose en cualquier plataforma pueden también consumir servicios ejecutándose en otras plataformas (e.g. servicios Web), lo cual facilita la reutilización.

Los beneficios principales de una arquitectura de estilo SOA son la independencia tecnológica, la independencia del ciclo de vida, el desacoplamiento, interfaces invocables y la variedad de protocolos de comunicación. A continuación son explicadas las características mencionadas anteriormente.

- **Independencia tecnológica:** Como consumidor de servicios, el SIM puede consumir servicios (e.g servicios Web) de cualquier tipo, independiente de la tecnología del proveedor de los servicios.
- **Independencia del ciclo de vida:** Los proveedores y consumidores de servicios pueden operar en un ciclo de vida diferente.
- **Desacoplamiento:** Los componentes consumidores de servicio deben definir su especificación independiente de los componentes proveedores de servicios. La responsabilidad de alinear las dos especificaciones es del componente de ensamblaje el cual hace de puente entre los dos mundos. En conclusión permite acoplamiento de componentes heterogéneos.
- **Interfaces invocables:** Las interfaces de los servicios pueden ser invocadas bien sea remotamente o localmente. A nivel arquitectural no importa como la interfaz es invocada.
- **Protocolo de Comunicación:** Los servicios podrían ser invocados por varios protocolos y la selección del protocolo no debe restringir el comportamiento del servicio. La selección del protocolo específico puede ser hecha en tiempo de ejecución o despliegue y no en tiempo de diseño o desarrollo.

Para el desarrollo de aplicaciones SOA, es utilizado SODA (Services-Oriented Development of Applications). SODA es un estilo de desarrollo utilizado dentro del paradigma SOA. SODA se centra en la creación y el montaje de servicios y de contratos de servicio primero, posponiendo el diseño e implementación de los objetos y componentes que realizan los servicios hasta después que la comunicación entre los

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

servicios se hayan resuelto. Los desarrolladores SODA se centran más en el flujo de procesos dentro y entre aplicaciones, y menos en el código que crea el sistema propiamente.

2.6.3 Patrones y Frameworks

En esta sección se definen los términos Patrón y Framework. A grandes rasgos los patrones esbozan soluciones a problemas y los Frameworks son un conjunto de bibliotecas de software que proveen funcionalidades específicas para solucionar un problema.

2.6.3.1 Patrones

Un patrón de programación o patrón de diseño es una solución para problemas comunes (que se presentan con cierta frecuencia) y cuya solución no es trivial.

El patrón aporta información para la solución a un problema que se presenta en un contexto determinado. Para elaborarlo, se aíslan sus aspectos esenciales y se añaden cuantos comentarios y ejemplos sean necesarios. En particular se puede identificar:

- el contexto en el que es posible aplicar el patrón,
- las fuerzas (objetivos y restricciones) que intervienen en ese contexto,
- y el diseño a aplicar para equilibrar objetivos y restricciones.

La ventaja del uso de patrones es que la solución que aportan es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y es reutilizable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

Un ejemplo (típico) de patrón es el “*Singleton*” o Instancia-Única garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

A principios de los 90's fue publicado el libro "Design Patterns" escrito por el GoF (Gang of Four) compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, en el que se recogían 23 patrones diseño comunes, después de la publicación del libro los patrones de diseño tuvieron un gran éxito en el mundo de la informática.

Finalmente es posible que un patrón de diseño complejo este compuesto por varios patrones de diseño básicos.

2.6.3.2 Frameworks

En este documento, la palabra Framework es utilizada para designar a un conjunto de clases o bibliotecas que son utilizadas para implementar una estructura estándar o funcionalidad específica dentro de una aplicación y para un sistema operativo específico. Al empaquetar una gran cantidad de código reutilizable dentro de un framework, se ahorra mucho tiempo de programación, en comparación con la reescritura de código nuevo cada vez que una aplicación es escrita.

Dentro de los ejemplos de comunes de frameworks se encuentran los que crean interfaces graficas para usuario (e.g. Struts, JSF), o los que hacen registro de eventos (log) entre muchos otros.

Normalmente los frameworks utilizan y/o recomiendan el uso de patrones de diseño. En el caso de interfaces de usuario es frecuente que el framework (e.g. JSF, Struts, etc) utilice el patrón Modelo Vista Controlador (MVC - Model View Controller).

La combinación del frameworks y patrones de diseño aceleran el desarrollo de aplicaciones y garantiza un nivel inicial de calidad alto.

2.6.4 Procesos de Negocio & BPEL

Los procesos son un conjunto de actividades que crean valor transformando una entrada en una salida con mayor valor. Ambos, la entrada o la salida pueden ser artefactos y/o información y la transformación puede ser realizada por humanos maquinas o ambos.

Cada etapa dentro de un proceso puede asimilarse a un servicio (informático), los procesos pueden ser orquestados por medio de un motor de procesos gracias a un hilo conductor. El hilo conductor puede ser una descripción del proceso por medio de un lenguaje de alto nivel (e.g. BPEL).

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

2.6.4.1 BPEL

El Lenguaje de Ejecución de Procesos de Negocio (BPEL- Business Process Execution Language) es un lenguaje basado en XML que permite la descripción formal de procesos de negocio y de protocolos de interacción. BPEL extiende el modelo de interacción de Servicios Web haciendo posible hacer transacciones de negocios.

En otras palabras BPEL permite la descripción de los intercambios de información realizados interna y externamente en los procesos. BPEL se encarga explícitamente de los aspectos funcionales de los procesos de negocios: control de flujo (flujos paralelos, ciclos, bifurcaciones), conversaciones asincrónicas y correlaciones, fallas y compensación. BPEL trata directamente esos aspectos funcionales mediante:

- Coordinando la comunicación asincrónica entre servicios
- Correlacionando el intercambio de mensajes entre las partes
- Implementando procesamiento en paralelo de actividades,
- Manipulando datos entre las interacciones de los socios (e.g. servicios)
- Soportando actividades y transacciones de negocios largas y complejas
- y proveyendo un manejo consistente de excepciones

Elementos en BPEL

Aunque BPEL se expresa con XML, existe la posibilidad de utilizar una interfaz gráfica para crear “procesos” en BPEL. La herramienta JDeveloper (de Oracle) tiene su propia propuesta gráfica para ese fin. A continuación se presentan los elementos de BPEL básicos de BPEL: actividades y actividades estructuradas.

Tabla 2. Actividades principales en BPEL

Elemento & Descripción	Imagen en Jdeveloper
Asignación (Assing): Para manipular variables dentro del proceso	
Vacio (Empty): Elemento vacío utilizado para unir un flujo paralelo	

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

Invocar: Para llamar un socio (partner link) que puede ser un servicio Web.	
Recibir (Receive): Para recibir información de un cliente	
Responder (Reply): Para generar una respuesta a una operación sincrónica.	
Acabar (Terminate): Para terminar con un proceso.	
Lanzar excepción (Throw): Para Lanzar una excepción.	

Tabla 3. Actividades estructuradas en BPEL

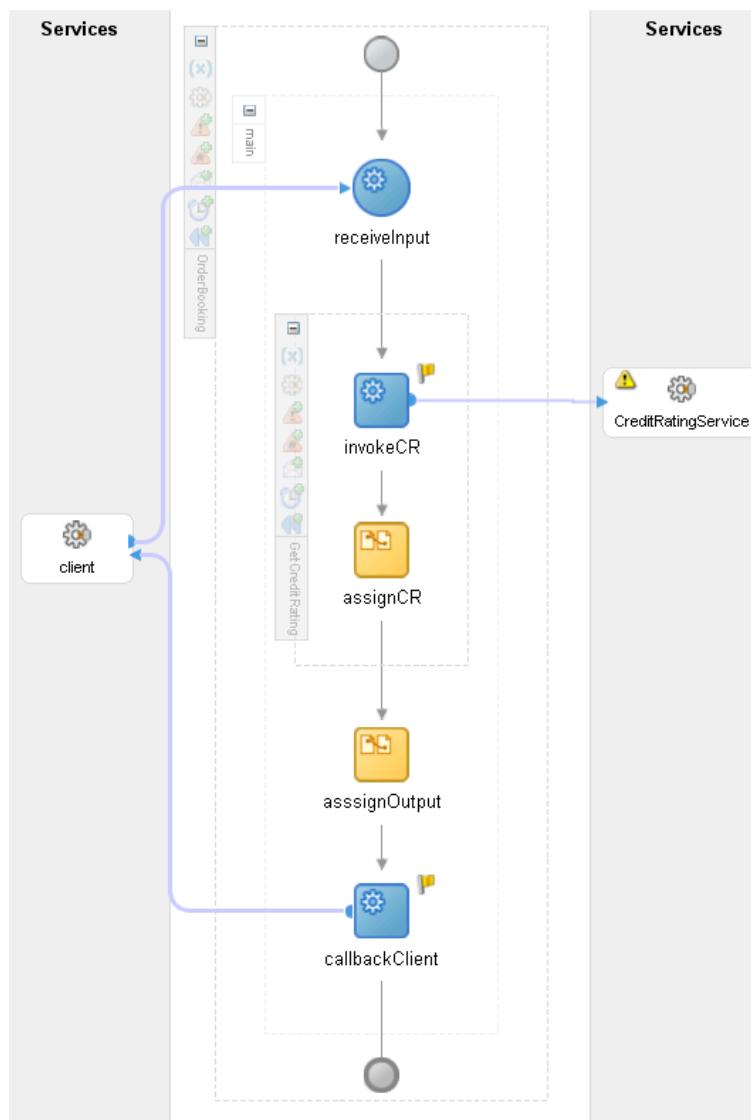
Elemento & Descripción	Imagen en Jdeveloper
Secuencia (sequence): Grupo de actividades ordenadas	
Mientras (While): Para definir ciclos dentro del proceso.	
Flujo (Flow): Un conjunto de actividades en paralelo	
Selección Múltiple (Case -Switch): Para flujos de actividades	
Selección: Para definir caminos alternativos.	

Existen otros elementos BPEL, externos a los procesos y otros que complementan la versión grafica. Esos elementos son presentados en la siguiente tabla.

Tabla 4. Otros elementos en BPEL

Elemento & Descripción	Imagen en Jdeveloper
Inicio y fin: Representa el inicio y e fin de un diagrama de un proceso.	
Sentido de flujo: Representa el sentido del flujo del proceso de un elemento a otro.	
Cliente externo (Partner Link): Representa un cliente externo que interactúa con los procesos. La información del un “cliente externo” debe ser suficiente para invocar el elemento externo (en principio un servicio Web) que representa.	

La Figura 2 muestra como se ve gráficamente un proceso BPEL. También se describe brevemente como es el flujo del proceso.



El proceso es iniciado

El proceso recibe una solicitud de un cliente

El proceso invoca un servicio externo o socio (partner link) solicitando alguna información.

Se asigna la respuesta a una variable.

Se asigna la respuesta a una variable de salida

Se envía la respuesta al cliente (que también es un socio).

El proceso finaliza.

Figura 2. Ejemplo de un proceso BPEL

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

3. BASES DE LA ARQUITECTURA

3.1 *Problemática*

3.1.1 Descripción del Sistema

En los últimos años la Procuraduría General de la Nación (PGN) ha enfrentado dificultades para cumplir con su mandato constitucional debido a una serie de problemas y deficiencias de tipo organizacional y funcional. Entre estos problemas se encuentran: i) un fraccionamiento de sus diversas dependencias especializadas y ii) la división vertical del trabajo funcional y de las diferentes oficinas de apoyo. Este tipo de organización crea duplicidad de funciones, ineficiencias administrativas, falta de coordinación en sus procesos y de claridad en la definición de sus productos institucionales al servicio de la ciudadanía. En respuesta a esta problemática y con el fin de cumplir con las responsabilidades que le asigna la Constitución, la Procuraduría General de la Nación, con apoyo del Banco Interamericano de Desarrollo, elaboró en 2001 un diagnóstico para identificar las debilidades que han limitado su capacidad de funcionamiento e inició un proceso de mejoramiento institucional. Este diagnóstico identificó como principales debilidades: i) la poca conceptualización de la Procuraduría General de la Nación sobre sus funciones y sus alcances; ii) la ausencia de instrumentos y herramientas para ejecutar las funciones misionales; iii) la estructura y la concentración de funciones y procesos en el nivel central; iv) el débil soporte estratégico de las funciones de planeación y de control; v) el rezago tecnológico; vi) la falta de cultura organizacional reflejada en el manejo del talento humano; vii) la imagen desdibujada que tienen los ciudadanos de la Entidad y los escasos espacios de participación, y viii) las deficiencias en el servicio ofrecido [LIC-PGN].

La Procuraduría General de la Nación con el apoyo del Banco Interamericano de Desarrollo y para resolver la problemática identificada, formuló el Programa de Apoyo al Fortalecimiento Institucional, el cual fue aprobado en documento CONPES 3211 de enero 9 de 2003, que autorizó a la Nación para contratar con el Banco Interamericano de Desarrollo un crédito para financiarlo.

El objetivo general del Programa es contribuir con la modernización del Estado mediante el afianzamiento y consolidación de su sistema de control, promoviendo su

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

mayor eficiencia a través de la disuasión de conductas contrarias a la probidad, el fomento al respeto a los derechos fundamentales y la facilitación del control ciudadano sobre la conducta del poder estatal. El propósito específico es la modernización de la Procuraduría General de la Nación de manera que sus capacidades institucionales para proteger los derechos humanos y vigilar la conducta de los servidores públicos queden fortalecidas.

Para ello, el Programa está agrupado conceptualmente en bloques temáticos, así: actividades jurídico-misionales, gestión institucional, infraestructura tecnológica y adecuaciones físicas. Tales bloques permiten, de manera agregada, el fortalecimiento de las funciones misionales a nivel central y territorial, el fortalecimiento del sistema de gestión y el mejoramiento del servicio a la ciudadanía, a la comunidad estatal y a la imagen institucional, a través del diseño, desarrollo e implementación de instrumentos y herramientas que contribuirán con la obtención de resultados más eficientes, ágiles y oportunos.

El proyecto de diseño desarrollo e implantación del Sistema de Información Misional - SIM, está enmarcado dentro de los bloques de actividades jurídico-misionales e infraestructura tecnológica y permite darle continuidad al trabajo realizado por la consultoría contratada en diciembre de 2004 con la Universidad de los Andes, mediante la cual se evaluaron los sistemas de información actuales, se optimizaron los macroprocesos de la PGN, se recomendó la arquitectura de sistemas de información para la PGN en materia misional y se determinaron los requerimientos funcionales y técnicos del nuevo sistema.

El objetivo general del proyecto es el diseño, desarrollo e implantación del Sistema de Información - SIM de manera tal que: i) soporte las funciones misionales de la Entidad; ii) cubra las necesidades de la Entidad en los ámbitos central, regional y local; iii) integre los sistemas de información misionales de la Entidad, y iv) permita ser adaptado a cambios en la normatividad.

3.1.2 Objetivos y Contexto

Existen tres factores contextuales principales por los cuales se adoptó la arquitectura base SOA y J2EE para el SIM. Los factores son descritos a continuación:

Soporte a procesos: La Universidad de los Andes dentro del estudio realizado para la PGN normalizó y homologó los macro-procesos misionales dentro de la PGN. El estudio de la Universidad conllevó a la conclusión que la PGN es una organización

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

orientada a los procesos. Al mismo tiempo los sistemas informáticos que soportan la misión de la PGN deberán tener una componente orientada a los procesos que permitan su seguimiento.

Organización dispersa geográficamente: La procuraduría es una organización dispersa geográficamente lo cual hace la actualización de aplicaciones monousuario costosa. Al mismo tiempo los usuarios en algunas regiones tendrán responsabilidades diferentes a usuarios de otras regiones. Por ejemplo las regionales de ciudades grandes como Bogotá o Bucaramanga, existirán usuarios especializados en un proceso, mientras que en ciudades pequeñas como Melgar, es posible que los usuarios tengan que saber de todos los procesos.

Accesibilidad: Algunas de las actuaciones de la PNG hacen que sea necesario el desplazamiento de funcionarios fuera de sus lugares habituales de trabajo lo que a su vez hace necesario que la aplicación este accesible desde cualquier lugar, cumpliendo con algunos requisitos de conexión. Además algunas de las funcionalidades de la aplicación deberán estar disponibles para los ciudadanos.

El soporte a procesos es garantizado por la adopción de BPEL para la orquestación de los procesos; algunos de los pasos de los procesos serían servicios Web. Para solucionar el factor “dispersión de la organización”, la aplicación será centralizada, lo que facilita a su vez el mantenimiento y seguridad de la información y de la aplicación. Finalmente, la accesibilidad a la aplicación es garantizada por el tipo de aplicación que será una solución con interfaz de usuario Web y con acceso a través de Internet.

La adopción de BPEL, la centralización de la aplicación y la interfaz Web, están reflejados en la arquitectura propuesta. Estos factores se verán principalmente en la vista física de la arquitectura y la vista lógica.

Además los tres factores anteriores también están planteados requerimientos funcionales y no funcionales (que pueden estar traslapados los factores) y son expuestos en la siguiente sección.

3.1.3 Requerimientos Significativos¹

Además de los factores descritos en la sección anterior, existen los normales requerimientos no funcionales dados por la ingeniería de software como lo son: desempeño, disponibilidad, escalabilidad, flexibilidad, mantenibilidad y seguridad.

¹ Esta sección tiene base en el documento [LICA4P2-PGN] que habla sobre los requerimientos no funcionales.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

También existen otros requerimientos con un impacto menor en la selección de la arquitectura como lo son: facilidad de uso e Ingreso de la información, facilidad para las pruebas, instalación, operatividad, validación de la información, backups e integración.

Sin embargo existen algunos requerimientos no funcionales con definiciones específicas para este proyecto, que de alguna forma influencian la arquitectura. Apartes de esos requerimientos son presentados a continuación:

Disponibilidad: Estar disponible 100% o muy cercano a esta disponibilidad durante el horario hábil laboral de la PGN a nivel nacional (i.e. de lunes a viernes de 8:00 a.m. a 5:00 p.m., con excepción de los días festivos).

Escalabilidad: a) El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible; para ello deben incorporarse aspectos de reutilización de componentes. b) El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

Interoperabilidad: El sistema debe estar en capacidad de interactuar con los otros sistemas de la PGN y con sistemas de entidades externas a través de la herramienta de middleware seleccionada para el sistema. La Interoperabilidad debe estar regida por la normas de Arquitectura de Integración e Interoperabilidad de la Comisión Intersectorial de Políticas y Gestión de la Información para la Administración Pública

Arquitectura: a) La solución debe ser 100% Web Based y toda la parametrización y administración debe realizarse desde un navegador. b) La solución debe tener interfaces gráficas de administración y de operación en idioma español y en ambiente 100% Web, para permitir su utilización a través de exploradores o navegadores de Internet.

Existen otros requerimientos no funcionales de la “herramienta base” que definen y condicionan las características del entorno de desarrollo, el motor de la base de datos, el sistema de workflow, el administrador de contenido (CMS²). El CMS esta orientado para la presentación de contenido Web y no debe confundirse con un DMS (acrónimo del término de la lengua inglesa Document Management System)

En la sección 3.2.2 Cobertura de los Requerimientos en la Arquitectura, se explica como la arquitectura plateada resuelve los principales requerimientos.

² CMS es un acrónimo de las palabras de la lengua inglesa “Content Management System”.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

Inicialmente se hizo una clasificación de las áreas misionales dentro de la procuraduría, y en su momento se plantearon subsistemas que de una u otra forma influencio la arquitectura. La Figura 3 presenta un esa aproximación inicial del SIM, cada módulo soportaría tareas de un área misional de la PGN.

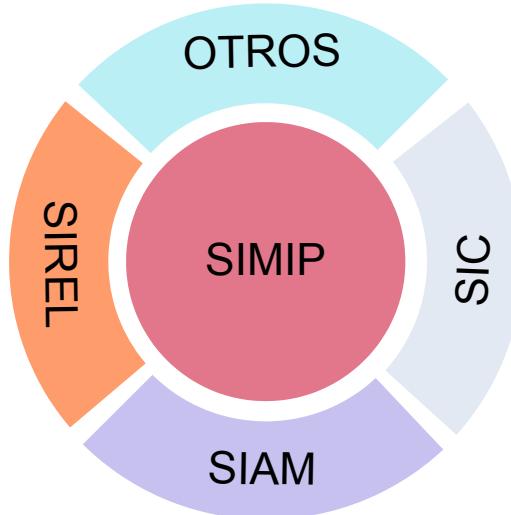


Figura 3. Clasificación modular de las tareas que soportaría e SIM

La explicación de esa clasificación es la siguiente (todas las siglas comienzan por SI que significa subsistema de información):

SIC: Tareas relacionadas con la información al Ciudadano

SIREL: Tareas relacionadas con la Relatoría de la PGN

SIAM: Tareas relacionadas con el apoyo a la misión de la PGN

SIMIP: Tareas relacionadas con los procesos.

OTROS: Son las demás tareas que no tienen una clasificación clara dentro de la misión de la PGN pero si dentro del SIM, como por ejemplo la autenticación de usuarios.

3.2 Solución Propuesta

3.2.1 Enfoque y descripción de la Arquitectura

Esta sección contiene las principales decisiones arquitecturales tomadas para el proyecto SIM. Las vistas Lógica, Conceptual y de implementación son presentadas con mayor detalle en capítulos posteriores. Una simplificación de la vista lógica es Diseño, desarrollo e implantación del Sistema de Información Misional (SIM) de la Procuraduría General de la Nación

presentada en la Figura 4. En el proyecto SIM la vista lógica es la que mejor representa e influye en la arquitectura de solución.

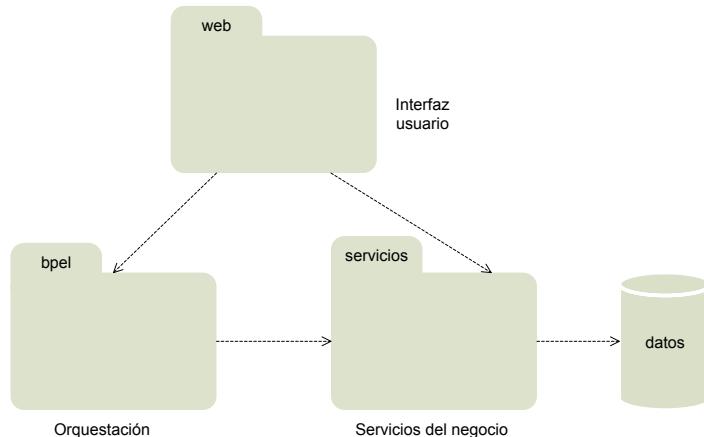


Figura 4. Vista lógica simplificada de la Arquitectura

A grandes rasgos el estilo de la arquitectura de la solución es orientado a servicios (SOA). La orquestación de los servicios se hace por medio del motor de BPEL. A su vez BPEL provee el soporte necesario a los procesos para el subsistema SIMIP. Los servicios contienen parte de la capa de negocios, y además parte de la capa de datos (persistencia-datos). Finalmente la interfaz del usuario es orientada a la Web (presentación).

A continuación se presentan las principales decisiones arquitecturales:

3.2.1.1 Decisión Principal 1: Selección de las herramientas base Oracle y J2EE.

Para el SIM se seleccionó el conjunto de aplicaciones de Oracle y J2EE como herramientas y tecnología base para el desarrollo del SIM.

Motivos conductores técnicos

Los productos de Oracle y J2EE cumplen con las características de las herramientas base consignadas dentro de los términos de referencia y la propuesta. Además el motor de procesos de Oracle (para BPEL) ofrece extensiones para BPEL. Estas extensiones son: la invocación directa a POJOS, el soporte a transacciones evitando hacer operaciones de compensación y la integración de flujos humanos.

Enfoque	<p>Se selecciono un conjunto de herramientas principales descrito a continuación:</p> <ul style="list-style-type: none"> - Selección de la plataforma J2EE (1.4) como plataforma base para el desarrollo. - Como contenedor de aplicaciones se selecciono el contenedor de aplicaciones de Oracle conocido como OC4J. - Como motor de procesos se selecciono BPM (BPEL Process Manager) de Oracle. - Como ambiente de desarrollo (IDE) se selecciono JDeveloper que permite una integración con las herramientas mencionadas arriba.
Beneficios	<ul style="list-style-type: none"> - BPM permite Integración de flujos humanos facilitando el desarrollo de código. - BPM permite la Invocación nativa de clases java (WSIF) aumentando el rendimiento y agregando transaccionalidad. - Los productos Oracle y Java pueden correr en las dos grandes plataformas comerciales Microsoft y Unix (Linux).
Desventajas	Los procesos creados en BPEL para BPM necesitan de ajustes si se desean ejecutar en otros motores de procesos.
Notas	J2EE contiene además una serie de frameworks que son nombrados adelante.

3.2.1.2 Decisión Principal 2: Estilo de arquitectura SOA

Se selecciono un estilo arquitectural SOA (Service Oriented Architecture)

Motivos conductores del negocio

La PGN es una organización orientada a los procesos.

Motivos conductores técnicos

Una arquitectura SOA permite soportar fácilmente procesos de negocio.

Enfoque	El estilo SOA, permite soportar fácilmente procesos de negocio, los
---------	---

	<p>procesos son un conjunto de actividades que crean valor transformando una entrada en una salida con mayor valor. Ambos, la entrada o la salida pueden ser artefactos y/o información y la transformación puede ser realizada por humanos maquinas o ambos.</p> <p>Los procesos de negocio son orquestados por medio de un motor de procesos y son descritos por medio de un lenguaje de alto nivel. Algunos de los pasos que componen los negocios son servicios.</p>
Beneficios	<p>Los beneficios de una arquitectura de estilo SOA son la independencia tecnológica, la independencia del ciclo de vida, el desacoplamiento, interfaces invocables, variedad de protocolos de comunicación. A continuación la explicación de las características mencionadas anteriormente.</p> <p>En resumen el estilo de arquitectura SOA permite la interoperabilidad entre diferentes componentes por medio de un hilo conductor (procesos). Los componentes serian las diferentes etapas de los procesos dentro de la <i>PGN</i>.</p> <p>Algunos componentes pueden pertenecer a otros sistemas lo que permite una integración con los mismos.</p> <p>Permite la reusabilidad de componentes en caso que el proceso cambie y la fácil integración de nuevos componentes.</p>
Desventajas	<p>El estilo de arquitectura SOA tiene un mal desempeño en sistemas de tiempo real y en general las velocidades de respuesta son relativamente bajas. En el estilo de arquitectura SOA, un controlador invoca servicios, que comúnmente son servicios Web, esa invocación necesita un tiempo para interpretar los mensajes, estos mensajes pueden ser enviados en diferentes formatos e.g. XML, mensajes binarios, etc. A diferencia con otras arquitecturas, donde la invocación es directa, los tiempos son mejores pero la flexibilidad es baja. Por ejemplo en un sistema en tiempo real, como el control de tráfico aéreo, la demora en interpretar los mensajes podría ser fatal, por lo que la arquitectura SOA está descartada para ese tipo de sistemas. Sin embargo, para el sistema de información misional, se adopta los conceptos de SOA a través del estilo de arquitectura SODA, el cual no utiliza mensajes XML sino objetos nativos JAVA lo que mitiga el problema del rendimiento, al menos, a nivel interno en la aplicación.</p>
Objetivos y motivos conductores Alcanzados	<p>El estilo de arquitectura SOA facilita la programación orientada a procesos.</p>



Notas	Aunque los tiempos de respuesta del estilo de arquitectura SOA son bajos comparados con sistemas de tiempo real, para sistemas con interfaz humana los tiempos de respuesta son los esperados. Además, utilizando una invocación de algunos servicios directamente en vez de por medio de servicios Web, la velocidad en las comunicaciones aumenta considerablemente. Esta invocación directa se hace por medio del framework WSIF (acrónimo de la lengua inglesa Web Services Invocation Framework).
-------	--

3.2.1.3 Decisión Principal 3: Selección de *Frameworks* dentro del SIM

Para el desarrollo se seleccionaron una serie de *Frameworks* para facilitar el desarrollo del Sistema.

Motivos conductores técnicos

El tiempo de desarrollo desde cero de algunas funcionalidades de software y el tiempo de realización de tareas repetitivas de desarrollo puede ser disminuido mediante el uso de *Frameworks* especializados.

Enfoque	Para el desarrollo de funcionalidades normales de los sistemas informáticos se utilizan <i>Frameworks</i> que proveen funcionalidades comunes. Los principales frameworks a utilizar son (la funcionalidad de cada framework se explica con mas detalle adelante en este documento): <ul style="list-style-type: none">- Hibernate: Para la persistencia de los objetos (en la base de datos). Hibenate es un ORM (Object Relational Mapper).- Acegi: Para los permisos de ejecución de recursos (componentes, objetos y paginas) y para autenticación y autorización de usuarios.- Struts: Para capa de presentación (interfaces con pocas funcionalidades) con los usuarios.- JSF-MyFaces: Para interfaces Web con muchas funcionalidades (eventos, listas de selección, etc.) con los usuarios. (MyFaces es una implementación de JSF)- BPEL API: Para la comunicación entre los componentes y el motor de procesos.- Log4j: Para el registro de operaciones, transacciones y errores.- Open Office Reports: Para la generación de documentos y
---------	--

	<p>plantillas.</p> <ul style="list-style-type: none"> - Spring: Para la comunicación entre la capa de negocio y de presentación. Brinda soporte al ensamblaje de la arquitectura J2EE a través de la aplicación de patrones como IoC, Singleton, Abstract Factory y Service Locutor. - AJAX: Ofrece servicios de capa de presentación complementarios a JSF para brindar procesamiento parcial de páginas web. <p>(No se mencionan los frameworks que componen la arquitectura J2EE).</p>
Beneficios	<p>Los beneficios del uso de frameworks entre otros son los siguientes:</p> <ul style="list-style-type: none"> - Aumento de la velocidad en el desarrollo. - Mayor dedicación de los desarrolladores a los componentes del negocio. - Uso de componentes (frameworks) probados. - Simplificación de la codificación en el desarrollo.
Desventajas	<p>La actualización de los componentes no se podría hacer en caso que existan cambios de funcionalidad en las nuevas versiones.</p> <p>Los posibles errores en los componentes obligarían a la actualización de las versiones.</p>
Aspectos/Consideraciones	<p>Los frameworks son bibliotecas utilizadas por el sistema pero no son parte del desarrollo o del producto SIM. Los frameworks utilizados son en principio de libre uso. Las condiciones específicas de uso están en las licencias de cada componente.</p>

3.2.1.4 Decisión Principal 4: División física de grandes funcionalidades y contingencia.

Las grandes funcionalidades de la aplicación (procesos y negocios, reportes, portal y base de datos) se ejecutarán en máquinas separadas. Se dispondrá de una máquina para contingencia. La contingencia utilizará el modelo virtualización.

Motivos conductores del negocio



El SIM debería estar la mayor parte del tiempo disponible para los usuarios.

Motivos conductores técnicos

Al separar las grandes funcionalidades se generan menos demoras en las respuestas de las aplicaciones, la solución de contingencia es más fácil de implementar.

Enfoque	<p>Cada una de las grandes funcionalidades va a estar en máquinas separadas. Las grandes funcionalidades como el portal, los reportes y la base de datos debería tener una maquina propia.</p> <p>El motor de procesos y los componentes y servicios del negocio estarán en el mismo servidor.</p> <p>El servidor de contingencia tendrá una configuración similar a los servidores principales y se utilizará un software de virtualización para tener imágenes de los otros servidores y montar la imagen en la maquina de contingencia cuando sea necesario³</p> <p>Existirán mínimo dos esquemas de base de datos, uno para reportes e históricos y uno para la información “en tiempo real”.</p>
Beneficios	<ul style="list-style-type: none">- Aumento en los tiempos de respuesta general del sistema.- Facilidad montar la contingencia en caso de fallos.

3.2.2 Cobertura de los Requerimientos en la Arquitectura

La Tabla 5 describe como se abordan los principales requerimientos no funcionales en la arquitectura. Los principales atributos de calidad del sistema son referenciados también en la tabla.

Tabla 5. Cobertura de los principales requerimientos no funcionales

No.	Requerimiento / Atributo	Descripción o escenario
1.	Interoperabilidad, Flexibilidad	
1.1	Interoperabilidad con diferentes sistemas	Se refiere a la habilidad del sistema a coexistir con sistemas externos como SIAF o SIRI.

³ El software de virtualización, la instalación, configuración y administración, es decir la contingencia en general, no están incluidos en la propuesta.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

	Impacto	Muy deseable por la evolución futura de sistema y nuevos stakeholders. Alto impacto en el diseño.
	Solución	SIM implementa interfaces bien definidas que le permiten inter-operar con las aplicaciones definidas. Estándares no propietarios y protocolos de código abierto facilitan la interoperabilidad.
	Conexiones	1.2 La interoperabilidad está relacionada (conexa) con la extensibilidad y escalabilidad.
	Motivación	SIM debe interactuar con los sistemas definidos en los términos.

1.2	Extensibilidad, Escalabilidad	El sistema se puede extender mediante la creación de nuevos procesos y servicios web asociados a esos procesos.
	Impacto	Muy deseable por la evolución futura de sistema y nuevos stakeholders. Alto impacto en el diseño.
	Solución	El estilo de arquitectura SOA permite la integración de nuevos servicios web como también nuevos servicios.
	Motivación	SIM debe permitir hacer nuevos desarrollos fácilmente.
	Observaciones	Para el tema de escalabilidad, se deberá realizar un estudio de crecimiento de información. Con ese estudio de crecimiento el administrador del SIM deberá configurar la aplicación y la infraestructura para obtener un mejor el desempeño de la mismo (<i>tuning</i>).

1.3	Estándares SOA y J2EE	El sistema aplica principios arquitecturales y de diseño especificados por SOA.
	Impacto	Alto impacto en la arquitectura del sistema y en el intercambio de datos.
	Solución	La plataforma fue diseñada para cumplir con la arquitectura SOA.
	Referencias	Uso de las siguientes tecnologías: BPEL WebServices

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

	SOAP XML XSD xHTML HTTP	
	Motivación	Re-usabilidad e interoperabilidad de los componentes de la arquitectura.

2.	Seguridad	
2.1	Mecanismo de Seguridad de Usuarios y recursos	El mecanismo de seguridad implementado tiene en consideración los derechos de acceso a recursos. Un recurso es considerado un formulario (pagina web) de la solución u otros objetos. Algunos formularios pueden tener recursos como los botones de Consultar, Crear, Modificar, Borrar, Inactivar, Cancelar (las tres últimas operaciones se usarán dependiendo del propósito de cada caso de uso).
	Impacto	Medio
	Solución	<p>La autenticación general de los usuarios (excepto para los usuarios de módulo SIC –Portal Ciudadano) se hace por medio de un LDAP, luego se valida si el usuario puede acceder al SIM por medio de una base de datos propia.</p> <p>Los usuarios tienen un perfil, los perfiles tienen roles y cada rol tiene permisos sobre los recursos.</p> <p>Los permisos de uso sobre los recursos son gestionados por medio del Framework Acegi. Este es un framework que sigue estándares de LDAP y permite conectividad a distintos tipos (Oracle Internet Directory, Open LDAP, Directorio Activo de Windows). Es importante tener en cuenta que el directorio activo de Windows no cumple 100% con el estándar LDAP, pero los frameworks J2EE como Acegi permiten realizar la conectividad y operación sobre el mismo. Además Acegi controla la Autorización en el sistema, impidiendo al acceso no permitido a recursos internos en el sistema.</p>
	Motivación	Los servicios y la aplicación tienen datos confidenciales. De tal forma que el acceso a los servicios o datos deben ser estrictamente limitados a usuarios autorizados.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

	Observaciones	Se presupone y necesita que exista para la solución una alta disponibilidad de LDAP.
--	----------------------	--

3	Desempeño	
3.1	Desempeño, Disponibilidad	Se refiere a la habilidad del responder a todos los usuarios manteniendo tiempos de respuesta uniforme según el número de usuarios conectados y en los periodos de tiempo requeridos.
	Impacto	Alto impacto en el diseño.
	Solución	A nivel de arquitectura física y de despliegue, los procesos que consumen CPU serán desplegados en máquinas separadas. Las tareas que permitan respuestas asíncronas (e.g. registro de eventos –log-) serán programadas de esa forma.
	Conexiones	
	Motivación	El sistema debe estar la mayor parte del tiempo disponible y los tiempos de respuesta deben estar acordes con los esperados.

3.2	Funcionalidad de la Interfaz del usuario	
	Facilidad de Uso e Ingreso de la Información, Validación de la Información	Se refiere a la habilidad de validar automáticamente la información ingresada por los usuarios y a la facilidad de uso del aplicativo
	Impacto	Medio impacto en el diseño.
	Solución	Se seleccionaron los frameworks JSF y Struts que combinadas con la tecnología AJAX proveen interfaces Web enriquecidas con funcionalidades de interfaces no orientadas a la Web.
	Conexiones	
	Motivación	El sistema debe evitar cargar la red con tráfico innecesario y mejorar la respuesta del sistema y a la vez proveer una experiencia agradable a los usuarios.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	--	--

4. VISTAS

4.1 Vista Conceptual

4.1.1 Descripción

La vista conceptual es usada para definir los requerimientos funcionales y la visión que los usuarios del negocio tienen de la aplicación y describir el modelo de negocio que la arquitectura debe cubrir.

En este caso particular, la arquitectura base propuesta consta de siete componentes de servicios, representados en la Figura 5.

4.1.2 Estructura General

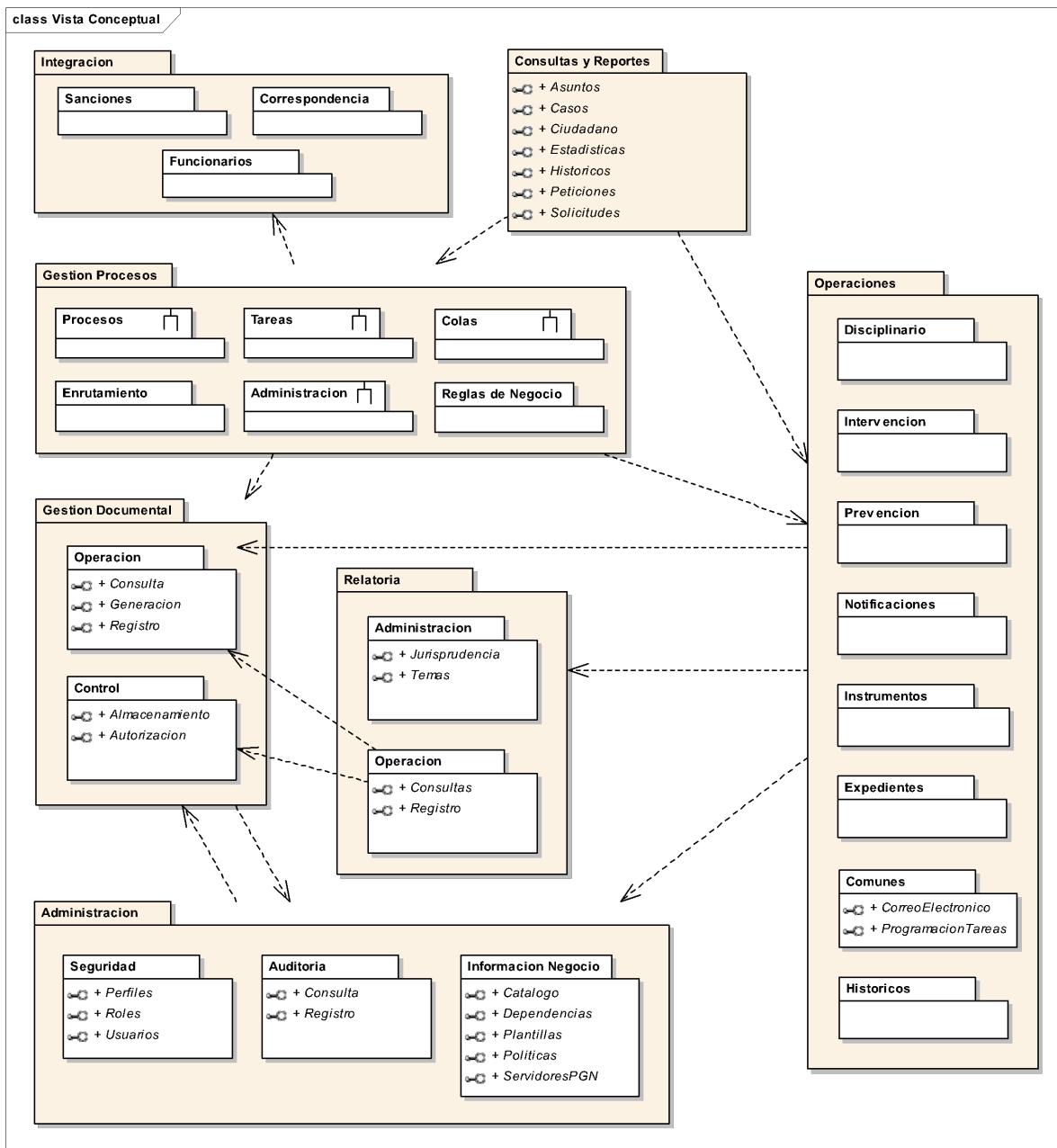


Figura 5. Vista Conceptual

En la vista conceptual se definieron siete grandes paquetes agrupando tipos de funcionalidades; los grandes paquetes de esta vista son:

- Administración
- Gestión Documental

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

- Relatoría
- Consultas y reportes
- Integración
- Gestión de Procesos
- Operaciones

Los paquetes de la lista son definidos con un mayor detalle en las siguientes secciones.

4.1.3 Paquetes de la vista

4.1.3.1 Paquete “Administración”

4.1.3.1.1 Presentación primaria

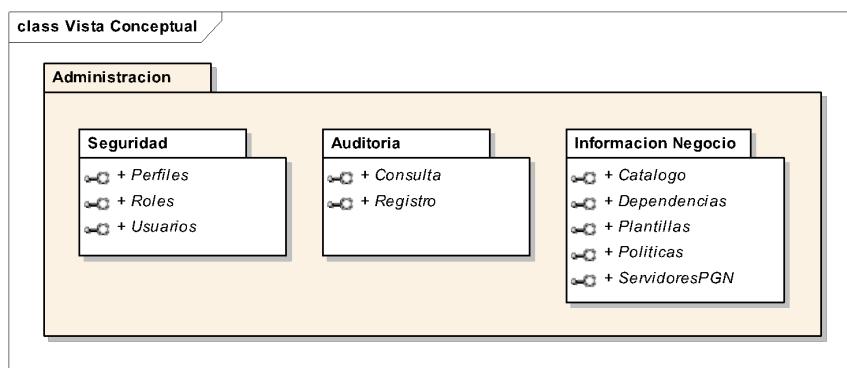


Figura 6. Paquete de Administración

El paquete de “Administración” es el encargado de la administración de usuarios, roles y perfiles, las consultas de auditoria y registro y la gestión de tablas de uso general en el sistema.

4.1.3.1.2 Catálogo de Elementos

4.1.3.1.2.1 Elementos

Los elementos del paquete de Administración son:

- Seguridad: Comprende el manejo de perfiles roles y usuarios.
- Auditoria: Comprende la información y manejo relacionada con el registro y consulta de eventos del sistema.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

- Información del negocio: Comprende la información y el manejo de los datos propios del negocio, como las políticas, dependencias y servidores de la PGN.

4.1.3.1.2.2 Comportamiento

Las funcionalidades de seguridad como la manejo de usuarios, roles y perfiles se realizarán en conjunto con el LDAP institucional. Los usuarios autorizados para SIM estarán en una base de datos local y la autenticación se hará contra el LDAP institucional. Los roles y perfiles se almacenarán localmente para luego asociarlos a un usuario o grupo de usuarios y a su vez a un grupo de recursos permitidos para ese rol-perfil.

4.1.3.1.3 Paquetes relacionados

El paquete de *administración* se relaciona con los paquetes *Gestión documental*, y *Operaciones*. Al ser la arquitectura de un estilo SOA cuando hay comunicación se realiza por servicios Web o innovación directa de clases e indirectamente por medio de los datos almacenados en la base de datos (por persistencia).

4.1.3.2 Paquete “Gestión Documental”

4.1.3.2.1 Presentación primaria

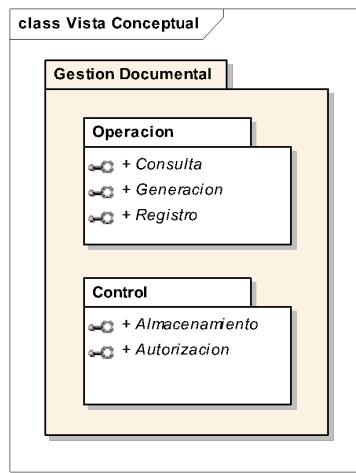


Figura 7. Gestión documental

El paquete de gestión documental agrupa las funcionalidades relacionadas con la gestión de documentos creados en el SIM de la PGN.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	--

4.1.3.2.2 Catálogo de Elementos

4.1.3.2.2.1 Elementos

El paquete de gestión documental tiene dos elementos:

- Operación: Relacionado directamente con la creación, consulta y registro de documentos dentro del SIM.
- Control: Relacionado con las funcionalidades de almacenamiento de los documentos y las autorizaciones sobre los documentos que tienen los usuarios.

4.1.3.2.2.2 Comportamiento

La generación de los documentos se hará por medio de plantillas, las plantillas generan documentos en formato OpenOffice (ODF).

4.1.3.2.3 Paquetes relacionados

El paquete de “Gestión Documental” está relacionado con el paquete de Operaciones y el paquete de Relatoría. En el caso del paquete de Operaciones, la mayoría de los elementos de ese paquete interactúan con la Gestión Documental en lo que tiene que ver con la generación de autos, comunicaciones, notificaciones y todos los demás tipos de documentos que sea posible ser generados mediante plantillas.

El paquete Relatoría es el cliente por excelencia del paquete de Gestión Documental, ya que las funcionalidades de los dos paquetes son complementarias.

4.1.3.3 Paquete “Relatoría”

4.1.3.3.1 Presentación primaria

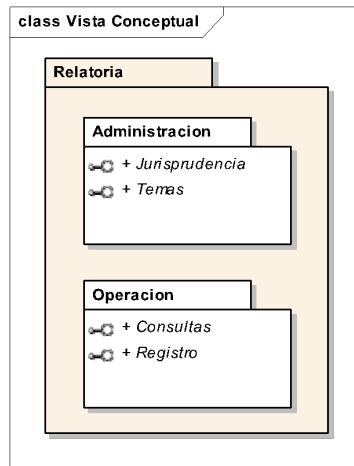


Figura 8. Vista Conceptual Paquete de Relatoría

El paquete de Relatoría agrupa las funcionalidades para la clasificación, consulta y registro de documentos en lo relativo a la clasificación de documentos, administra los temas de clasificación y los descriptores y restrictores.

4.1.3.3.1.1 Elementos

Los dos elementos del paquete de Relatoría son de administración y operación. El elemento de administración está relacionado con la clasificación de documentos, y el tema de operación con las consultas y registro de los documentos. El elemento Operación y todo el paquete se soporta el paquete de Gestión Documental.

4.1.3.4 Paquete “Integración”

4.1.3.4.1 Presentación primaria

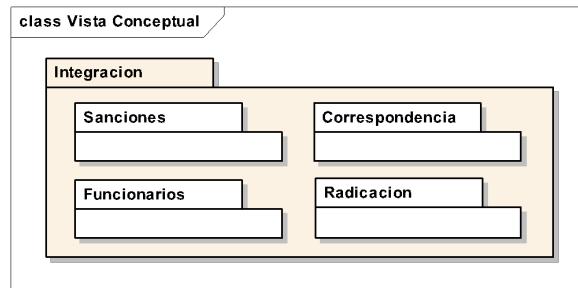


Figura 9. Vista Conceptual Paquete de Integración

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

El paquete de Integración contiene los elementos con las funcionalidades para la integración con otras aplicaciones dentro de la *PGN* como lo son SIRI, SIAF.

4.1.3.4.2 Catálogo de Elementos

4.1.3.4.2.1 Elementos

Los elementos del paquete de Integración son:

- Sanciones: El elemento de sanciones agrupa las funcionalidades de comunicación con el aplicativo SIRI de la *PGN*.
- Correspondencia: Para la comunicación con SIAF en lo referente a la recepción y envío de correspondencia que tenga que entrar o salir de SIM por medio del SIAF. No tiene que ver con correo electrónico a no ser que el correo sea radicado en SIAF.
- Funcionarios: La aplicación SIAF es consultada y/o informada con la información de los funcionarios (disponibilidad) de la *PGN*, potenciales usuarios del SIM.
- Radicación: La aplicación SIAF es consultada sobre la información de radicaciones..

4.1.3.4.3 Paquetes relacionados

El paquete Integración se relaciona con el paquete Gestión Procesos.

4.1.3.5 Paquete “Gestión de procesos”

4.1.3.5.1 Presentación primaria

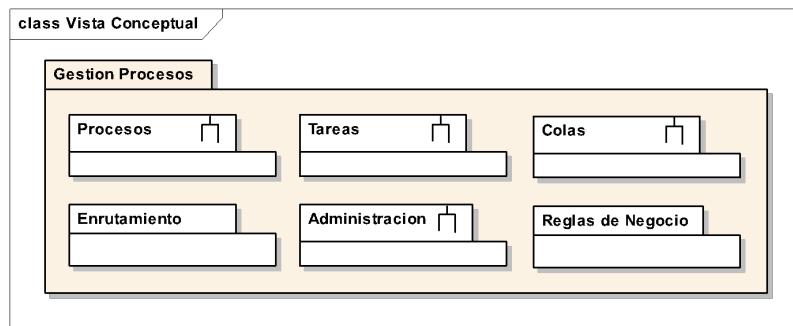


Figura 10. Vista conceptual Gestión de Procesos

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <small>Soluciones y Servicios IT</small>
---	---	---

El paquete *Gestión Procesos* contiene las funcionalidades que permiten la orquestación de los procesos (de la *PGN*), consultas sobre el estado de los procesos y todas las tareas relacionadas con la gestión de procesos que serán relacionadas a los elementos del paquete.

4.1.3.5.1.1 Elementos

El paquete Gestión Procesos tiene seis elementos relacionados a continuación:

- Procesos: Este elemento es el encargado de llevar el hilo conductor de los procesos.
- Tareas: Tiene las funcionalidades de consulta y asignación de tareas dentro de los procesos a los funcionarios correspondientes en la *PGN*.
- Colas: Es el elemento en donde se guardan las tareas de los usuarios.
- Enrutamiento: Componene que tiene la lógica de reparto hacia los usuarios o estaciones.
- Administración: Modulo donde se hace el monitoreo de los proceso de negocio.
- Reglas de negocio: Contiene las reglas del negocio que pueden modificar el flujo de los negocios.

4.1.3.6 Paquete “Operaciones”

4.1.3.6.1 Presentación primaria

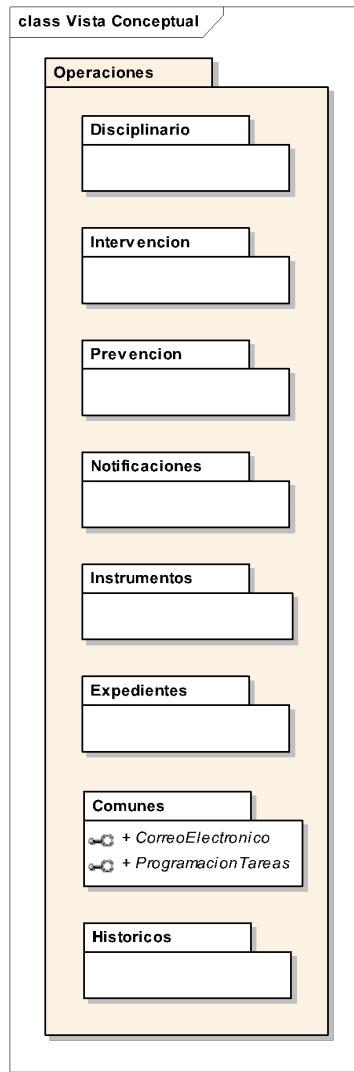


Figura 11. Vista conceptual Paquete Operaciones

El paquete Operaciones agrupa las funcionalidades propias del negocio de la PGN, es decir los frentes misionales (*i.e.* Disciplinario, Intervención y Prevención –DIP-) y algunas funcionalidades de apoyo a los frentes misionales.

4.1.3.6.2 Catálogo de Elementos

4.1.3.6.2.1 Elementos

Los elementos del paquete Operaciones tiene elementos relacionados directamente con la misión como también de apoyo a la misión; los elementos son:

 PROCURADURÍA GENERAL DE LA NACIÓN	<p>Documento de Arquitectura de la Solución V 0.2</p>	 synapsis Soluciones y Servicios IT
---	---	---

- Disciplinario: Agrupa las funcionalidades relacionadas los procesos disciplinarios de la PGN.
- Intervención: Concentra las funcionalidades relacionadas los procesos de intervención de la PGN.
- Prevención: Contiene las funcionalidades relacionadas los procesos preventivos de la PGN.
- Notificaciones: Agrupa las funcionalidades de notificación a funcionarios y/o personas dentro de los procesos.
- Instrumentos: Concentra las funcionalidades de Instrumentos de los procesos DIP dentro de la PGN, los principales grupos de instrumentos son los de Recursos, Traslado de competencias, Causales, Asignación de funcionarios, Prorroga o Suspensión y Otros Instrumentos.
- Expedientes: Relacionado con la gestión básica de los expedientes dentro del SIM.
- Historicos: Relacionado con los casos cerrados disponibles para consulta.
- Comunes: Contiene las funcionalidades comunes a todos los módulos, como la interfaz de correo electrónico.

4.1.3.6.3 Paquetes relacionados

El paquete Operaciones contiene la mayoría de los componentes de núcleo del negocio tiene comunicación con todos los demás paquetes. La principal relación es con el paquete Gestión Procesos que orquesta los “servicios” y/o componentes que integran el paquete Operaciones.

4.1.3.7 Paquete “Consultas y reportes”

4.1.3.7.1 Presentación primaria



Figura 12. Vista Conceptual Paquete Consultas y Reportes

Diseño, desarrollo e implantación del Sistema de Información Misional (SIM) de la Procuraduría General de la Nación

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

El paquete Consultas y Reportes agrupa las funcionalidades relacionadas con las consultas sobre los datos.

4.2 Vista Lógica

4.2.1 Descripción

La vista lógica es una visión más concreta de lo expuesto en la vista conceptual. A aquí se muestran los componentes principales de diseño y sus relaciones de forma independiente de los detalles técnicos. También esta vista es independiente de cómo la funcionalidad será implementada en la plataforma de ejecución. Debido a la naturaleza de la arquitectura aquí presentada en la vista lógica incluimos nombres de las tecnologías y frameworks, que en la vista de implementación se presentarán a mayor profundidad.

4.2.2 Estructura General

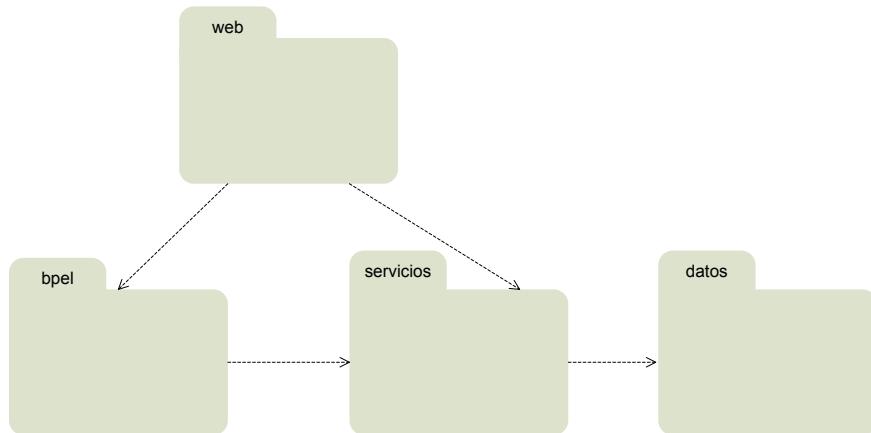


Figura 13. Vista Lógica SIM (simplificada)

La Figura 13 representa la vista Lógica del SIM, en el ANEXO1 se presenta la Vista Lógica del SIM contiene cuatro capas: web, bpel, servicios y datos. A continuación se explica a grandes rasgos cada una de las capas:

- **web**: La capa web encapsula los servicios de presentación en módulos funcionales definidos por el negocio como "subsistemas", y cuya cohesión ofrece claridad sobre los actores que interactúan con dichos servicios.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

- **bpel:** En BPEL se despliegan los procesos de negocio, conformados por la orquestación de los servicios expuestos en el paquete de Servicios. Los procesos de negocio reflejan las funciones misionales Disciplinaria, Intervención y Prevención.
- **servicios:** El paquete de servicios contiene los servicios de negocio organizados por módulos funcionales, dependiendo del objetivo de cada uno. Estos servicios pueden ser llamados desde un cliente externo (por ejemplo capa web), ser publicados para consumo externo (por ejemplo los de integración como servicios Web), o ser orquestados dentro del motor de procesos de BPEL.
- **datos:** La capa de datos y/o persistencia de la aplicación.

4.2.3 Modelo Arquitectónico

La arquitectura lógica propuesta sigue los parámetros de una arquitectura orientada a servicios (SOA) de tal manera que se garantice la extensibilidad del SIM, permitiendo su interconexión con otros sistemas de la PGN, esto es que el sistema permita a otras entidades consumir la información del SIM, y permita a SIM extraer información relevante, que es generada en otros sistemas. El estilo de arquitectura SOA fue presentado en la sección 3.2.1.

Todos los servicios que poseen interacción con el cliente tienen una capa de presentación que se encarga de realizar el puente entre el cliente y éstos servicios que pertenecen a la capa de negocio, es por esto que el modelo vista controlador, que se entiende como un patrón de diseño que se emplea para separar la presentación de la lógica de negocio, constituye otro de los elementos de diseño incluidos en la vista lógica de la arquitectura.

4.2.4 Paquetes de la vista

En las siguientes secciones se presentan los paquetes de la vista lógica. La Figura 13 presenta una vista lógica simplificada. En las imágenes de los paquetes presentados a continuación se presenta una miniatura de la vista lógica y un diagrama detallado de cada paquete; en la miniatura el paquete correspondiente esta de color verde.

4.2.4.1 Paquete “web”

4.2.4.1.1 Presentación primaria

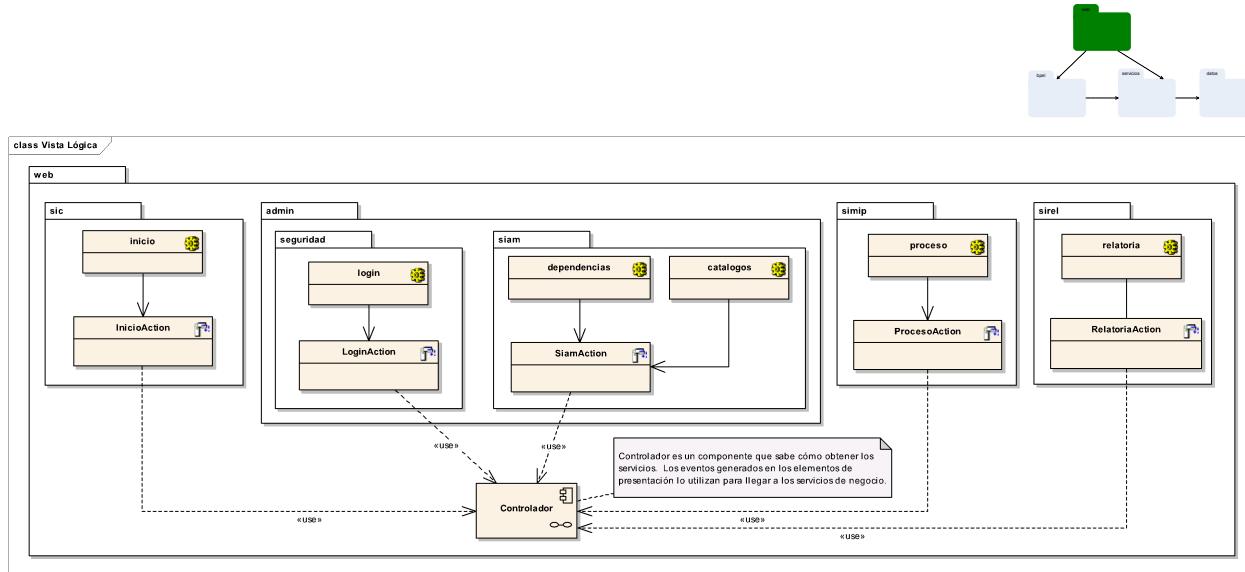


Figura 14. Vista Lógica Paquete web

La capa web encapsula los servicios de presentación en módulos funcionales definidos por el negocio como subsistemas: sic, admin, simip y sirel. Además existe un elemento controlador que provee los servicios (*service provision*) a los demás paquetes.

4.2.4.1.2 Catálogo de Elementos

Los elementos de este paquete tienen dos componentes, un elemento de presentación que es una página (e.g. JSF, JSP) y un componente (*servlet*) que se encarga de buscar el servicio respectivo a través del controlador.

4.2.4.1.2.1 Elementos

- **Elemento admin:** Se encarga de la administración de usuarios y configuraciones del sistema. Dentro del paquete admin se encuentra el manejo de seguridad y el subsistema SIAM, que administra la información relacionada con el negocio para dar apoyo a la misión.
- **Elemento sic:** Corresponde a la interfaz que es presentada los ciudadanos.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	--

- Elemento **simip**: Corresponde a la interfaz relacionada con el modulo de procesos, interactúa principalmente con los procesos de negocios desde la capa web.
- Elemento **sirel**: Corresponde a la interfaz con las funcionalidades para relatoría, además tiene algunas funcionalidades de la gestión documental.
- Elemento **controlador**: Es un componente que sabe cómo obtener los servicios. Los eventos generados en los elementos de presentación lo utilizan para llegar a los servicios de negocio. Cada servicio tiene una interfaz del servicio que define las operaciones que ofrece, definiendo los llamados contratos. Internamente cada servicio puede tener interfaces que definen la cohesión del mismo, y pueden ser diseñados como componentes de negocio. La interfaz del servicio heredaría de estas interfaces internas para ofrecer el conjunto total de operaciones.

4.2.4.1.2.2 Relaciones

Los elementos *simip*, *admin.*, *sirel* y *sic* de este paquete utilizan el elemento *controlador* como proveedor de servicios, el elemento controlador es una implementación del patrón ServiceLocator.

4.2.4.1.3 Bases de la Arquitectura

Service Locator: Para el elemento controlador se utilizo el patrón “Service Locator”, en resumen el patrón se encarga de proveer un servicio al cliente que lo solicita (recursos listos para utilizar), además de proveer el servicio, también crearía de ser necesario el contexto, la carga de clases y las demás acciones necesarias para utilizar el servicio y/o recurso.

La interfaz con el usuario fue desarrollada mediante el framerwork JSF que utiliza como principal patrón de diseño el MVC (Joshi 2005).

4.2.4.2 Paquete “bpel”

4.2.4.2.1 Presentación primaria

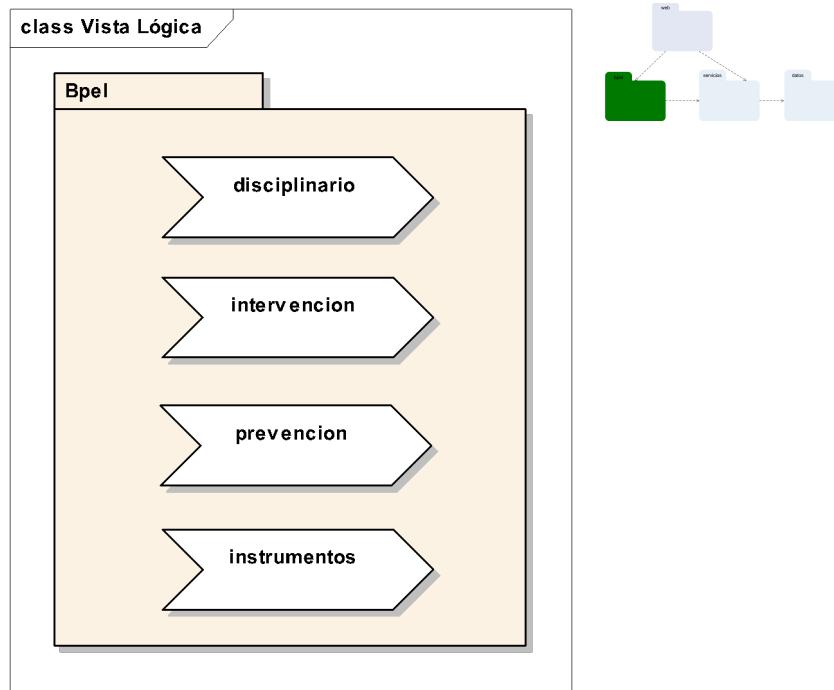


Figura 15. Vista Lógica del Paquete bpel

El paquete Bpel hace referencia a todos los procesos creados en el lenguaje BPEL. Existen grupos de procesos como grupos de procesos dentro de la *PGN*. Los grupos son disciplinario, intervención y prevención mas los procesos de instrumentos.

4.2.4.2.2 Catálogo de Elementos

El paquete BPEL tiene cuatro elementos:

- Disciplinario
- Intervención
- Prevención
- Instrumentos

Cada paquete representa los grupos de procesos (con el mismo nombre) propios de la misión de la *PGN*.

4.2.4.2.3 Bases de la Arquitectura

Este paquete corresponde al motor de procesos de BPEL (BPM) para la orquestación de servicios Web.

La orquestación de servicios Web se basa en un modelo centralizado en el cual las interacciones no se realizan directamente entre los servicios Web sino que existe una entidad encargada de definir la lógica de interacción.

El lenguaje BPEL permite definir esa lógica de orquestación entre los diferentes servicios Web. Para que esto sea posible los diferentes partners (servicios Web) tienen que proporcionar los archivos WSDL que contienen la información necesaria para poder invocarlos. Finalmente, la entidad encargada de la orquestación es el conductor, que no es otro que el motor de ejecución BPEL.

El BPM será el encargado de controlar la orquestación definida en el archivo BPEL. Este último se publicará también como un servicio Web (con su correspondiente archivo WSDL) para que los clientes del SIM puedan lanzar el proceso.

El BPM de Oracle permite además orquestar una variedad de componentes (e.g. POJOS, EJB, bases de datos, etc) como si fuesen servicios Web.

4.2.4.2.4 Paquetes relacionados

Este paquete tiene una fuerte relación con el paquete servicios, ya que coordina los servicios que allí se encuentran. Además tiene relación con el paquete Web, en lo que tiene que ver con la intervención de los usuarios en los procesos.

4.2.4.3 Paquete “Servicios”

4.2.4.3.1 Presentación primaria

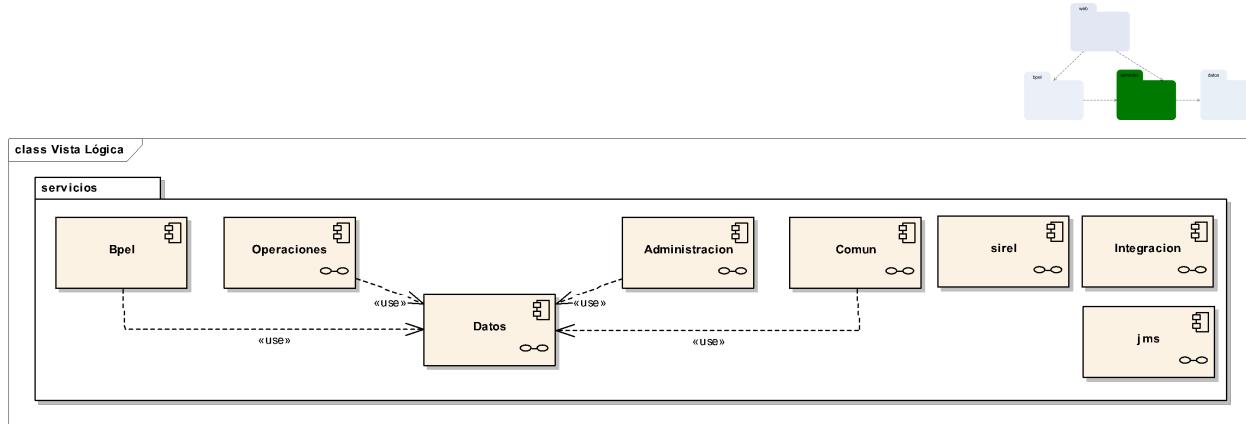


Figura 16. Vista Lógica Paquete servicios.

Como se mencionó al inicio de la vista el paquete servicios contiene los servicios de negocio organizados por módulos funcionales, dependiendo del objetivo de cada uno. Estos servicios pueden ser llamados desde un cliente externo (por ejemplo capa web),

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

ser publicados para consumo externo (por ejemplo los de integración como web services), o ser orquestados dentro del motor de procesos de BPEL.

4.2.4.3.2 Catálogo de Elementos

4.2.4.3.2.1 Elementos

- Bpel: Hace referencia a los servicios que se comunican directamente con el motor de procesos hacia el interior del paquete servicios.
- Operaciones: Corresponde a los servicios que tienen funcionalidades del núcleo del negocio.
- Administración: Contiene las funcionalidades (servicios) propias de administración del negocio (creación de usuarios, listas, plantillas, configuraciones, etc.).
- Común: Provee funcionalidades comunes para todos los demás elementos como correo electrónico.
- Datos: Contiene todos los servicios de persistencia de la aplicación.
- Integración: Contiene las funcionalidades para comunicación con los sistemas externos al SIM.
- Jms: Contiene los servicios para mensajes asincrónicos, se utiliza principalmente para el envío de correos electrónicos y registro de eventos.

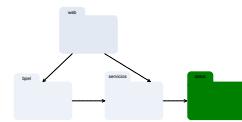
4.2.4.3.3 Bases de la Arquitectura

La gran mayoría de servicios en este paquete se pueden asimilar a objetos Java (POJOS, javabeans, etc) que exponen una interfaz y carecen de estado.

Para el elemento datos es utilizado el patrón de diseño DAO, soportado por bibliotecas de los frameworks Spring e Hibernate. Básicamente el elemento datos extiende una serie de interfaces para manipulación de datos dependiendo de las necesidades de la aplicación.

4.2.4.4 Paquete “datos”

4.2.4.4.1 Presentación primaria



 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

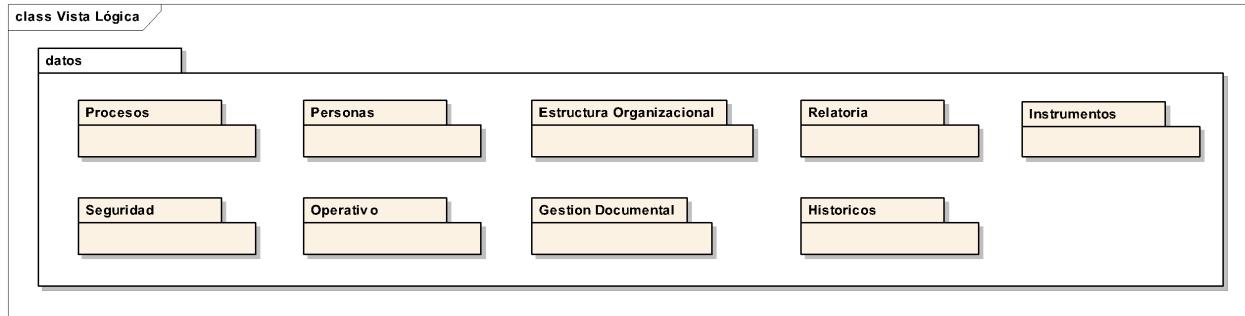


Figura 17. Vista Lógica Paquete datos

El paquete de datos representa el modelo de dominio de la aplicación. Información detallada sobre el modelo de datos se encontrara en la sección de Modelo de Dominio(ver sección 6).

4.2.4.4.2 Catálogo de Elementos

4.2.4.4.2.1 Elementos

Los elementos del paquete de datos son:

- Procesos: Representa la información relativa a los casos dentro de la *PGN*.
- Personas: Representa la información de las personas dentro de los casos.
- Estructura Organizacional: Representa la *PGN* desde el punto de vista de cargos, grupos y dependencias.
- Relatoria: Se refiere a la información para la clasificación de los documentos de relatoria (“tesauro”) y al inventario de los documentos de relatoría.
- Instrumentos: Contiene la información de los instrumentos de cada caso.
- Seguridad: Se refiere a la información de seguridad de los usuarios (permisos sobre recursos, perfiles y roles)
- Operativo: Representa las lista de tareas y datos operativos de los procesos asignados a los servidores de la *PGN*.
- Gestión Documental: Relativo a las plantillas de generación de documentos y a los documentos generados con las plantillas dentro de los procesos de la *PGN*.
- Historicos: Relacionado con los casos cerrados disponibles para consulta.

4.2.4.4.3 Paquetes relacionados

El paquete de datos esta relacionado directamente con el paquete de servicios (gracias al elemento Datos que es una implementación del patrón de diseño DAO).

4.2.4.5 Frameworks utilizados transversalmente en la aplicación

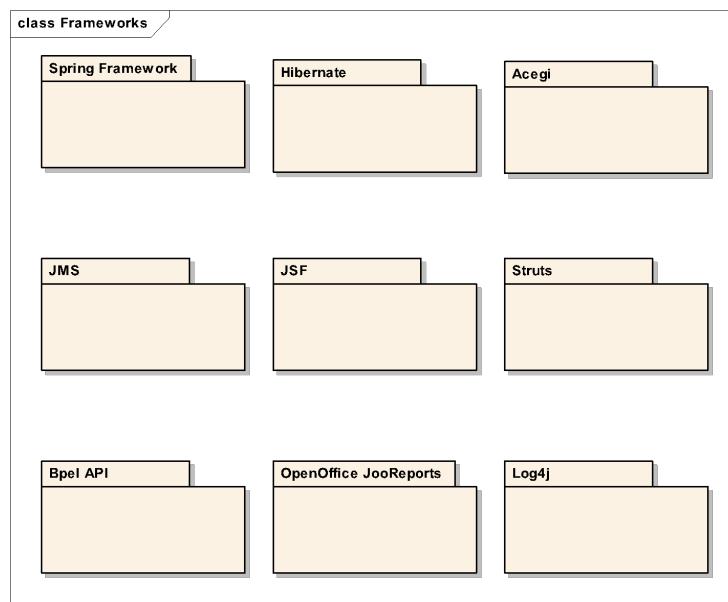


Figura 18. Frameworks Utilizados

4.2.4.5.1 Catálogo de Elementos

- Acegi: Acegi es un framework de seguridad, con facilidades de desarrollo en aplicaciones que usan Spring. Acegi provee funcionalidades para autenticación autorización, control de acceso a instancias de objetos (entre otros). Además Acegi permite la conectividad con servidores de LDAP.
- Hibernate: El framework Hibernate es una herramienta que automatiza la tarea de persistencia de los objetos, permitiendo la comunicación con varios tipos de proveedores de datos y ofrece soporte para obtener la conexión directa a la base de datos (para el caso en el cual se quiera realizar la inclusión de procedimientos almacenados). Posee una serie de herramientas para generar automáticamente archivos de configuración, creación de modelo físico, y transaccionalidad.

- JMS: La API de Servicios de Mensajería de Java (también conocido por sus siglas JMS), es un estandar de mensajería que permite a los componentes de aplicaciones basados en la plataforma de Java 2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona. El servicio de mensajería también es conocido como Middleware Orientado a Mensajes (MOM por sus siglas en inglés) y es una herramienta universalmente reconocida para la construcción de aplicaciones empresariales.
- JSF: JavaServer Faces (JSF) es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL. JSF incluye:
 - Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad
 - Un conjunto por defecto de componentes para la interfaz de usuario
 - Dos bibliotecas de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP
 - Un modelo de eventos en el lado del servidor
 - Administración de estados
 - Beans administrados
- Log4j: Log4j es un API para manejar el registro (log) de operaciones en programas. Log4j es uno de los componentes del proyecto Jakarta (<http://jakarta.apache.org/log4j/>). Log4j permite configurar cinco prioridades diferentes de registro, que de mayor a menor son: DEBUG, INFO, WARN, ERROR, FATAL, aunque de ser necesario también es posible crear prioridades propias. Log4j además permite registrar la información de log en varios formatos, desde archivos planos hasta bases de datos.
- OpenOffice & JooReports :
 - OpenOffice es el Formato de Documento Abierto para Aplicaciones Ofimáticas de OASIS (en inglés, OASIS Open Document Format for Office Applications), también referido como OpenDocument u ODF, es un formato de archivo estándar para el almacenamiento de documentos

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

ofimáticos tales como hojas de cálculo, memorandos, gráficas y presentaciones. Su desarrollo ha sido encomendado a la organización OASIS y está basado en un esquema XML inicialmente creado por OpenOffice.org. Open Document es el primer estándar para documentos ofimáticos implementado por distintos competidores, visado por organismos de estandarización independientes y susceptible de ser implementado por cualquier proveedor.

- JOORReports (Open Source Java/OpenOffice Document and Report Generator) es una solución para crear documentos en le formato OpenDocument desde plantillas. Los documentos pueden ser editados en el programa "OpenOffice.org Writer". Los documentos creados pueden ser convertidos a otros formaros como PDF, Word y RTF.
- Spring Framework: Spring permite el apoyo de construcción de la aplicación en varias capas; para el despliegue de servicios es apropiado ya que permite la definición de un contrato y luego de una implementación. De igual forma ayuda a mantener la construcción de aplicaciones de forma consistente, como intermediario de componentes que se concentran en un solo aspecto, con el fin de tomar lo mejor de cada uno de ellos y lograr apoyo para la construcción de una arquitectura flexible.

Para la parte de acceso a datos, se usa el mecanismo de persistencia Hibernate; Sin embargo Spring posee una abstracción de esta funcionalidad mediante la clase HibernateDaoSupport (incluida en una funcionalidad de Spring denominada Spring DAO). Por medio de ella, se pueden realizar operaciones transaccionales que a partir de dialectos son traducidas a operaciones DML del motor específico mediante dialectos (series de clases ya existentes que traducen operaciones sobre objetos a operaciones específicas del motor). Sin embargo, en casos excepcionales en los cuales se requiera llamar a un Procedimiento almacenado o se necesite hacer alguna funcionalidad que solo se puede lograr con llamados directos a funciones específicas del proveedor de datos, se puede incluir la especialización de la clase JdbcDaoSupport u obteniendo la conexión a la base de datos mediante una consulta a Spring y solicitando la conexión directa a Hibernate.

- Struts: Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC (Modelo Vista Controlador) bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Entre las características de Struts se pueden mencionar:

- Configuración del control centralizada.



- Interrelaciones entre acciones y página u otras acciones se especifican por tablas XML en lugar de codificarlas en los programas o páginas.
- Componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
- Bibliotecas de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas JSP.
- Struts contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página (en javaScript) hasta las validaciones de fondo hechas a nivel de las acciones.
- Struts permite que el desarrollador se centre en el diseño de aplicaciones complejas como una serie simple de componentes del Modelo y de la vista intercomunicados por un control centralizado. Diseñando de esta manera se debe obtener una aplicación más consistente y más fácil de mantener.

4.3 Vista Física

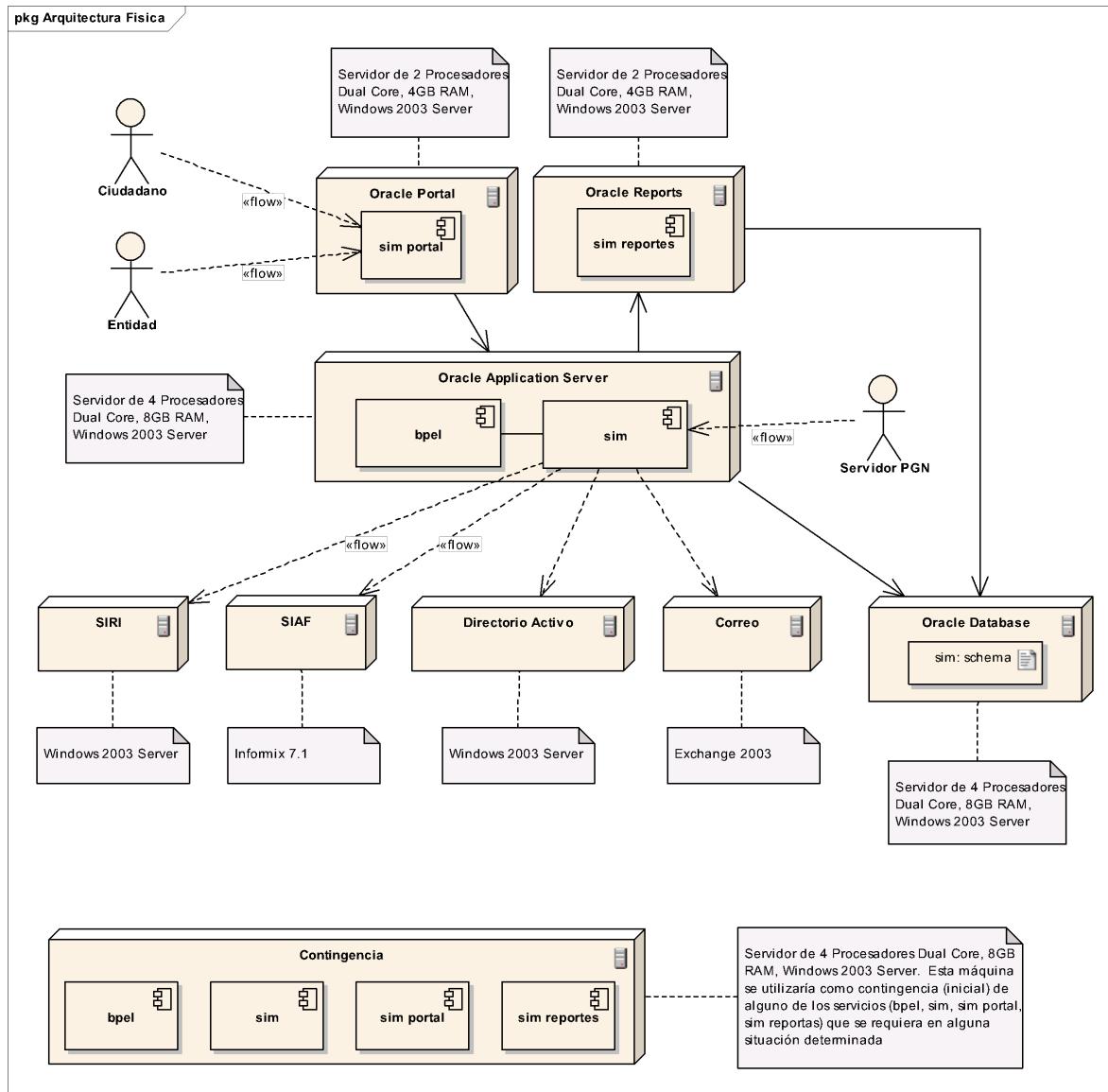


Figura 19. Arquitectura Física

La arquitectura física de SIM esta soportada directamente por cinco servidores, que en principio son físicos (aunque no se descarta la posibilidad que en algún momento uno pueda ser lógico). Indirectamente SIM interactúa con otros servidores. La descripción de cada uno de los cinco servidores es:

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis <i>Soluciones y Servicios IT</i>
---	---	--

- Servidor "Oracle Portal": En este servidor corre el programa Oracle Portal, y su principal uso es para atender los usuarios externos de la Procuraduría, es decir los ciudadanos. Muchas de las funcionalidades del modulo SIC de SIM están soportadas por este servidor.
- Servidor "Oracle Reports": Esta maquina esta encargada de soportar los reportes para el SIM. Se escogió esta solución para no cargar la maquina de procesos.
- Servidor "Oracle Application Server": En este servidor corre el contenedor de aplicaciones de Oracle (Oracle Application Server - OC4J). OC4J tiene el núcleo del negocio del SIM. Sobresale el paquete BPEL y el paquete SIM.
- Servidor "Oracle Database": Contiene el motor de base de datos del sistema SIM. Esta base de datos se encarga de la persistencia de los datos del los procesos como de la persistencia mismo de los procesos. Se utilizaran dos esquemas (bases de datos), una exclusiva para los procesos otra para los reportes e históricos.
- Servidor "Contingencia": Esta máquina se utilizará como servidor de contingencia de alguno de los otros servidores/servicios en caso de fallo. Se propone el uso de un software de virtualización para la obtención de imágenes de los otros servidores y en caso de alguna eventualidad, la imagen sea montada en el servidor de contingencia.

Otros aspectos físicos

- Dentro de la figura también existen otros maquinas que interactúan con el SIM : El servidor de correos, el sistema SIRI, el sistema SIAF, el servidor de Directorio Activo (LDAP) que tiene la base de datos de usuarios central de la PGN.
- Aunque en la Figura 19 se presentan los servidores utilizando el sistema Operativo Windows, es posible que las aplicaciones corran en un ambiente Unix (e.g. Red Hat Enterprise)

5.

RELACIONES ENTRE LAS VISTAS

Cada una de las vistas especificadas en la sección 4 provee una perspectiva y manejo del diseño, diferentes sobre un sistema, y cada una es válida y útil por sí sola, no son independientes. Los elementos de una vista estarán relacionados con elementos de otras vistas, y es necesario discutir dichas relaciones.

Esta sección describe las relaciones existentes entre las vistas descritas en la sección 4.

5.1 Relaciones Generales

5.2 Relaciones Vista Conceptual y Lógica

En esta sección se explica la relación que existe entre la vista conceptual y la vista lógica, es decir, la manera en que la vista lógica soluciona los requerimientos arquitectónicos planteados en la vista conceptual.

La Vista Conceptual que es presentada en la Figura 5 tiene componentes distribuidos transversalmente dentro de la Vista Lógica que es presentada en la Figura 13, que a su vez es una composición de los siguientes paquetes: bpel (Figura 15), servicios (Figura 16), datos (Figura 17) y web (Figura 14).

En general todos los paquetes de la vista Conceptual tienen una representación clara en la vista de Lógica. A excepción de los paquetes de Integración, Operaciones y Consultas y reportes las funcionalidades de estos paquetes están distribuidas a lo largo de todos los elementos de la vista lógica.

Para mostrar las relaciones es presentada la imagen del paquete de la vista conceptual y una miniatura de toda la Vista Conceptual (con dicho paquete de color verde); seguido se muestra una imagen con los paquetes de la Vista Lógica que mayor relación tienen con el paquete de la vista conceptual. A continuación se presentan como cuatro de los siete paquetes de la Vista Conceptual se relacionan dentro de los 4 paquetes de la Vista Lógica.

5.2.1 Relación del Paquete Administración de vista Conceptual

Vista Conceptual paquete de Administración

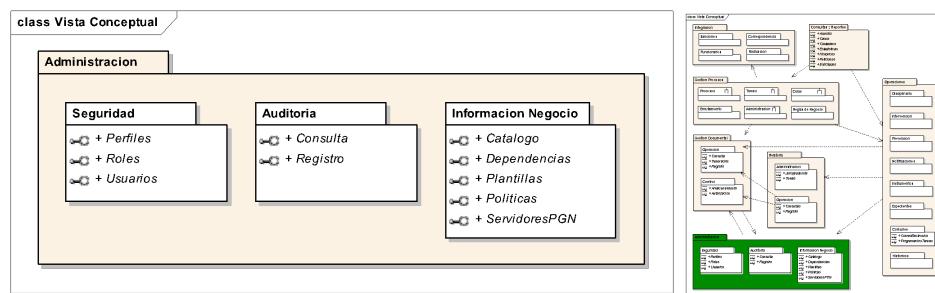


Figura 20. Vista Conceptual paquete de Administración

Relaciones con la Vista Lógica del paquete de Administración-Conceptual

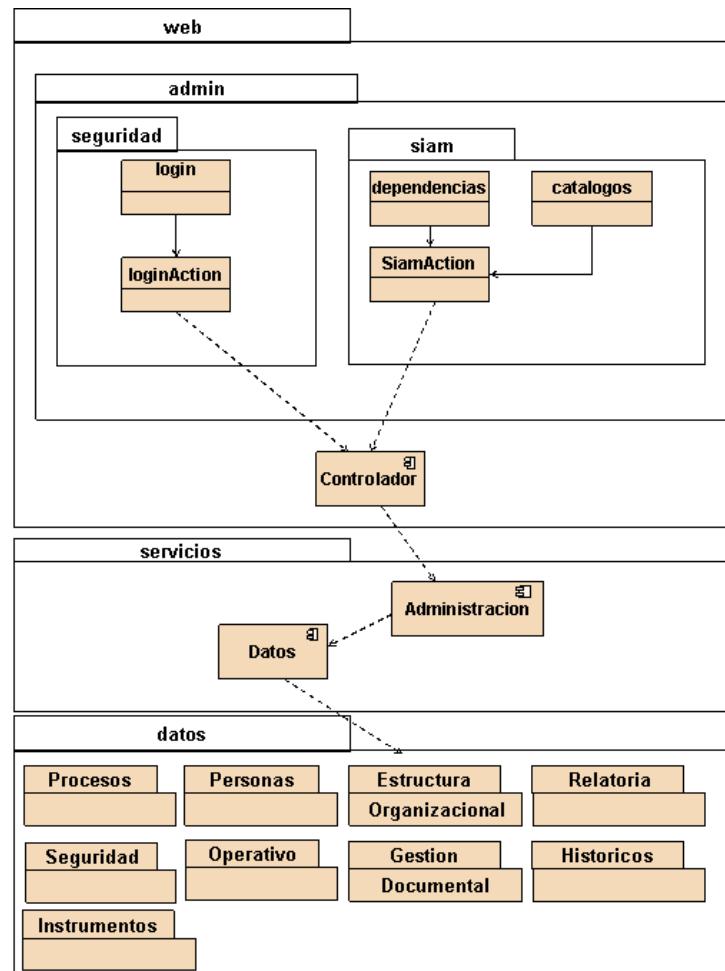


Figura 21. Relaciones con la Vista Lógica del paquete de Administración-Conceptual

5.2.2 Relaciones del Paquete Gestión Documental de la vista Conceptual

Vista Conceptual paquete de Gestión documental

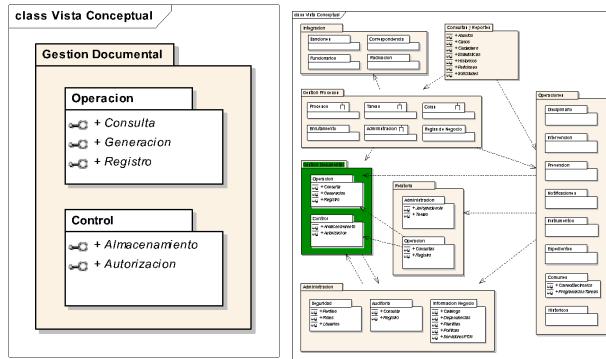


Figura 22. Vista Conceptual paquete de Gestión documental

Relaciones del paquete de Gestión Documental en la Vista Lógica

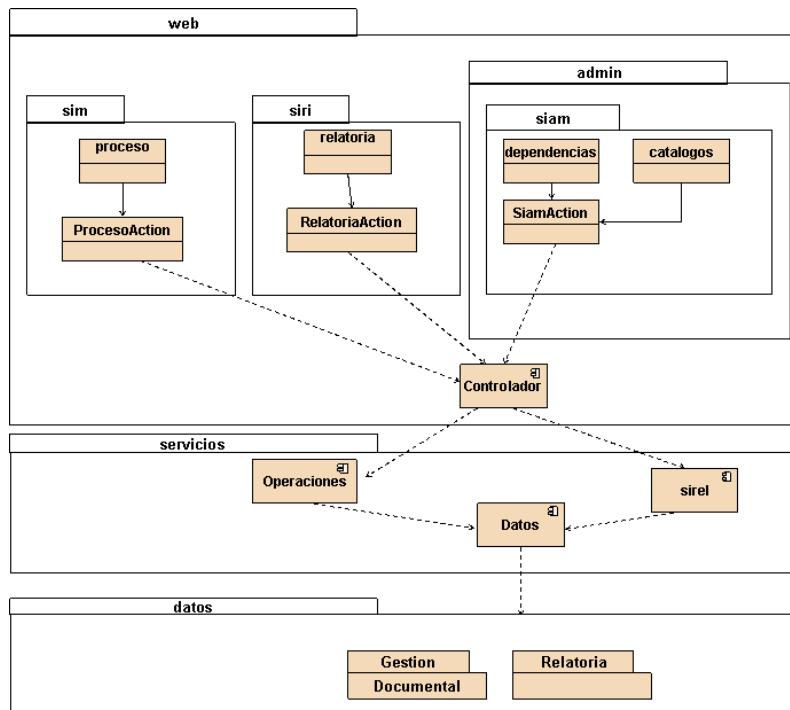
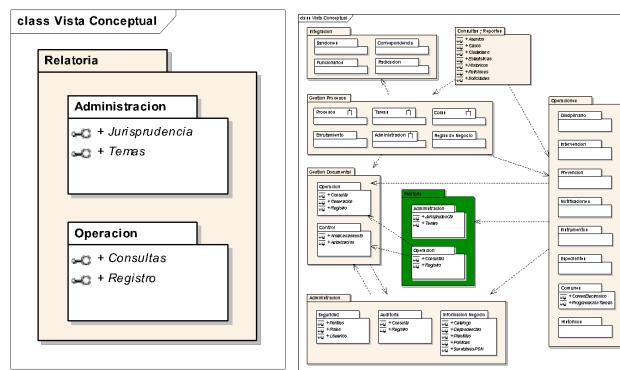


Figura 23. Relaciones del paquete de Gestión documental en la Vista Lógica

5.2.3 Relaciones del Paquete Relatoria de la vista Conceptual

Vista Conceptual paquete Relatoria



Paquetes relacionados en la vista Lógica

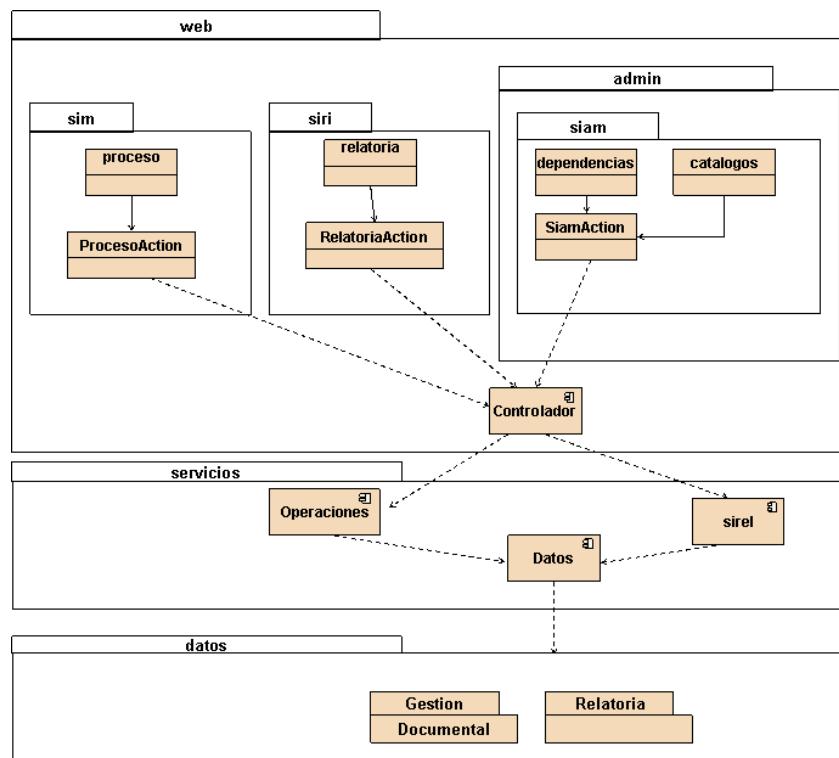
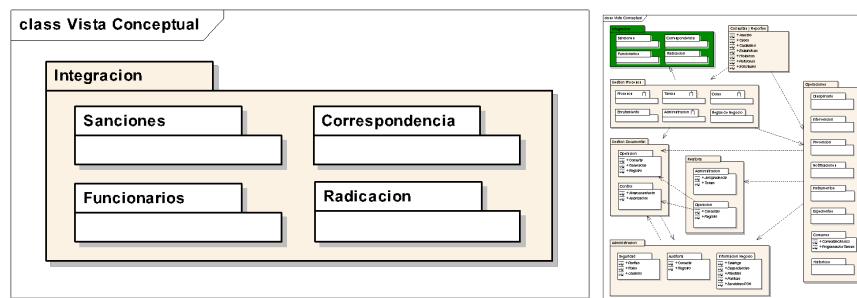


Figura 24. Paquetes relacionados en la vista Lógica del paquete relatoria

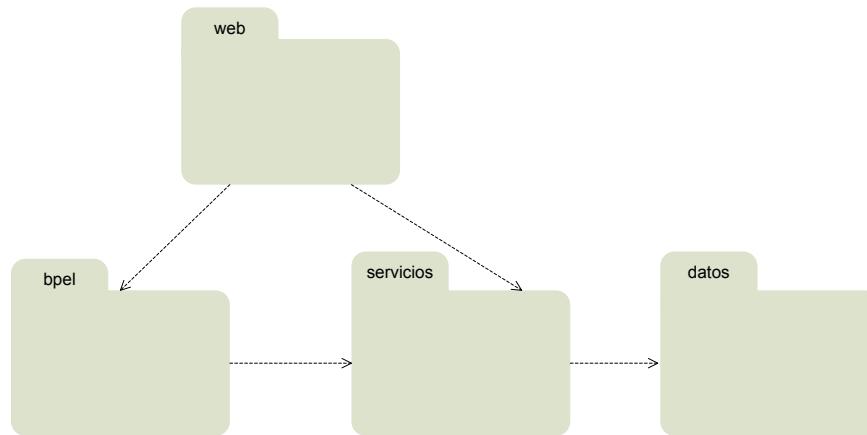
5.2.4 Relaciones del Paquete Integración de la vista Conceptual

El paquete Integración tiene relaciones con la gran mayoría de paquetes de presentados en la Vista de Lógica.

Vista Conceptual paquete Integración



Paquetes relacionados en la vista Lógica



5.2.5 Relaciones del Paquete Gestión de la vista Conceptual

Vista Conceptual paquete Gestión procesos

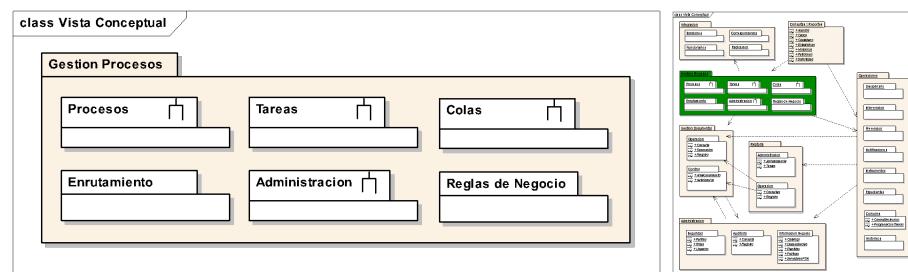


Figura 25. Vista Conceptual paquete Gestión procesos

Relaciones con la vista lógica con el paquete Gestión procesos-conceptual.

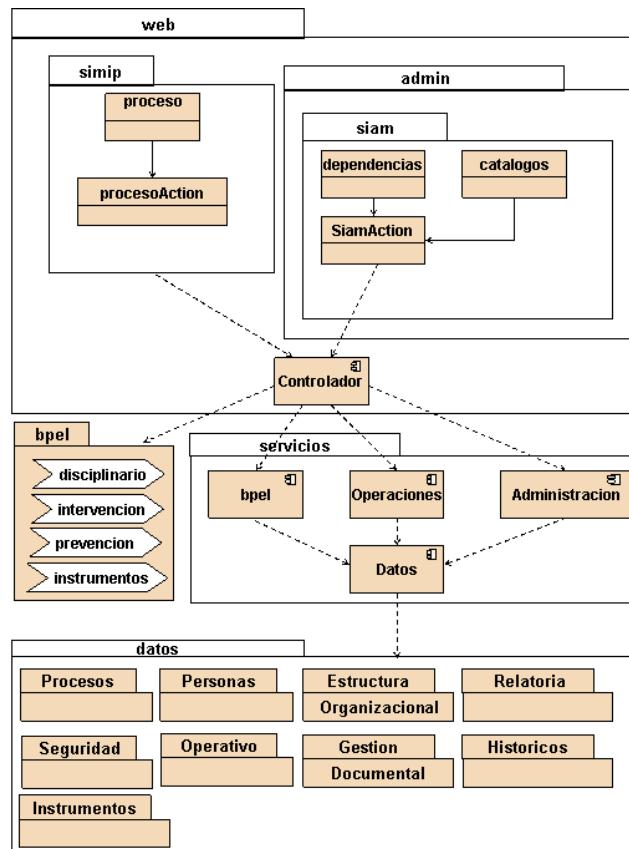


Figura 26. Relaciones con la vista lógica con el paquete Gestión procesos-conceptual

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

6. MODELO DE DOMINIO

6.1 Descripción

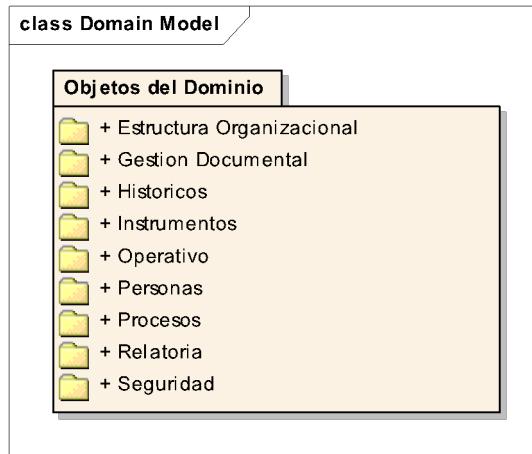


Figura 27. Paquete Modelo de Dominio

El Modelo de Dominio es un modelo conceptual de alto nivel, que contiene objetos físicos y abstractos, en un área de interés para el Proyecto.

Se puede usar para documentar relaciones entre ellos y responsabilidades de clases conceptuales, capturando objetos significantes en el sistema, organización u otro contexto.

La estructura de datos es la representación física del modelo de dominio. La estructura de datos es explicada al detalle en el documento de estructura de datos. Con cada iteración el modelo de dominio puede sufrir cambios menores que se reflejarán en otros documentos.

6.2 Elementos

El modelo de dominio de la Figura 27 tiene nueve grupos para la persistencia del SIM. Cada uno de los grupos es descrito en las siguientes secciones.

6.2.1 Procesos

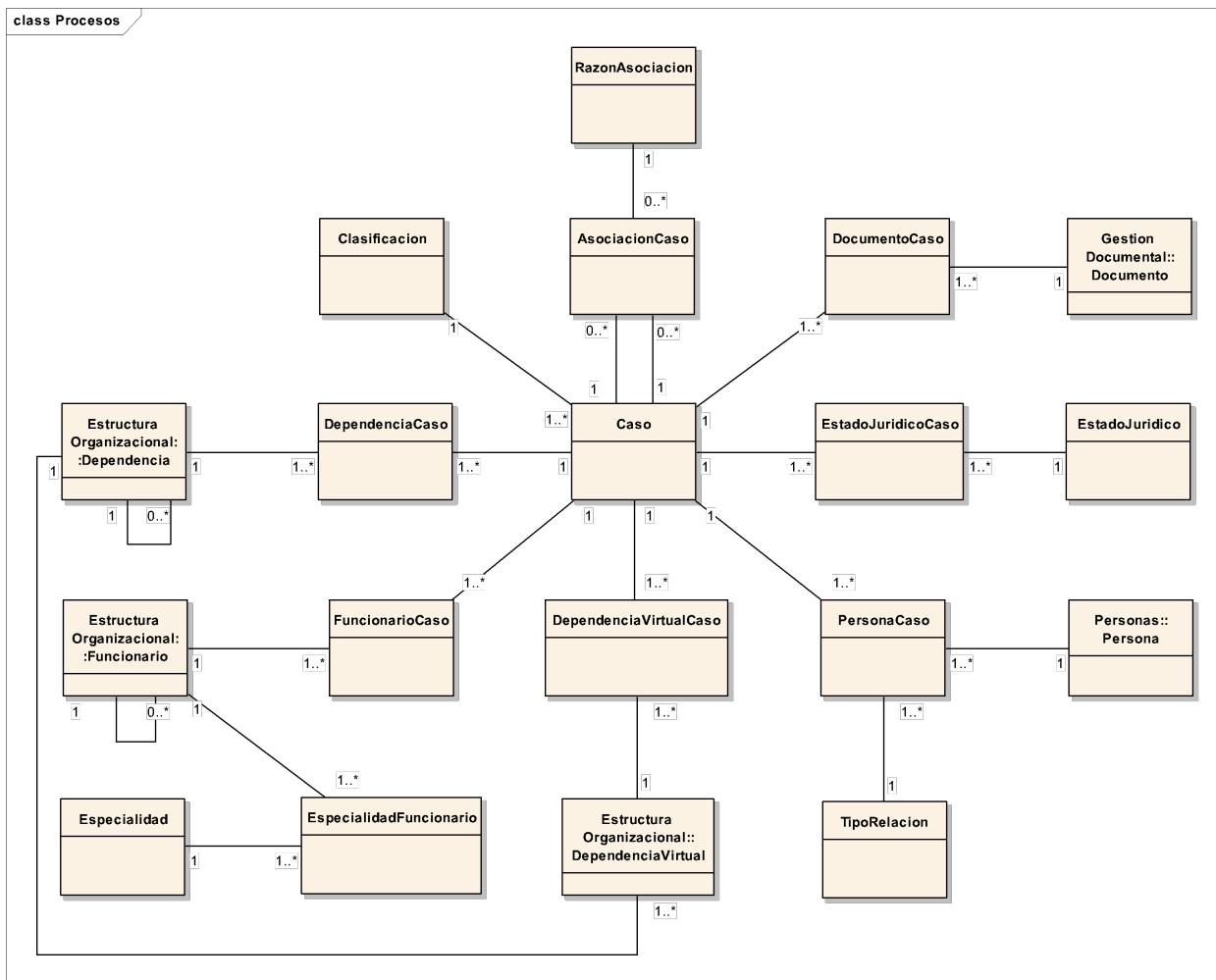


Figura 28. Modelo de dominio – Procesos

En Procesos se representa el almacenamiento de los datos de los casos dentro del SIM.

6.2.2 Personas

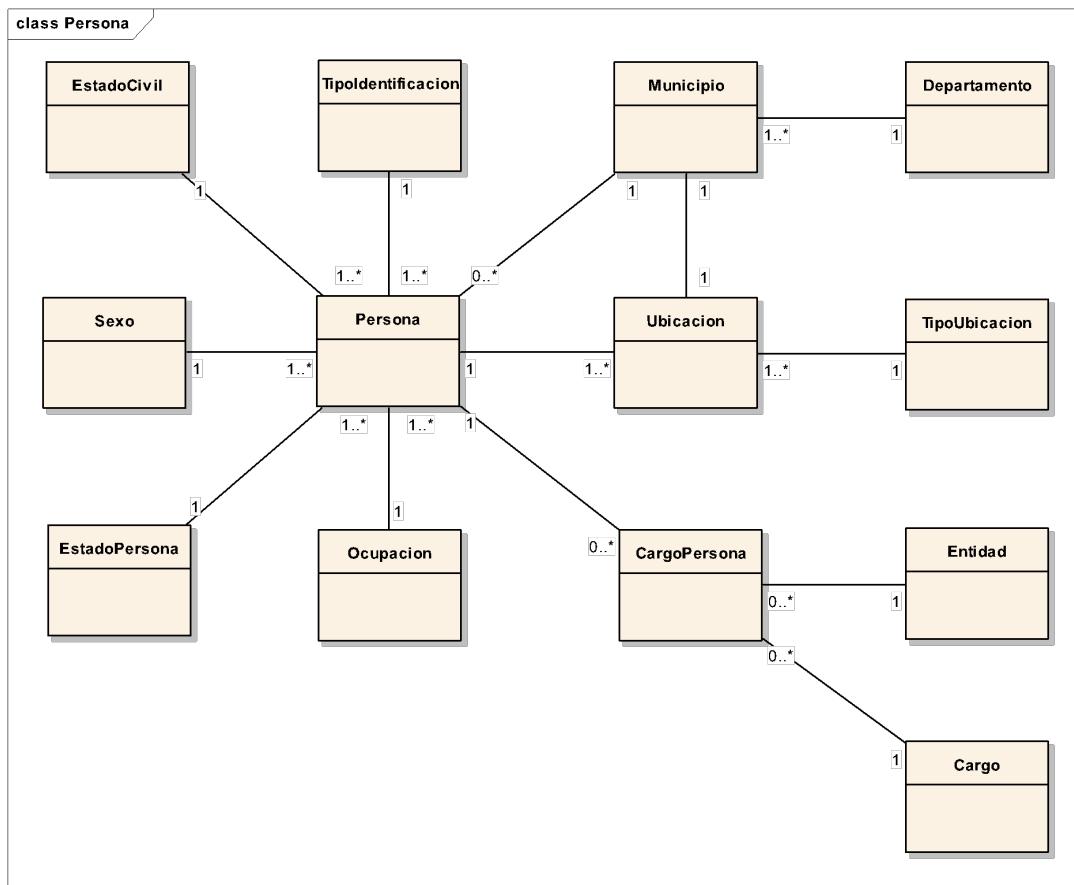


Figura 29. Modelo de Dominio - Personas

El elemento Personas se encarga de la persistencia de las personas relacionadas en las casos.

6.2.3 Estructura Organizacional

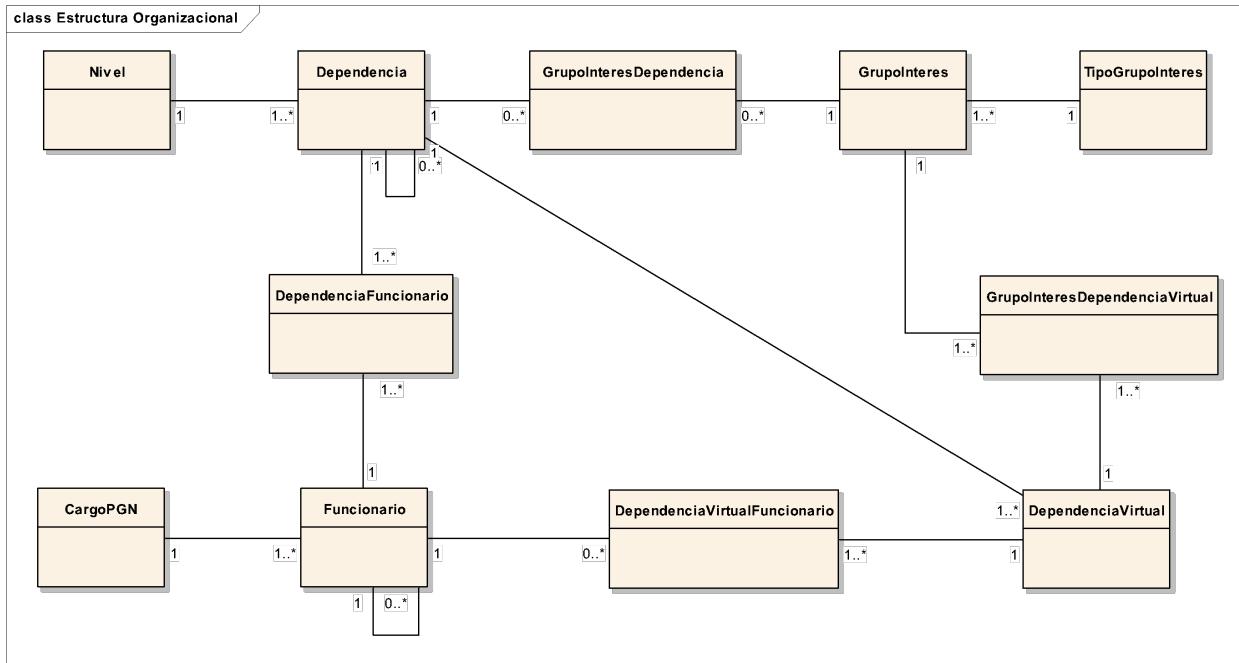


Figura 30. Modelo de dominio - Estructura Organizacional

El elemento Estructura Organizacional, representa de la información de la organización, principalmente para relacionar las competencias de funcionarios con los procesos.

6.2.4 Relatoría

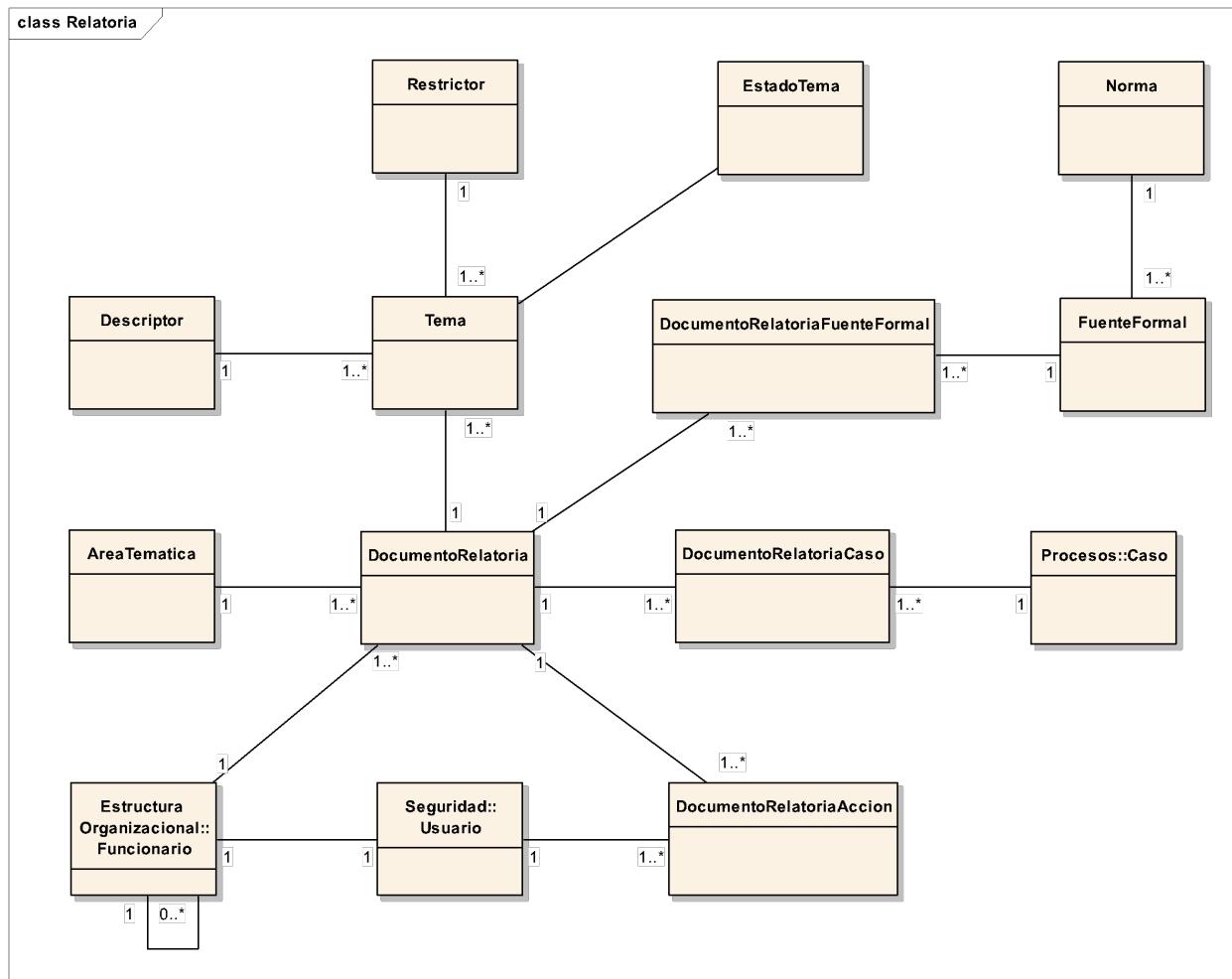


Figura 31. Modelo de Dominio – Relatoría

El elemento de Relatoría representa la persistencia de la información para la descripción (metadatos) y administración de los documentos de relatoría.

6.2.5 Instrumentos

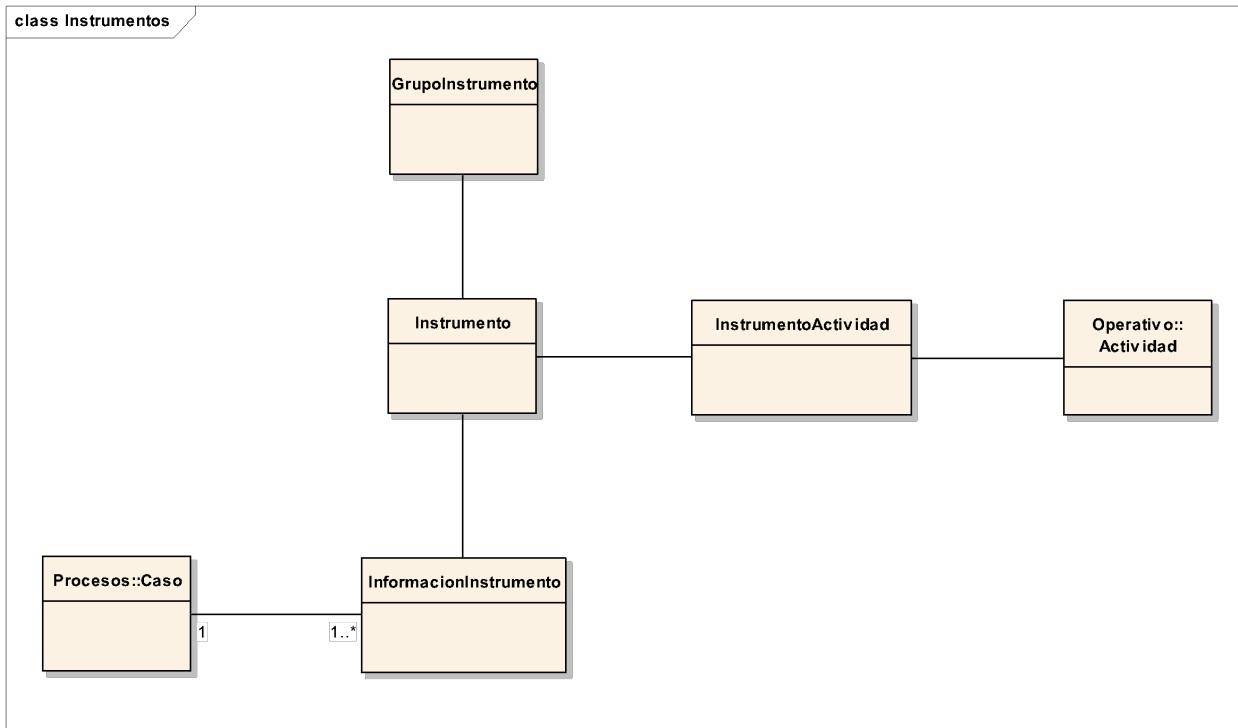


Figura 32. Modelo de dominio - Instrumentos

El elemento Instrumentos trata de la persistencia de los procesos de instrumentos en el SIM.

6.2.6 Operativo

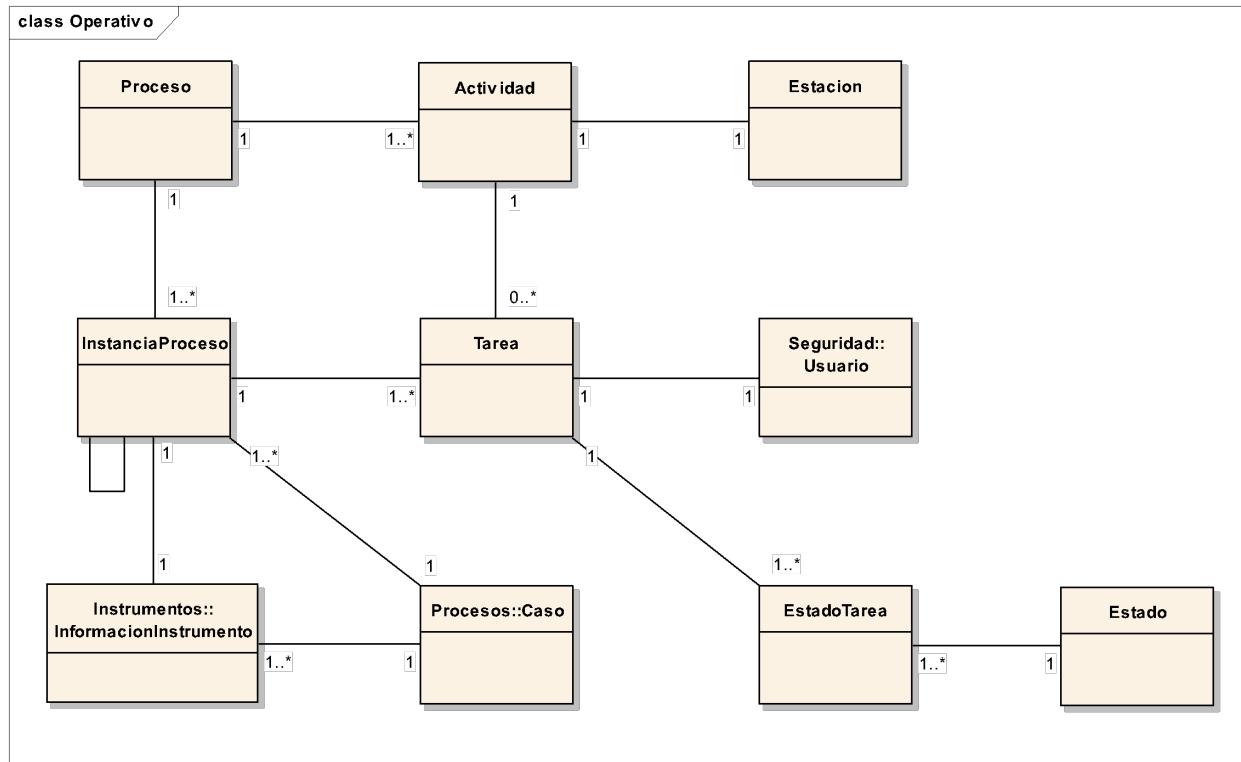


Figura 33. Modelo de Dominio - Operativo

El elemento Operativo representa la persistencia de la lista de tareas (de los procesos) de cada funcionario dentro de los procesos de la PGN.

6.2.7 Seguridad

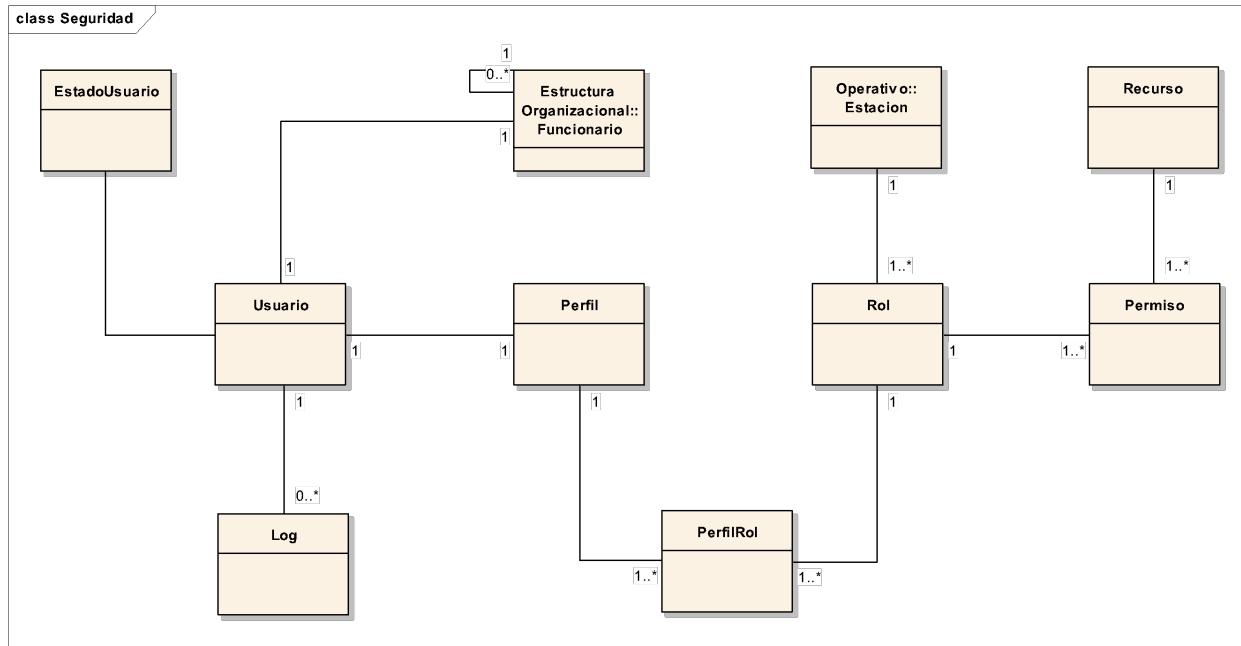


Figura 34. Modelo de dominio - Seguridad

El elemento de seguridad se representa de la persistencia y el modelo de los usuarios en lo que hace referencia a permisos de ejecución de recursos y competencias dentro del SIM.

6.2.8 Gestión Documental

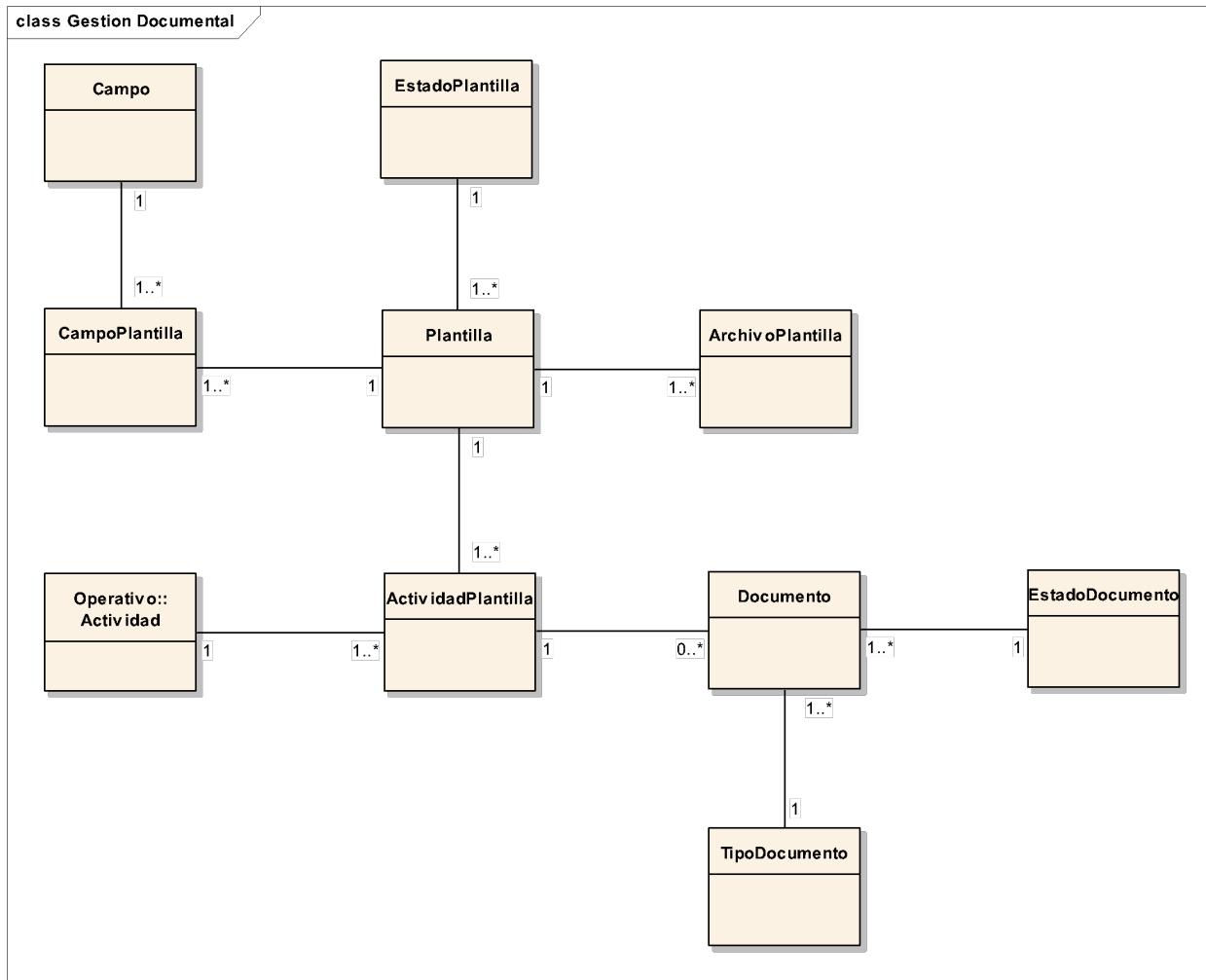
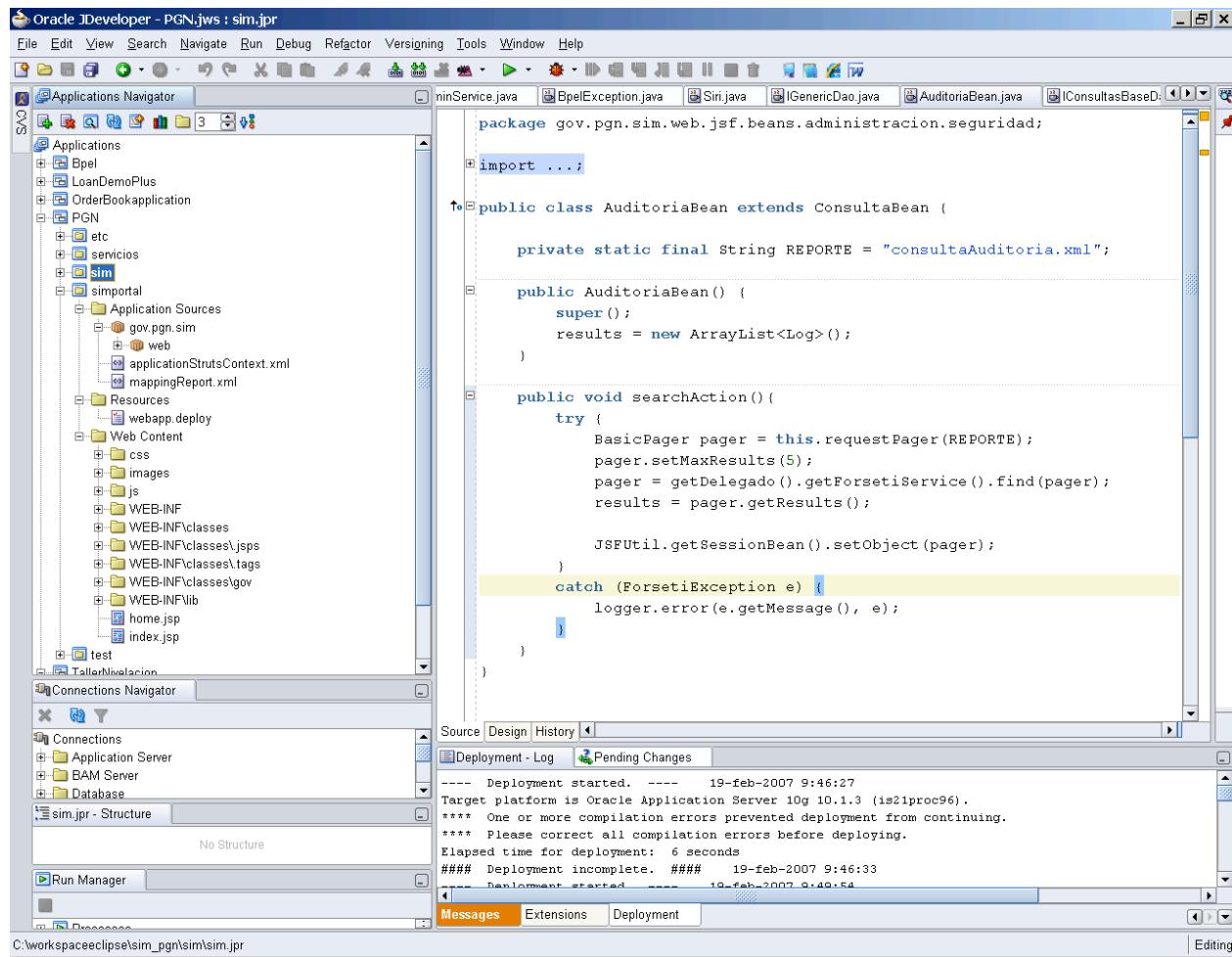


Figura 35. Modelo de domino - Gestión Documental

El elemento Gestión documental representa la persistencia de la información necesaria para generar documentos a partir de plantillas dentro del SIM.

7. AMBIENTE DE DESARROLLO

Como ambiente de desarrollo se utilizo la herramienta JDeveloper. A continuación se presenta una Imagen del ambiente de desarrollo en JDeveloper.

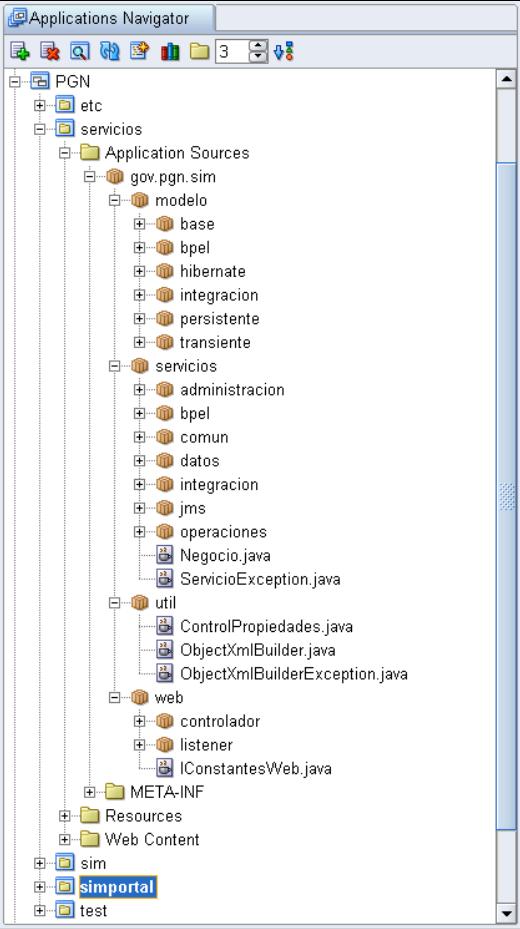


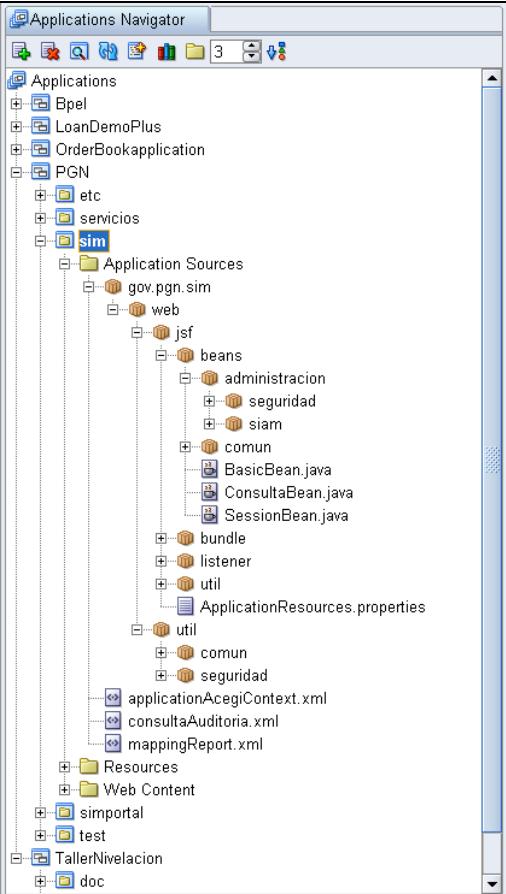
7.1 Aplicaciones dentro del ambiente de desarrollo

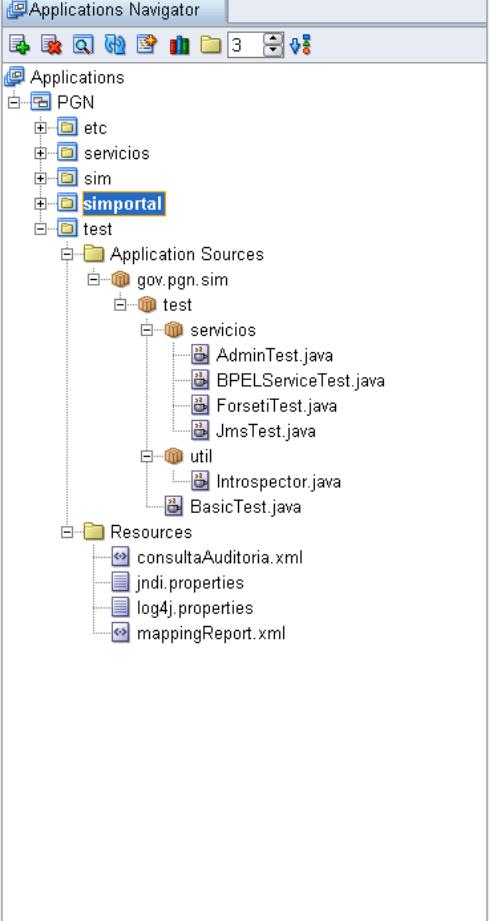
Dentro del ambiente de desarrollo el SIM tiene dos proyectos (los proyectos son PGN y BPEL). A su vez el proyecto PGN esta dividido en varias aplicaciones. Las aplicaciones del proyecto PGN son: etc, servicios, simportal y test.

A continuación se hace una breve descripción de cada aplicación dentro del proyecto PGN.

Aplicación servicios

Aplicación: servicios	Descripción
	<p>La aplicación servicios, contiene principalmente los componentes de la capa de negocio.</p> <p>La aplicación servicios tiene cuatro grandes paquetes:</p> <ul style="list-style-type: none"> - gov.pgn.sim.modelo: donde se encuentran las clases asociadas al modelo del mundo, incluyendo el persistente a partir del modelo de datos - gov.pgn.sim.servicios: donde se encuentran los componentes de los servicios de negocio - gov.pgn.sim.util: donde se encuentra clases de apoyo J2EE para diferentes necesidades. - gov.pgn.sim.web: donde se encuentran clases base de la capa de presentación.

Aplicación: sim	Descripción
	<p>La aplicación sim, contiene principalmente la capa de presentación y la invocación a los servicios.</p> <p>La aplicación sim tiene un paquete:</p> <ul style="list-style-type: none"> - gov.pgn.sim.web: donde se encuentran las clases que hacen parte del framework JSF.

Aplicación: test	Descripción
	<p>La aplicación test es utilizada para las pruebas unitarias.</p> <p>La aplicación test tiene un paquete:</p> <ul style="list-style-type: none"> - gov.pgn.sim.test

7.2 Lista principal frameworks utilizados

La lista principal de bibliotecas a usar inicialmente es la siguiente:

Elemento	Versión	Descripción/Uso
acegi-security-1.0.3.jar	1.0.3	Seguridad y permisos sobre recursos
ajax4jsf.jar		Librería del framework de Ajax para JSF
antlr-2.7.6.jar	2.7.6	Librería utilizada por hibernate para realizar parking de objetos
bcel-5.2.jar	5.2	Librería utilizada para manipular clases de java en bytecode
bsh-2.0b4.jar	2.0	Librería utilizada por hibernate para soporte de ejecución
castor-1.1.jar	1.1	Para mapeos entre Java y XML
cplib-nodep-2.1_3.jar	2.1_3	Para generación de código dinámicamente
commons-beanutils-1.7.0.jar	1.7.0	Biblioteca común incluida dentro de J2EE (para el uso y manejo de JavaBeans)
commons-codec-1.3.jar	1.3	Biblioteca común incluida dentro de J2EE (para la codificación de datos binarios)
commons-collections-3.1.jar	3.1	Biblioteca común incluida dentro de J2EE (colecciones)
commons-digester-1.6.jar	1.6	Biblioteca común incluida dentro de J2EE (procesamiento de acciones mediante XML)
commons-el-1.0.jar	1.0	Biblioteca común incluida dentro de J2EE (intérprete de JSP)
commons-fileupload-1.0.jar	1.0	Biblioteca común incluida dentro de J2EE (carga de archivos en aplicaciones con interfaz Web)
commons-io-1.1.jar	1.1	Biblioteca común incluida dentro de J2EE (operaciones E/S)
commons-lang-2.1.jar	2.1	Biblioteca común incluida dentro de J2EE (tipos de datos básicos)
commons-logging-1.0.4.jar	1.0.4	Biblioteca común incluida dentro de J2EE
commons-validator-1.2.0.jar	1.2.0	Biblioteca común incluida dentro de J2EE
dom4j-1.6.1.jar	1.6.1	Serialización de documentos XML.
facestrace-0.8.1.jar	0.8.1	Librería de apoyo para servicios de JSF
groovy-1.0-jsr-04.jar	1.0	Librería para apoyo de Java en lenguajes de scripting
hibernate3.jar	3.0	Persistencia de datos (Framework de hibernate)
jaxen-1.1-beta-7.jar	1.1	Para procesar sentencias XPath sobre documentos XML
log4j.jar		Registro de eventos

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---

Elemento	Versión	Descripción/Uso
myfaces-api-1.1.5-SNAPSHOT.jar	1.1.5	Contiene las interfaces de elementos comunes para la interfaz grafica con el usuario.
myfaces-impl-1.1.5-SNAPSHOT.jar	1.1.5	Contiene la implementación para de las interfaces de los elementos de la interfaz grafica con el usuario.
ojdbc14.jar	1.4	Conexión con la base de datos
orabpel-common.jar	10.1.3.1	Para la invocación de procesos BPEL
orabpel.jar	10.1.3.1	Para la invocación de procesos BPEL
oscache-2.3.2.jar	2.3.2	Para almacenamiento temporal (tipo <i>cache</i>) de páginas JSP
spring.jar	2.0.1	Librería del framework de Spring
tomahawk-1.1.5-SNAPSHOT.jar	1.1.5	Componente de interfaz grafica para JSF.
tomahawk-sandbox-1.1.5-SNAPSHOT.jar	1.1.5	Contiene componentes varios para la interfaz con el usuario.
xerces.jar		Manejo e interpretación de XML

7.3 *Interfaz de Usuario*

Esta sección introduce de una forma general la jerarquía y algunas funcionalidades la interfaz del usuario, pero no el diseño grafico final.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---



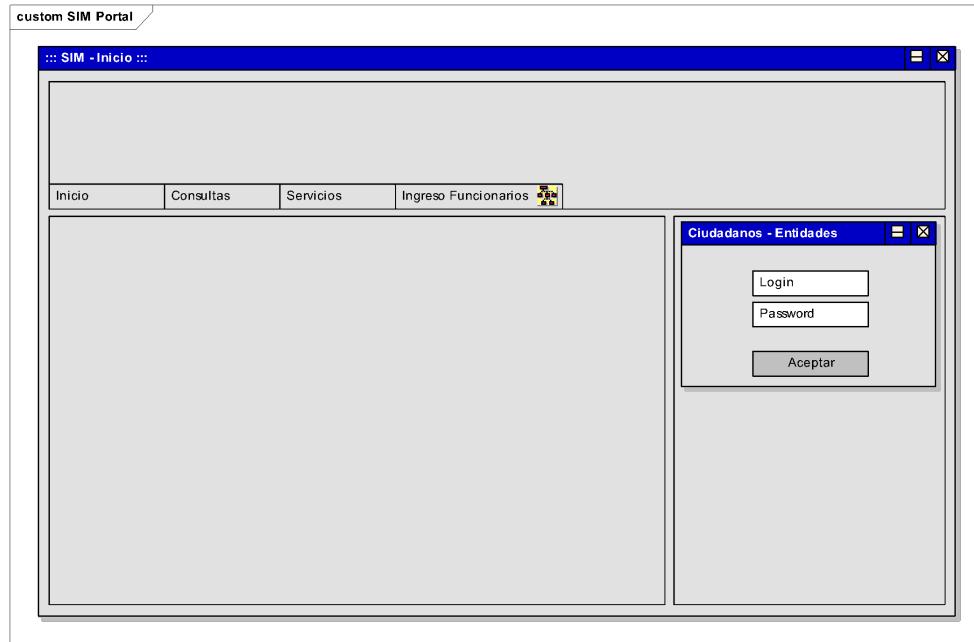
Figura 36. Interfaz de usuario

La interfaz de usuario se puede agrupar en cuatro grandes paquetes:

- Formas Portal: Interfaces de usuario de la página inicial del sistema
- Formas Primarias: Interfaces de usuario principales expuestas por el sistema
- Formas Administrativas: Interfaces de usuario de administración (creación, modificación, consulta).
- Ayuda en Línea: Interfaces de usuario para el manejo de la ayuda en línea.

7.3.1 Elementos

7.3.1.1 Formas Portal



7.3.1.2 Formas Primarias

Este paquete contiene las interfaces de usuario primarias del sistema. Estos son los puntos de interacción principal de los usuarios con el sistema.

Inicialmente el usuario ingresa por la forma presentada en la Figura 37. El texto que aparece en el botón puede cambiar y en algunos casos el texto sería configurable.

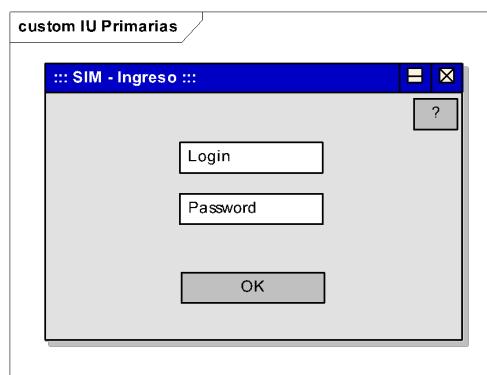
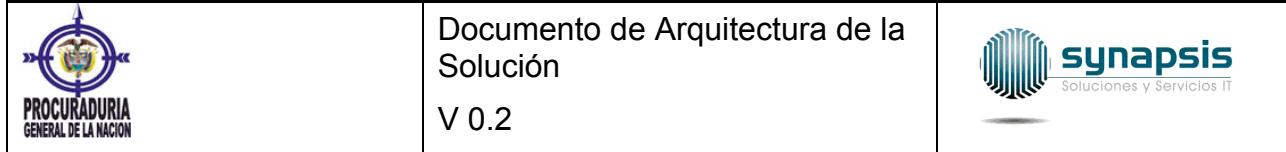


Figura 37. Forma de entrada al sistema

Luego de ingresar al sistema el usuario deberá aparecer la página de listado de tareas (ver la Figura 37).



The screenshot shows the 'custom IU Primarias' interface for the SIM - Principal application. At the top, there is a menu bar with options like Admin, SIMIP, SIREL, Consultas, Reportes, Alberto Perez, 01-11-2006, and Salida Segura. A tooltip message is displayed: 'A través del menú de opciones se accede a funcionalidades diversas dependiendo del módulo y la responsabilidad asignados. El menú es desplegable verticalmente.' Below the menu, there is a section titled 'Lista de Trabajo' containing a table with columns labeled 'Código'. The table has three rows, each corresponding to 'Reporte 1', 'Reporte 2', and 'Reporte 3'. At the bottom of the interface, there is a footer bar with the text 'Procuraduría General de la Nación (PGN) : Sistema de Información Misional (SIM) : Derechos Reservados (C) 2007'.

Figura 38. Forma de Lista de Tareas

Si selecciona un registro de la lista de trabajo que corresponde a una tarea de uno de los procesos de negocio determinado, se muestra una interfaz de usuario donde se procesa la correspondiente información del negocio (ver Figura 39).

custom IU Primarias

... SIM - Interna (Negocio) ...

La opción de Infomacion BPEL pretende encapsular las funcionalidades ofrecidas por el motor de procesos para la interacción con este.

Admin	SIMIP	SIREL	Consultas	Reportes	Alberto Perez	01-11-2006	Salida Segura
-------	-------	-------	-----------	----------	---------------	------------	---------------

Informacion Negocio	Informacion BPEL
---------------------	------------------

Datos Basicos	Datos Complementarios	Anexos	?
---------------	-----------------------	--------	---

IUC Fecha Ingreso Fecha Modificacion

Descripcion General

Clasificacion Dependencia

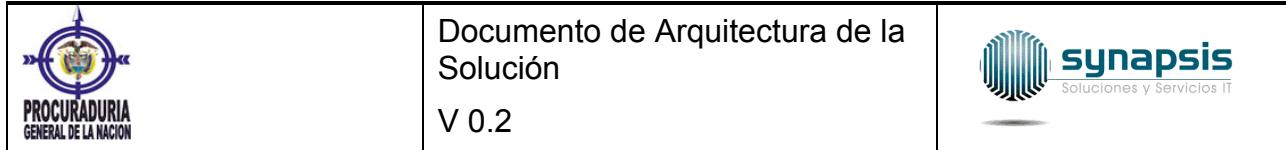
Equipo de Trabajo	Conclusion
-------------------	------------

Guardar Cancelar

Procuraduría General de la Nación (PGN) : Sistema de Información Misional (SIM) : Derechos Reservados (C) 2007

Figura 39. Forma específica de una tarea.

La Figura 39 representa una forma que aparece al seleccionar una tarea. La forma conteniente información específica para esa tarea. La forma de “tarea específica” tiene varias pestañas. Al seleccionar la pestaña de información BPEL aparecerá la forma de la Figura 40.



The screenshot shows a web-based application window titled 'custom IU Primarias'. At the top, there's a blue header bar with the text ':: SIM - Interna (BPEL) ::'. Below the header is a toolbar with buttons for Admin, SIMIP, SIREL, Consultas, Reportes, and a user profile for Alberto Perez (01-11-2006). A 'Salida Segura' (Secure Output) button is also present. The main content area has tabs for 'Informacion Negocio' and 'Informacion BPEL', with 'Informacion BPEL' currently selected. This tab contains several input fields: 'Tarea Número', 'Fecha Creación', 'Fecha Expiración', 'Estado', 'Titulo' (with a text input field), 'Acción' (with a dropdown menu), and 'Comentarios' (with a large text area). At the bottom of the form is an 'Aceptar' (Accept) button. A footer at the bottom of the page displays the copyright notice: 'Procuraduría General de la Nación (PGN) : Sistema de Información Misional (SIM) : Derechos Reservados (C) 2007'.

Figura 40. Forma de información BPEL.

La forma de la Figura 40 muestra el formulario asociado a la operación con el motor de procesos de negocio. En esta forma permite informar el avance del proceso una vez la tarea es completada.

7.3.1.3 Formas Administrativas

Las interfaces de usuario administrativas son usadas para el ingreso de datos de referencia, manejar seguridad, configuración y definición de preferencias.

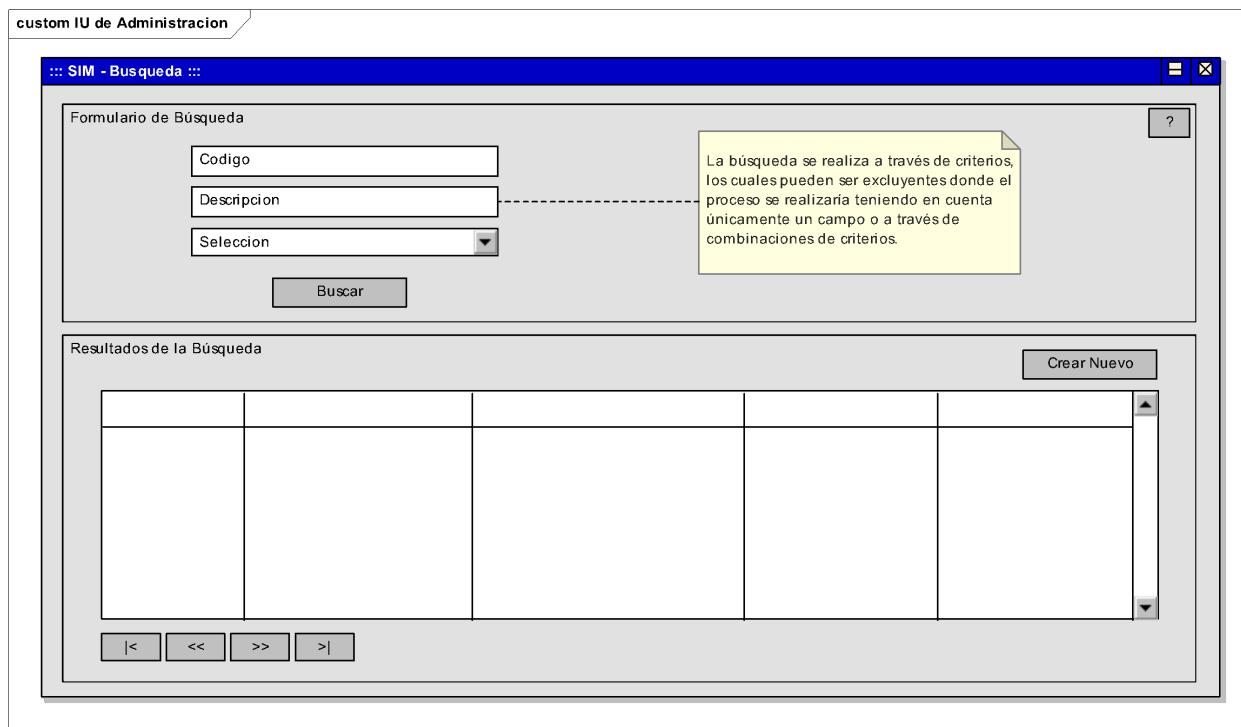


Figura 41. Forma genérica de búsqueda.

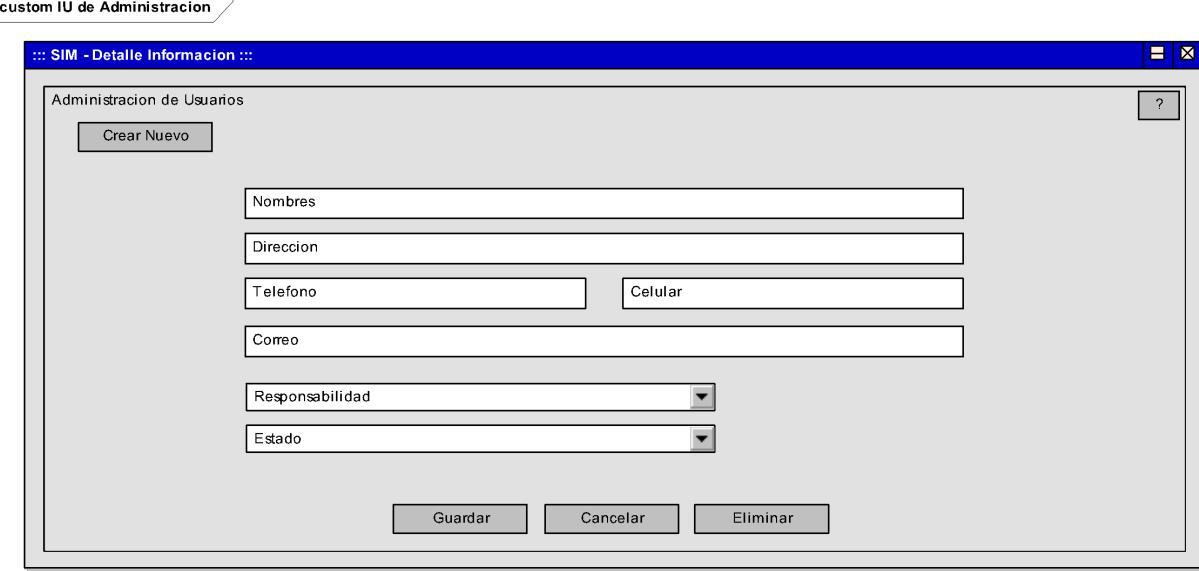
En la Figura 41 se muestra un ejemplo de forma administrativa, en este caso una forma de consulta-búsqueda de un usuarios.

Si se selecciona un elemento de la tabla de resultados se muestra una interfaz de usuario con el detalle de la información. Si se seleccionó la opción de Crear Nuevo, muestra la misma interfaz de usuario pero con los campos vacíos.

En las formas administrativas el encabezado y pie de página serían los mismos que se definieron en las formas primarias.

Un ejemplo típico de una forma de detalle se ve en la Figura 42.

 PROCURADURÍA GENERAL DE LA NACIÓN	Documento de Arquitectura de la Solución V 0.2	 synapsis Soluciones y Servicios IT
---	---	---



The screenshot shows a window titled "SIM - Detalle Información" with a blue header bar. Below the title, it says "Administración de Usuarios". There is a button labeled "Crear Nuevo" on the left. The main area contains several input fields: "Nombres", "Dirección", "Teléfono" (with a "Celular" field next to it), "Correo", "Responsabilidad" (with a dropdown arrow), and "Estado" (with a dropdown arrow). At the bottom, there are three buttons: "Guardar", "Cancelar", and "Eliminar". A question mark icon is in the top right corner of the window.

Figura 42. Ejemplo de detalle de forma administrativa

Una forma administrativa típica es presentada en la Figura 42, en donde aparecen los datos de un usuario o del detalle a modificar, también aparecen los botones de Guardar, Cancelar, Eliminar y crear.

7.3.1.4 Formas de Ayuda

El sistema de ayuda está modelado en un alto nivel que muestra las pantallas básicas de la ayuda en línea y los mecanismos de navegación. El diseño de la ayuda en línea presenta tres zonas principales: una sobre el lado izquierdo, donde se encuentra la tabla de contenido de la ayuda, la otra en el lado central sobre el que aparece el contenido del tema seleccionado, y una en la parte superior, donde aparece elementos de navegación y opciones de búsqueda. La ayuda en línea está orientada hacia el diseño de un books on line.

Las pantallas del sistema tendrían un elemento (por ejemplo el símbolo de interrogación -?- para llamar al sistema de ayuda en línea. Dependiendo de donde es llamado, se cargaría la ayuda asociada a la forma actual. Dentro del sistema de ayuda se podría revisar temas diferentes a través de una tabla de contenido (ver Figura 43).

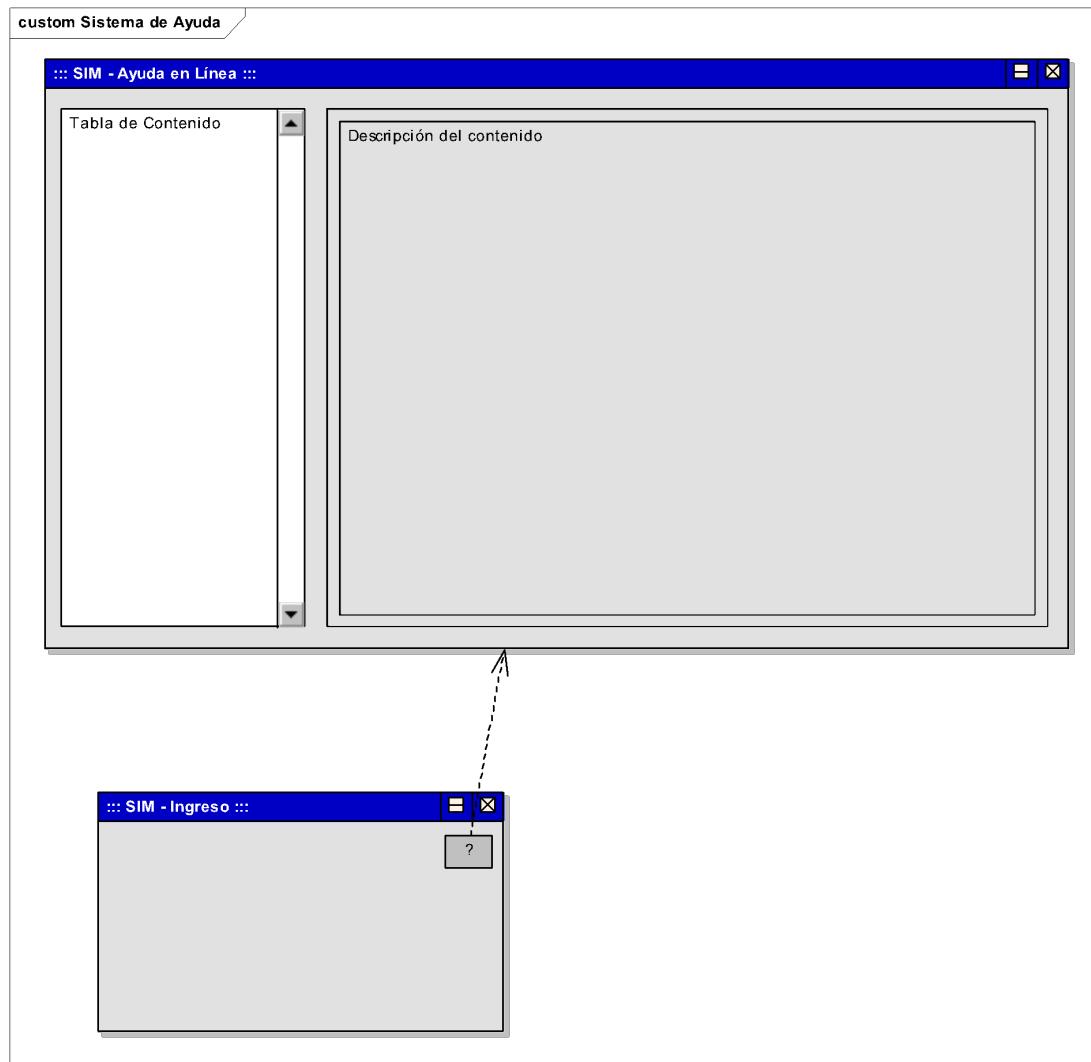


Figura 43. Modelo de Navegación de la Ayuda

8. MATERIAL DE REFERENCIA

Barbacci 2003	Barbacci, M.; Ellison, R.; Lattanze, A.; Stafford, J.; Weinstock, C.; & Wood, W. <i>Quality Attribute Workshops (QAWs)</i> , Third Edition (CMU/SEI-2003-TR-016). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. < http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html >.
Bass 2003	Bass, Clements, Kazman, <i>Software Architecture in Practice</i> , second edition, Addison Wesley Longman, 2003.
Clements 2001	Clements, Kazman, Klein, <i>Evaluating Software Architectures: Methods and Case Studies</i> , Addison Wesley Longman, 2001.
Clements 2003	Clements, Bachmann, Bass, Garlan, Ivers, Little, Nord, Stafford, <i>Documenting Software Architectures: Views and Beyond</i> , Addison Wesley Longman, 2003.
IEEE 1471	ANSI/IEEE-1471-2000, <i>IEEE Recommended Practice for Architectural Description of Software-Intensive Systems</i> , 21 September 2000.
SEI	SAD Template. < http://www.sei.cmu.edu >.
Adrián Lasso, MVP	Arquitectura de Software < http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/article110.asp >
LIC-PGN	Procuraduría General de la Nación, <i>Diseño, desarrollo e implantación del sistema de Información Misional (SIM) de la Procuraduría General de la Nación y adquisición del software de base necesario para el funcionamiento del sistema</i> , Marzo 2006
LICA4P2-PGN	Procuraduría General de la Nación, <i>Diseño, desarrollo e implantación del sistema de Información Misional (SIM) de la Procuraduría General de la Nación y adquisición del software de base necesario para el funcionamiento del sistema Anexo 4 Parte 2</i> , Marzo 2006

9. DIRECTORIO

9.1 Índice

Contenido: Esta sección provee un índice de los nombre de los elementos, relaciones, etc.., identificando para cada una la ubicación, donde está definido y donde se usa, dentro del documento.

9.2 Glosario

Término	Definición
Arquitectura de software	Estructura o estructuras de un sistema que comprimen elementos, sus propiedades visibles externamente y las relaciones entre ellos [Bass 2003]. “Visible externamente” se refiere a aquellas presunciones que otros elementos pueden hacer de un elemento, como servicios provistos, características de rendimiento, manejo de fallos, uso de recursos compartidos, etc.
Vista	Representación de todo un sistema desde la perspectiva de un grupo relacionado de preocupaciones [IEEE 1471]. Una representación de un tipo particular de elementos de arquitectura de software que ocurren en un sistema, sus propiedades y relaciones entre ellos.
Paquete de vista	La agrupación más pequeña de documentación de arquitectura que puede brindarse de manera útil a un stakeholder. La documentación de una vista se compone de uno o más paquetes de vista.
Punto de vista	Un punto de La especificación de las

	convenciones para construir y usar una vista; un patrón o plantilla de la cual desarrollar vistas individuales por medio del establecimiento de los propósitos y la audiencia de la vista, las técnicas para su creación y análisis [IEEE 1471]. Identifica el grupo de preocupaciones a ser atacadas, así como las técnicas de modelamiento, evaluación, verificación de consistencia, etc..., usado por cualquier vista.
--	--

9.3 Listado de Acrónimos

API	Application Programming Interface; Application Program Interface; Application Programmer Interface
IEEE	Institute of Electrical and Electronics Engineers
OS	Operating System
QAW	Quality Attribute Workshop
RUP	Rational Unified Process
SEI	Software Engineering Institute Systems Engineering & Integration Software End Item
UML	Unified Modeling Language
SOA	Service Oriented Architecture
MVC	Model View Controller
SIREL	Subsistema de Relación
SIM	Sistema de Información Misional
WSDL	Web Service Definition Language