



Universidad de las Ciencias Informáticas  
Facultad 3

# Propuesta de arquitectura de interoperabilidad para el intercambio de datos en el Órgano de Justicia-MININT

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**Autores:** Madelaine Hernández Matos, Onier Hogguit Rodríguez

**Tutor:**

Ing. Jorge Luis Valdés González

**Co-Tutora:**

MsC. Yarina Amoroso Fernández

Junio 2011

## **DECLARACIÓN DE AUTORÍA**

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

<Nombre y apellidos del o de los autores>

<Nombre y apellidos del o de los tutores>

---

**Firma de Autor**

---

**Firma de Tutor**

## **AGRADECIMIENTOS**

**Made:** A mi madre por ser la mejor madre del mundo, por ser mi amiga mi consejera mi ejemplo a seguir.

A mi padre por su ejemplo y apoyarme siempre.

A mi abuelita lola por darme su amor desde pequeña.

A mi novio Carlos.

A mi tío pele que siempre lo llevo presente.

A mi hermana mayor.

A mis hermanos pequeños (los mellizos) que los amo mucho y estos años no lo he visto crecer como quisiera.

A mi amiga y hermana Aleanna por estar siempre conmigo en las buenas y malas por apoyarme siempre.

A Evelio y Isabel por ser mis padres, por apoyarme en todos momentos.

A mis suegros por ser los mejores del mundo.

A mi profesor guía Oneikis por guiarnos el camino a seguir.

A mis amigos en general por estar siempre conmigo y confiar siempre en mí.

A mi tutor Jorge Luis por su paciencia y apoyo.

**Onier:** A mi madre por apoyarme siempre y depositar su confianza en mí, por ser la mejor del mundo.

A mi padrastro por aconsejarme siempre y apoyarme en todo.

A mis abuelos Chicho y Rosa por haberme dado todo desde pequeño, por confiar en mí, por ser los mejores.

A mis hermanos por confiar en mí en todo momento.

A mis tíos Nini y Chichi por ayudarme siempre y darme la mano cuando me ha hecho falta.

A la flaca Made.

A todos mis amigos por estar conmigo en todas las situaciones, en las buenas y en las malas.

A los profesores que me han formado, a Wilfredo en especial y a Eydi Marien.

A mi tutor Jorge Luis por haber tenido paciencia y habernos ayudado en todo.

## DEDICATORIA

**Made:** A mi mama, mi papa, mi abuelita lola, mis hermanos, mis tíos, profes, amigos, a mis suegros y mi novio que son los mejores, mis otros padres, Evelio Isabel...

**Onier:** A mi mama, mi padrastro, mis abuelos, mis hermanos, a mis tíos, profesores, amigos. A todos los que de una u otra forma me han apoyado siempre y han confiado en mí...

*<Dedicatoria de la tesis>.*



*“Los ordenadores son increíblemente rápidos, exactos y estúpidos.  
Los seres humanos son increíblemente lentos, inexactos y brillantes.  
Juntos son poderosos más allá de la imaginación.”*

*Albert Einstein*

## **RESUMEN**

Con el fin de lograr que los sistemas y aplicaciones informáticas que intervienen en el proceso de Justicia Penal en Cuba, puedan interactuar en el intercambio de datos, a fin de lograr más eficiencia y eficacia en el proceso, surge la necesidad de desarrollar una plataforma informática que permita la interacción entre dichos sistemas informáticos.

Fue necesario diseñar una arquitectura de interoperabilidad que sirva como base para el posterior desarrollo de la plataforma informática, respondiendo a atributos como la seguridad, flexibilidad y escalabilidad entre otras. Entre los aspectos que influyeron para alcanzar estos atributos se encuentra el estudio de algunas de las soluciones de interoperabilidad existentes a nivel internacional, la definición y puesta en práctica de un diagnóstico organizacional que permitió evaluar el estado tecnológico y de infraestructura de los distintos organismos que forman parte del proceso. Gracias a esta evaluación fue posible la definición de los estándares y herramientas que forman parte de la arquitectura de interoperabilidad y posteriormente la evaluación de la misma a través de métodos que permitieron evaluar los atributos de calidad deseados en los escenarios definidos.

Como parte del proceso de desarrollo de la Plataforma de Interoperabilidad del Órgano de Justicia-MININT se realizó el diseño de la arquitectura. Luego del análisis de la misma y los resultados obtenidos se comprobó que la arquitectura cumple con los requerimientos propuestos y satisface los atributos que se desean alcanzar de forma general permitiendo que exista la interoperabilidad en el intercambio de datos en el mencionado órgano.

**Palabras Claves:** Arquitectura de interoperabilidad, estándares, atributos de calidad.

# ÍNDICE

<b>DECLARACIÓN DE AUTORÍA.....</b>	<b>I</b>
<b>AGRADECIMIENTOS .....</b>	<b>I</b>
<b>DEDICATORIA.....</b>	<b>II</b>
<b>RESUMEN.....</b>	<b>IV</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>7</b>
<b>INTRODUCCIÓN.....</b>	<b>7</b>
1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA .....	7
1.1.1 Interoperabilidad.....	7
1.1.1.1 Niveles de Interoperabilidad.....	8
1.1.1.2 Beneficios de la Interoperabilidad .....	9
1.2 DISEÑO DEL DIAGNÓSTICO TECNOLÓGICO A LOS ORGANISMOS FGR, TSP Y MININT. FUNDAMENTACIÓN DE LA PROBLEMÁTICA .....	9
1.3 ARQUITECTURA DE INTEROPERABILIDAD .....	12
1.3.1 Pasos para definir la arquitectura de interoperabilidad .....	12
1.4 SOLUCIONES DE INTEROPERABILIDAD EN EL ÁMBITO INTERNACIONAL .....	13
1.4.1 Proyecto ISA.....	13
1.4.2 EJIS (Esquema Judicial de Interoperabilidad y Seguridad) .....	14
1.4.3 Proyecto ATLANTE .....	15
1.4.4 Proyecto SIMAC (Sistema Integral Multicanal de Atención al Ciudadano).....	16
1.4.5 Proyecto de Interoperabilidad en Chile (Minsegpres). .....	16
1.5 HERRAMIENTAS Y ESTÁNDARES:.....	17
1.5.1 ¿Qué es un estándar?.....	17
1.5.1.1 Estándares abiertos como pilares de interoperabilidad .....	18
1.5.2 SGML.....	19
1.5.3 XML (Extensible Markup Language) como lenguaje de intercambio de datos .....	19
1.5.4 XML-Schema para la estandarización de datos y documentos XML.....	20
1.5.5 WSDL para la descripción de los contratos o interfaces de los servicios .....	21
1.5.6 Protocolo REST (Representational State Transfer) .....	21
1.5.7 Protocolo SOAP (Simple Object Access Protocol).....	22
1.5.8 Comparación entre SOAP y REST.....	22
1.5.9 UDDI (Universal Description, Discovery and Integration) .....	23
1.5.10 Servidores de aplicación .....	24
1.5.10.1 JBOSS.....	24
1.5.10.2 EAServer .....	24
1.5.10.3 GlassFish .....	25
1.5.10.4 Jonas.....	25
1.5.11 Herramientas y componentes de seguridad.....	26
1.5.11.1 Acaxia.....	26
1.5.11.2 Kerberos.....	26
1.5.11.3 SAP NetWeaver .....	27
1.5.12 Algunos estándares o protocolos de seguridad .....	27
1.5.12.1 Web Services Security .....	28
1.5.12.2 XML-Encryption.....	28

1.5.12.3	XML Digital Signature.....	28
1.5.12.4	Security Assertion Markup Language (SAML).....	29
<b>CONCLUSIONES DEL CAPÍTULO:</b> .....		<b>31</b>
<b>CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>		<b>33</b>
<b>INTRODUCCIÓN.....</b>		<b>33</b>
2.1	REQUERIMIENTOS DE LA PLATAFORMA DE INTEROPERABILIDAD PARA EL SISTEMA DE JUSTICIA PENAL EN CUBA...33	
2.2	TIPOLOGÍA ESTRUCTURAL Y TIPOS DE DATOS. ....	34
2.3	DATOS A LOS QUE RESPONDEN LOS SERVICIOS WEB.....	35
2.4	PROCEDIMIENTO DE ESTANDARIZACIÓN DEL FORMATO DE INTERCAMBIO DE DATOS .....	35
2.5	INTERFACES DE LOS SERVICIOS WEB .....	36
2.6	COMPONENTES DE LA ARQUITECTURA DE LA PLATAFORMA DE INTEROPERABILIDAD DEL ÓRGANO DE JUSTICIA EN CUBA 41	
2.6.1	Portal Corporativo.....	41
2.6.2	Repositorio de Servicios Web basado en el estándar UDDI.....	42
2.6.2.1	Sistema de Administración del repositorio de servicios web .....	43
2.6.3	Repositorio de esquemas y metadatos .....	44
2.6.4	Requerimientos de los sistemas de administración de los repositorios de registros web y de esquemas y metadatos. ....	44
2.6.5	Servidor de Aplicaciones .....	45
2.6.5.1	Configuración del servidor.....	45
2.6.5.2	Persistencia del servidor en producción.....	46
2.7	SEGURIDAD EN LA PLATAFORMA.....	47
2.7.1	Elementos de seguridad del intercambio de datos y el acceso a los sistemas de la plataforma .....	47
2.7.1.1	Estándares que aportan seguridad a la plataforma .....	47
2.7.1.2	Acaxia como componente de seguridad centralizada .....	48
2.7.2	Elementos de seguridad del entorno de despliegue de los componentes de la plataforma.....	49
2.7.2.1	Firewall como forma protección de los servidores.....	49
2.8	VISTA DE DESPLIEGUE DE LOS COMPONENTES DE LA ARQUITECTURA DE INTEROPERABILIDAD .....	49
<b>CONCLUSIONES DEL CAPÍTULO:</b> .....		<b>50</b>
<b>CAPÍTULO 3: ANÁLISIS Y VALIDACIÓN DE LA ARQUITECTURA.....</b>		<b>51</b>
<b>INTRODUCCIÓN.....</b>		<b>51</b>
3.1	OBJETIVOS.....	51
3.2	NECESIDAD DE EVALUAR UNA ARQUITECTURA .....	51
3.3	RESULTADOS QUE SE OBTIENEN A TRAVÉS DE LA EVALUACIÓN DE LA ARQUITECTURA .....	52
3.4	¿CUÁNDO EVALUAR LA ARQUITECTURA?.....	52
3.5	ALGUNOS MÉTODOS QUE PERMITEN EVALUAR ARQUITECTURA.....	52
3.6	DEFINICIÓN DE LOS ATRIBUTOS DE CALIDAD A EVALUAR. ....	54
3.7	EVALUACIÓN DE LA ARQUITECTURA DE INTEROPERABILIDAD.....	54
<b>CONCLUSIONES DEL CAPÍTULO:</b> .....		<b>58</b>
<b>CONCLUSIONES GENERALES .....</b>		<b>59</b>
<b>RECOMENDACIONES .....</b>		<b>60</b>
<b>BIBLIOGRAFÍA.....</b>		<b>61</b>

## **INTRODUCCIÓN**

En la actualidad las tecnologías del desarrollo y las comunicaciones (TIC) se han convertido en una herramienta muy poderosa e indispensable para la sociedad y todas las esferas de un país. Desde hace ya algunos años, Cuba ha emprendido el reto de informatizar la sociedad para lograr una mayor calidad de vida del pueblo, más eficiencia en la economía, así como satisfacer las necesidades de información y conocimiento de las personas y sectores de la sociedad.

En el Proceso de Justicia Penal intervienen diversos Órganos y Organismos del Estado cubano tales como la Fiscalía General de la República (FGR), Tribunal Supremo Popular (TSP) y el Ministerio del Interior (MININT) que junto al Ministerio de Justicia, conforman el Órgano de Justicia-MININT. Este Órgano entre otras funciones y objetivos contribuye al proceso de toma de decisiones del Sistema Judicial en Cuba que a la vez, contribuye a la agilidad, eficiencia y eficacia del Proceso de Justicia Penal.

El proceso de toma de decisiones se nutre de los datos que son obtenidos a partir de la realización del proceso de Justicia Penal, destacando que estos no son generados por solo un organismo, sino que forman parte del resultado de las funciones que realizan los sistemas informáticos que intervienen en la realización del proceso y que se encuentran ubicados en las distintas entidades a las que pertenecen.

Los sistemas o plataformas informáticos con que cuenta hoy el Sistema Judicial en Cuba, que son los encargados de automatizar gran parte de los procesos internos que se desarrollan en cada uno de los organismos, son las principales herramientas a través de las cuales, se maneja la información que circula entre dichas entidades.

Algunas de las plataformas o sistemas informáticos son el Sistema de Informatización de la Gestión de las Fiscalías (SIGF), el Sistema Automatizado Jurídico Operativo (SAJO) y el Sistema de Informatización del Tribunal (SIT). En algunos casos como SAJO, son sistemas legados que llevan más de 30 años al servicio de la Justicia. A pesar de la relación que existe entre los procesos que realizan dichos sistemas, las plataformas informáticas se encuentran aisladas por la incapacidad de interacción de los sistemas internos con los externos.

Sin embargo, se han llevado a cabo algunas ideas para solucionar este problema, aunque todavía existen cuestiones como el tiempo, la agilidad y la eficiencia que no son los óptimos debido a que aún son afectadas e influyen directamente en el éxito de la toma de decisiones judiciales.

Por ejemplo, en el MININT existen sistemas que son capaces de interactuar e intercambiar información internamente mediante el compartimiento centralizado de datos, es decir, el acceso común por parte de los sistemas informáticos internos a las bases de datos, recalando que a pesar de que de cierta forma se resuelve este problema, no es la forma correcta y óptima, pues evidentemente se exponen los datos a problemas de seguridad desde dentro y fuera del propio organismo y todavía no son capaces de interactuar con sistemas externos a la organización. Así mismo, el SIGF es capaz de generar datos y reportes a nivel nacional como es el caso del PMI (Parte Mensual Informativo). Sin embargo, este sistema solamente tributa datos para lo interno de la FGR y aún no es capaz de interactuar con otros sistemas informáticos ya sean internos o externos.

Por otro lado, se han intentado resolver cuestiones como la redundancia de datos que existe actualmente debido a que las soluciones informáticas existentes no son capaces de colaborar y los datos que se generan en otras organizaciones u organismos, son obtenidos a través de diferentes e inseguras vías como son el correo, teléfono, dispositivos de almacenamiento o documentos impresos en papel. Como solución actual a este problema, luego de la obtención de estos datos provenientes de otras fuentes, son ingresadas manualmente en las aplicaciones informáticas. Esto trae consigo problemas como la validez de los datos, pues debido a la forma en que son manipulados los mismos, existe la posibilidad de que sean modificados durante su envío, traslado o recepción. También se evidencia que existe lentitud en el intercambio de datos que se realiza actualmente y una inconsistencia y dispersión de los mismos. Además, debido a que la información es tratada manualmente desde un primer momento, la misma se encuentra expuesta a problemas de deterioro y el volumen de información va creciendo exponencialmente, haciéndose más difícil encontrar datos específicos.

Igualmente, existe un consumo de recursos innecesario que influye en el presupuesto de cada uno de los organismos y se observa que los datos son tratados aisladamente y no como

una cadena o secuencia sincronizada con el estado del propio proceso, influyendo negativamente en la calidad del mismo.

La incapacidad de que los sistemas informáticos puedan interactuar en el intercambio de datos e información, es algo que se encuentra fuertemente atado al desarrollo tecnológico en general, pues desde el propio surgimiento de los sistemas informáticos y las tecnologías de la información es una cuestión que ha afectado duramente a todos y todo lo que de una u otra forma esté vinculado directamente con estas tecnologías.

El sistema Judicial en Cuba no está exento de este problema, pues es a causa de que los sistemas informáticos que aportan datos al Órgano de Justicia-MININT no interactúan en el intercambio de datos, que es afectado directamente el proceso de toma de decisiones evidenciando que existe un problema de interoperabilidad que requiere una solución urgente.

La solución para que los sistemas informáticos puedan interactuar entre sí e intercambiar información (datos, documentos, etc.), es un factor clave para la implementación de cualquier infraestructura de Gobierno electrónico o lo que es lo mismo eGobierno, según la Comisión Económica para América Latina y el Caribe (CEPAL). La misma plantea varias acciones a llevar cabo que sirven como punto de partida para lograr la interoperabilidad.

Centrando la atención en el Proceso de Justicia Penal, se decide por parte de la Comisión Nacional del Órgano de Justicia-MININT, la definición de un equipo de trabajo que pueda desarrollar guiados por los lineamientos que propone la CEPAL, una plataforma informática que permita la interacción entre los sistemas informáticos que tributan al desarrollo del proceso de Justicia Penal en Cuba.

A partir de lo antes expuesto, se ha decidido por parte del equipo de trabajo y aprobado por la Comisión Nacional del Órgano Justicia-MININT, en acuerdo con la Universidad de las Ciencias Informáticas, un análisis que ha permitido de forma general observar las condiciones en que se encuentran los órganos y organismos que intervienen en el proceso de Justicia Penal, centrándose en aspectos como la disponibilidad tecnológica y el grado de informatización de los procesos internos de dichos organismos.

Se necesita realizar un marco de interoperabilidad para lograr un mayor desarrollo, cooperación, y prestación de servicios por parte de cada uno de los sistemas informáticos y organismos que intervienen en el proceso de toma de decisiones, impactando positivamente

en la eficiencia, eficacia, la técnica y la economía del proceso de Justicia Penal en Cuba, haciendo énfasis en aspectos como la seguridad, integridad y autenticidad así como la consistencia de los datos. Con la propuesta de la arquitectura de interoperabilidad será posible minimizar todos o la mayoría de los problemas existentes actualmente en el intercambio de datos que se lleva a cabo durante el desarrollo del proceso en general.

Por dichas razones se define como **problema de investigación** El proceso de intercambio de datos entre los organismos de la FGR, el TSP y el MININT es ineficiente e impide que el Órgano de Justicia-MININT de Cuba tome las decisiones eficazmente.

Partiendo del problema anterior se define como **objeto de estudio** la interoperabilidad entre sistemas informáticos y el **campo de acción** está delimitado por las arquitecturas de interoperabilidad entre sistemas informáticos judiciales.

El **objetivo general** de la presente investigación es realizar una propuesta de arquitectura de interoperabilidad para el intercambio de datos en el Órgano de Justicia-MININT, a partir de la cual se pueda implementar una plataforma informática que contribuya a la eficiencia en el proceso de toma de decisiones en el mencionado órgano.

Partiendo de lo anteriormente expuesto, se plantea como idea a defender que la definición de una propuesta de arquitectura de interoperabilidad para el intercambio de datos entre los sistemas informáticos de los organismos de la FGR, el TSP y el MININT, permitirá implementar una plataforma informática que contribuya a la eficiencia del proceso de toma de decisiones en el Órgano de Justicia-MININT.

Teniendo como objetivos específicos los siguientes:

1. Elaborar el marco teórico sobre Arquitecturas de Interoperabilidad entre sistemas informáticos.
2. Describir las tecnologías en la que se implementan y despliegan los sistemas informáticos que intervienen en el proceso de Justicia Penal.
3. Definir una Arquitectura de Interoperabilidad que permita la colaboración entre los sistemas informáticos que intervienen en el proceso de Justicia Penal.
4. Validar la propuesta realizada.

Para lograr el cumplimiento de los objetivos trazados es necesario llevar a cabo las siguientes tareas de la investigación:

1. Realización del diseño teórico de la investigación.
2. Elaborar el Marco Conceptual sobre lo que es Interoperabilidad, haciendo énfasis en la Interoperabilidad en un entorno Jurídico.
3. Estudio del estado del arte sobre soluciones de Interoperabilidad en el mundo y Cuba.
4. Diagnóstico tecnológico a los Órganos de Justicia que intervienen en el proceso de Justicia Penal en Cuba.
5. Proponer arquitectura interoperabilidad a partir del diagnóstico realizado.
6. Definición de los instrumentos y métodos de validación de la propuesta.
7. Validar la propuesta a través de métodos y técnicas que permitan evaluar los atributos de calidad en una arquitectura.

Entre los métodos de investigación científica existentes se utilizaron los siguientes:

#### **Métodos Teóricos:**

- Histórico-Lógico: se utiliza este método debido a la necesidad de realizar un estudio acerca de las soluciones de interoperabilidad existentes en el mundo en la actualidad enfocándose en los estándares y tecnologías que permiten desarrollar una arquitectura de interoperabilidad.

#### **Método Empírico:**

- Entrevista: se utiliza para obtener información detallada acerca del grado informatización de los procesos internos de los organismos así como el estado tecnológico de los mismos utilizando como instrumento el cuestionario.

Estructura del Trabajo de Diploma:

#### **Capítulo 1: Fundamentación de la problemática.**

En este capítulo se realiza un estudio acerca de algunas de las soluciones que existen en el mundo, así como se analizan los principales conceptos y definiciones asociados al dominio del problema que sirven de apoyo durante el desarrollo de la investigación, centrándose en

los estándares y tecnologías abiertas que permiten el desarrollo de una arquitectura de interoperabilidad aclarándose el significado o concepto de la misma.

**Capítulo 2:** Descripción de la solución propuesta.

Este capítulo se centra en la definición de las interfaces o contratos de los servicios web que brindarán los datos de cada uno de los organismos pertenecientes al Órgano de Justicia-MININT. Se definen y describen los estándares y tecnologías utilizadas para la propuesta de solución y la interacción entre sus componentes. Además, se definen los principales requerimientos de la plataforma de interoperabilidad y se realiza la propuesta de solución de la arquitectura de interoperabilidad.

**Capítulo 3:** Análisis y validación de la Arquitectura propuesta.

En este capítulo, como aspecto fundamental, se analizan varios métodos que permiten evaluar los atributos de calidad en la arquitectura. Se definen los atributos y escenarios principales a evaluar y que se desean alcanzar en la plataforma, demostrándose a partir de la relación de estos atributos con los escenarios definidos. Por último y no menos importante, se obtienen los resultados de la evaluación realizada a la arquitectura.

## Capítulo 1: FUNDAMENTACIÓN TEÓRICA

### INTRODUCCIÓN

En el presente capítulo se expone una visión general de los conceptos más significativos relacionados con la interoperabilidad para el estudio y realización de una propuesta de arquitectura tal que los sistemas informáticos del Órgano de Justicia Penal en Cuba sean capaces de interactuar entre ellos, logrando así una mayor eficiencia durante todo el proceso. Se describen los distintos niveles con que cuenta la Interoperabilidad y se realiza un análisis general acerca de las tecnologías, estándares y herramientas utilizadas en la actualidad para el desarrollo de una arquitectura que permita la interoperabilidad entre sistemas heterogéneos.

#### 1.1 Conceptos asociados al dominio del problema

##### 1.1.1 Interoperabilidad

La interoperabilidad tiene hoy varios conceptos y definiciones aplicados desde diferentes puntos de vista por cada uno de los diferentes autores que hacen alusión al tema, pero todos y cada uno de ellos literalmente llegan a la misma conclusión.

La Comisión Europea es una de las organizaciones que brinda apoyo a la CEPAL, en la versión 2 del IEF (*European Interoperability Framework for Pan-European eGovernment Services v2.0*) (IDABC, 2008) plantea que la interoperabilidad es “*la capacidad de las organizaciones y sistemas diversos y dispares de trabajar juntos eficazmente en el cumplimiento de los objetivos comunes en beneficio mutuo mediante el intercambio de datos entre los sistemas de las TIC*”.

Por otro lado, se define la interoperabilidad como “*la habilidad de compartir e intercambiar información usando sintaxis y semánticas comunes para cumplir una relación funcional de una aplicación específica a través del uso de una interfaz común*” según el Boletín Trimestral del Instituto Tecnológico de Informática dado en el año (2006).

Teniendo en cuenta las anteriores definiciones, para el marco de la presente investigación se define la interoperabilidad como la posibilidad de que sistemas informáticos, con características, funciones y tecnologías distintas; y por ende heterogéneos; pertenecientes a

organizaciones distintas, puedan interactuar en el intercambio de datos con el fin de lograr uno o varios objetivos comunes.

#### **1.1.1.1 Niveles de Interoperabilidad**

Según la bibliografía consultada, se pudo observar que existen distintos niveles de interoperabilidad, algunos de estos autores como (Manso, y otros, 2007) señalan que se han identificado al menos 15 niveles; sintáctico; semántico; técnico; pragmático; organizacional; conceptual; dinámico; legal; social; estructural o esquemático; intracomunitario; político-humano; internacional; empírico y físico.

Teniendo en cuenta que la palabra nivel<sup>1</sup> está asociada a cuestiones de jerarquía o secuencia de escalones, se decide aclarar que los niveles antes mencionados no serán tratados según el significado de la palabra, sino como tipos de interoperabilidad, además, solo se tendrán en cuenta la interoperabilidad técnica, la semántica y la organizacional.

- **Interoperabilidad Técnica:**

Posibilita la interconexión de los sistemas a nivel de protocolo y el intercambio de información a su nivel más básico, bits.

- **Interoperabilidad Semántica:**

Es la que posibilita el intercambio de información, utilizando un vocabulario común y compartido que evite las inexactitudes, en la interpretación del significado de los términos.

- **Interoperabilidad Organizacional:**

Permite conocer los objetivos de negocio, los modelos de procesos, las leyes y políticas de acceso y el uso de los datos y servicios.

La interoperabilidad no solo se limita a cuestiones de tecnología, pues muchos suelen hacer alusión a otras dimensiones un poco más allá de lo funcional definiendo la Interoperabilidad Horizontal, Interoperabilidad Vertical, Interoperabilidad Intra-administrativa y la Regional o Transfronteriza según (Criado, et al., 2010) en el documento “Bases para una Estrategia Iberoamericana de Interoperabilidad” dado entre el 1ero y 2 de julio del año 2010 en Buenos Aires-Argentina.

---

<sup>1</sup>Referencia a una escala determinada según la Real Academia Española con sitio en <http://www.rae.es>

### **1.1.1.2 Beneficios de la Interoperabilidad**

La interoperabilidad entre los sistemas informáticos puede multiplicar los rendimientos que se obtienen con la automatización de los procesos a los que responden dichos sistemas, pues permite combinar diferentes tecnologías para lograr la cooperación a los distintos niveles con otras empresas, se obtiene además una simplificación de la actividad administrativa y de los procesos de negocio, dando lugar a una mayor eficiencia. Por otro lado, posibilita el uso de estándares abiertos que en la mayoría de las ocasiones reduce el costo en la adopción de tecnologías así como también permite la reutilización de datos, información y funcionalidades mejorando así los procesos de tomas de decisiones puesto que los datos son más confiables y poseen más calidad.

También es posible a través de la interoperabilidad la promoción de la cooperación internacional en ámbitos en los que se requieran nuevas tecnologías, se gana además en aspectos como la sencillez en el intercambio de información influyendo positivamente en factores como el tiempo y el presupuesto. Visto incluso desde otro punto de vista, es posible salvar vidas humanas ante la ocurrencia de desastres si todos los sistemas de la policía, salud, bomberos y demás organismos inmersos en estas situaciones son capaces de interactuar entre ellos, ganando en organización y a la vez disminuyendo el tiempo de respuesta ante el desastre.

En resumen, la interoperabilidad aporta beneficios para todas las esferas de un país y alcanza su máximo nivel cuando es aplicado al nivel de gobierno, permitiendo que se logre interactuar con otros países a través de los sistemas informáticos.

## **1.2 Diseño del diagnóstico tecnológico a los organismos FGR, TSP y MININT.**

### **Fundamentación de la problemática**

Con el objetivo de obtener información acerca de la disponibilidad tecnológica así como de los procesos internos de cada uno de los órganos y organismos que intervienen y aportan datos al proceso de Justicia Penal se realiza un cuestionario [Anexo 1](#) a dichos organismos.

Los objetivos específicos del diagnóstico son:

1. Determinar el estado actual de la organización en cuanto a seis dimensiones fundamentales: negocio, sistemas, datos, infraestructura, recursos humanos e integración.

2. Realizar un análisis del estado actual e identificar las principales debilidades y problemas que presenta la organización en las seis dimensiones.
3. Determinar un estado deseado de las seis dimensiones que garantice y potencie el cumplimiento de la misión y los objetivos estratégicos de la organización.
4. Identificar, evaluar y priorizar iniciativas de solución a los problemas identificados en cada dimensión que permitan la transición al estado deseado.
5. Identificar y evaluar los posibles proyectos, así como realizar una priorización teniendo en cuenta criterios como la urgencia, la prioridad de los procesos de negocio involucrados, el tiempo, el costo, el retorno de la inversión, y el aporte al resto de los proyectos.
6. Determinar una hoja de ruta de proyectos que establezca el camino a seguir para lograr la transición del estado actual al estado deseado de forma gradual y organizada en estas seis dimensiones, alineando las iniciativas tecnológicas al cumplimiento de la misión y los objetivos estratégicos de la organización.

La aplicación de este diagnóstico le permitió a la gerencia del grupo de trabajo definir una proyección estratégica para llevar a cabo las principales tareas encaminadas a desarrollar la plataforma de interoperabilidad. En este sentido la contribución fundamental para el caso del diseño de la arquitectura de interoperabilidad, consistió en identificar el estado actual de las tecnologías de cada uno de los organismos involucrados en el proceso de intercambio de datos.

El diagnóstico abarcó seis dimensiones:

- **Dimensión de negocio:** la que permitió, en sentido general: identificar las necesidades, requerimientos y puntos de contacto, de cada organismo que interviene en el proceso de Justicia Penal.
- **Dimensión de Sistemas:** Caracterizar los sistemas informáticos que intervienen en el proceso de Justicia Penal e identificar y caracterizar la interacción de estos sistemas informáticos con otros sistemas que contribuyen a la realización del proceso; además de identificar potenciales desarrollos necesarios para la solución de interoperabilidad. Esta dimensión fue la que más información aportó para el diseño de la plataforma propuesta en este trabajo.

- **Dimensión de datos:** Identificar y caracterizar las fuentes y/o orígenes de datos además de clasificar los datos/documentos/información que se intercambian entre los organismos para la realización del proceso.
- **Dimensión de infraestructura:** Caracterizar la infraestructura tecnológica en cada organismo en cuanto a conectividad, seguridad y equipamiento; al mismo tiempo que permitió identificar requerimientos potenciales de equipamiento para el despliegue de la solución de interoperabilidad.
- **Dimensión de recursos humanos:** la cual tiene como objetivo identificar y clasificar de acuerdo a su responsabilidad, a las personas que participan en la realización del proceso.
- **Dimensión de Integración:** la cual se encarga de identificar la relación y/o alineación entre las dimensiones anteriores.

El análisis de los resultados del diagnóstico, permitió concluir que el proceso de intercambio de datos que se lleva a cabo actualmente entre los organismos que intervienen en el proceso de toma de decisiones, se realiza con lentitud, perdiendo eficiencia y eficacia. Se evidenció que los sistemas informáticos que intervienen en el proceso son incompatibles y heterogéneos, además, se pudo observar que algunas de las variantes que existen en algunos organismos como forma de intercambio de datos a través de las aplicaciones informáticas no son las óptimas. El caso de compartir los datos centralizadamente para que accedan organismos y aplicaciones informáticas que no forman parte del propio organismo que los comparte, no es una solución factible, pues se exponen los datos a problemas de seguridad debido a que los mismos pueden ser accedidos por aplicaciones y usuarios que no pertenecen a la intranet o al organismo que le concierne.

Por otro lado, se obtuvo que debido a que el intercambio de datos se realiza de forma física, se arrojan dificultades con el volumen de documentos impresos, pues va aumentando considerablemente y conlleva a problemas como la búsqueda de algún dato específico. También, se evidenció que los datos están expuestos a la corrupción debido a que son tratados manualmente, por lo que además existe la posibilidad de que los datos no sean confiables e íntegros.

A partir de los resultados obtenidos a través de la aplicación del diagnóstico, se definen los principales pasos a seguir para el diseño y posterior implementación de la arquitectura de la plataforma de interoperabilidad.

### **1.3 Arquitectura de interoperabilidad**

La mayoría de las personas que no poseen conocimientos acerca de lo que es la interoperabilidad y las arquitecturas de software, tienden a pensar que una arquitectura de interoperabilidad es lo mismo que una arquitectura de software.

En el libro blanco de interoperabilidad (Moreno Escobar, y otros, 2007) que es parte del resultado de la investigación de la CEPAL, se plantea que una arquitectura de interoperabilidad ayuda a definir la forma en que las aplicaciones serán construidas, cómo se desarrollarán los componentes y servicios, y sobre todo, cómo podrán interactuar con los demás sistemas con los que se cuenta. Una arquitectura provee la vista global de cómo será construida tecnológicamente, teniendo en cuenta aspectos como la flexibilidad y escalabilidad. Al final, las aplicaciones deben interactuar transparentemente entre sí usando estándares y formas de comunicación previamente definidas para todas en conjunto.

En otras palabras, una arquitectura de interoperabilidad no es más que la proposición de estándares, formas, métodos de comunicación, componentes y tecnologías, que permitan que sistemas informáticos logren interactuar entre sí para permitir el flujo de datos, teniendo en cuenta elementos o aspectos como la seguridad, autenticidad e integridad de los datos que son objeto de esta interacción.

#### **1.3.1 Pasos para definir la arquitectura de interoperabilidad**

El libro blanco de interoperabilidad de Gobierno Electrónico para América Latina y el Caribe (CEPAL, 2007), plantea un itinerario a seguir para lograr la interoperabilidad entre regiones o países diferentes. Dicho itinerario, inicia con un consenso sobre las definiciones y la conceptualización de la arquitectura de interoperabilidad, a partir del cual se programaron las actividades prioritarias que aseguren contar con una plataforma de interoperabilidad regional.

Adaptando algunas de estas actividades que propone la CEPAL a las necesidades y condiciones que apremian al Órgano de Justicia-MININT, se consideran como pasos básicos diseñar una arquitectura de interoperabilidad los siguientes:

- 1) Evaluación del estado tecnológico e infraestructura en general, de las entidades involucradas directamente en el proceso de interoperabilidad.
- 2) Análisis de soluciones existentes que aporten elementos para fundamentar las decisiones que incidan directamente en la arquitectura.
- 3) Definición del lenguaje de intercambio.
- 4) Definición de la vía a través de la cual se compartirán los datos entre las entidades involucradas en el proceso.
- 5) Definir los componentes que formarán parte de la arquitectura basándose en estándares abiertos.
- 6) Descripción de la interacción de todos los componentes de la arquitectura.

## **1.4 Soluciones de interoperabilidad en el ámbito internacional**

### **1.4.1 Proyecto ISA**

Según el documento Los servicios públicos europeos de Administración electrónica del año (2010), el programa ISA persigue apoyar la cooperación entre administraciones públicas, siempre al servicio de actos y políticas comunitarias, sobre la base de un uso eficaz y eficiente de las tecnologías de la información para el establecimiento de servicios públicos transfronterizos, evitando barreras a las interacciones transfronterizas por motivo de falta de interoperabilidad y de soluciones comunes. Para desarrollar sus actividades ISA se apoya en los principios de neutralidad con respecto a la tecnología y adaptación, apertura, reutilización, privacidad y protección de los datos personales y la seguridad. El programa ISA no parte de cero, es construido sobre las bases de sus antecesores, IDABC, IDA II e IDA, que en su momento obtuvieron muy buenos resultados.

IDABC constituyó la base legal del programa para el periodo de 2005-2009. Entre sus principales objetivos estuvo la identificación, promoción y desarrollo de los denominados servicios paneuropeos de administración electrónica, dirigidos a los ciudadanos, a las empresas y a las administraciones, así como las infraestructuras y servicios necesarios para el despliegue de los primeros. IDA II, constituyó para el periodo 1999-2004 la base legal para el establecimiento de los servicios entre administraciones, denominados Proyectos de Interés Común, para el soporte a la aplicación de políticas y actos comunitarios, el apoyo a la comunicación interinstitucional en la Unión Europea y el apoyo al proceso de decisión

comunitario; así como el establecimiento de las denominadas Acciones y Medidas Horizontales, para interoperabilidad de infraestructuras, servicios y contenidos en apoyo del despliegue de los citados proyectos.

Anterior a IDA II se encuentra la Decisión IDA. Se lanza en 1995, dentro de sus objetivos se encuentra abordar el establecimiento efectivo de la interoperabilidad de los sistemas de información de los Estados miembros, la contribución comunitaria de proyectos de ámbito sectorial, aprovechando en la mayor medida posible los sistemas preexistentes, y a la vez, promoviendo una decidida política de estándares para evitar la creación de barreras innecesarias y conseguir condiciones de economía a escala.

Hay que tener en que ISA está orientado a la comunidad pública y no a los sistemas judiciales, además, las condiciones, objetivos y regulaciones bajo las que fue construido no son las que se persiguen en Cuba. También, se evidencia un alto coste de consumo en cuanto al valor monetario por lo que es de difícil acceso para Cuba. En sentido general este proyecto no se desenvuelve en un marco jurídico aunque hay que destacar que sí logra la interoperabilidad.

#### **1.4.2 EJIS (Esquema Judicial de Interoperabilidad y Seguridad)**

EJIS no es más que la colaboración entre el Ministerio de Justicia, el Consejo General del Poder Judicial, la Fiscalía General del Estado y las Comunidades Autónomas para abogar por la incorporación de las tecnologías al ámbito de la prestación de servicios de la justicia en España. Las bases fundamentales de este proyecto son la interoperabilidad y la seguridad. Dentro de sus objetivos se encuentra facilitar el máximo aprovechamiento de las Tecnologías de la Información y las Comunicaciones en la actividad administrativa, así como establecer el marco semántico, técnico y organizativo a través de las plataformas tecnológicas necesarias para el funcionamiento integrado y conjunto de todas las aplicaciones informáticas al servicio de la administración de justicia.

EJIS dispone de un lenguaje común de comunicación entre los servicios de información pertenecientes a la justicia. También identifica las infraestructuras de comunicaciones y de interoperabilidad comunes y descentralizadas disponibles, así como establece un marco de gobernanza que permite prestar servicios definidos con las respectivas garantías de seguridad y calidad. Aparentemente, EJIS pretende establecer la Interoperabilidad en lo que antes no era más que papel, puede ser que este proyecto en caso de lograrse con

satisfacción sea una verdadera revolución dentro del sistema de Justicia de España, pues se logrará sincronizar la información existente entre los sistemas que sean interoperables mejorando así sus procesos, pues se automatizará el traslado de la información, evitando que existan retrasos. Se destaca también el hecho de que se puede compartir la información evitando ser víctimas de trámites y demoras innecesarias.

Evidentemente, este proyecto tiene de cierta forma objetivos en común con esta investigación, por lo que es una solución a tener en cuenta y seguir muy de cerca. Entre las tecnologías que usa este proyecto se encuentra una arquitectura SOA, utiliza también un Bus de Integración y el almacenamiento lo hace a través de XML, como se mencionó anteriormente. Hay que resaltar que no se rige por el mismo sistema de leyes ni regulaciones que nuestro país, así como tampoco existe la misma situación ni las mismas necesidades del sistema judicial cubano, por lo que es uno de los obstáculos a la hora de tomar esta alternativa como solución al problema de la presente investigación. En el [anexo 2](#) se muestra la imagen de la arquitectura que se utiliza para la implementación de EJIS, tomada de (ortíz).

#### **1.4.3 Proyecto ATLANTE.**

ATLANTE es un sistema de información que integra a todos los órganos judiciales de la Comunidad Autónoma de Canarias en una red privada judicial dentro de la red corporativa del Gobierno de Canarias. Es una arquitectura tecnológica modular cuyo objetivo es cubrir todas las necesidades de las TIC en la justicia. Actualmente se encuentra en su versión 2.2.1.0 (ATLANTE II) y está desarrollada bajo la tecnología .NET Remoting. Este proyecto, que como se menciona anteriormente se encuentra ya en su versión 2, y que es llamado ATLANTE II, es una solución a tener en cuenta por los problemas que ha resuelto dentro del marco en el que se desenvuelve. Según algunas de las bibliografías consultadas esta solución ha presentado algunos problemas como la caída de servidores por más de una hora, incluso una mañana completa, afectando el trabajo de los juristas.

Lo que más llama la atención es que está desarrollado en .NET, aspecto a tener en cuenta antes de decidirse por esta solución, pues no todos tienen acceso a esta tecnología, por la razón de que es propietaria y hay que pagar cierta cantidad de dinero para acceder a ella, destacando que nuestro país se encuentra bloqueado por la mayor potencia económica del mundo. El hecho de que esté desarrollado en .Net puede traer problemas una vez que se quiera o se haga necesaria la integración con otros sistema que por causa de estar

desarrollado en otra plataforma que no sea .NET se incompatible con esta solución, situación que entorpecerá cualquier intento de colaboración y será un obstáculo difícil de tratar, pues se necesitaría de ciertos cambios entre los sistemas que opten por dar este paso de avance hacia una interacción en busca de la eficiencia.

#### **1.4.4 Proyecto SIMAC (Sistema Integral Multicanal de Atención al Ciudadano)**

SIMAC es un proyecto que arranca en el año 2005, su objetivo es transformar el modelo de atención al ciudadano de la Generalitat a través de un nuevo modelo de prestación de servicios acompañado de una nueva plataforma tecnológica (Plataforma de interoperabilidad). Esta plataforma proporciona un único punto de acceso integrado, una interfaz común de peticiones y respuestas de los clientes o usuarios, un mecanismo de control de acceso y autorización y un conjunto de funcionalidades de gestión y administración de las peticiones realizadas. El sustento de dicha plataforma no es más que una arquitectura SOA en la cual para la orquestación de servicios se utiliza Oracle BPEL, para el acceso a las bases de datos y la implementación de soporte se utiliza Java J2EE, mensajes en formato XML y tecnología de encriptación de peticiones mediante las firmas con certificados de aplicación.

Esta solución de cierto modo abarca algunas de las tecnologías en las que se pretende enfocar esta investigación, pero hay que señalar que no se desenvuelve en un entorno jurídico, por lo que no es factible optar por su uso como solución a la problemática que se pretende resolver con la presente investigación. Además, no está orientada a las necesidades que apremia la situación del Sistema de Justicia en Cuba. Hay que destacar que posee variantes que sí son factibles incorporar a la plataforma que se pretende desarrollar en Cuba, pues como se puede apreciar posee una arquitectura orientada a servicios, un único punto de acceso y para su desarrollo utiliza estándares abiertos.

#### **1.4.5 Proyecto de Interoperabilidad en Chile (Minsegpres).**

Es una plataforma integrada que tiene como objetivos mejorar la calidad de la información, los servicios y atención a las personas, facilitar los procedimientos administrativos asociados a los trámites de modo de no solicitar a las personas la información que el Estado posee, transparentar ante el ciudadano la información que tiene el Estado respecto a él. Sirve como medio de traspaso de información entre organismos y que permite que la información de las personas que el Estado conserva esté disponible para la realización de trámites en los

diferentes organismos según las competencias pertinentes. Facilita que la información entregada cumpla con los requisitos legales de privacidad de los datos y con las normativas de intercambio de datos entre organismos. Además, comunica distintos tipos de tecnología presentes en los organismos del Estado, así como también garantiza altos niveles de seguridad tanto en el acceso como en las transiciones de la información. También permite la interoperabilidad con condiciones y estándares predefinidos. Hay que destacar que a pesar de que este proyecto no se desenvuelve en el marco jurídico, dentro de su arquitectura utiliza SOAP, el protocolo HTTP, un motor de orquestación (BPEL) que se encarga de la lógica de proceso y cuenta con un Enterprise Service Bus (ESB) que se encarga del ruteo inteligente, transformación de datos, así como su publicación y suscripción.

Esta solución trae muchas ventajas, entre ellas, disminuye el tiempo de trámite a los ciudadanos, proporciona una base escalable de interoperabilidad para los servicios públicos, uso más eficiente y racional de las plataformas tecnológicas de los servicios públicos, entre otras.

A pesar de que no es una solución que se desenvuelva específicamente en el ámbito jurídico( aunque si está contenido en su campo de acción), sino en los servicios públicos, es una solución que se puede tener en cuenta a la hora de describir y fundamentar la arquitectura para el proyecto de interoperabilidad que se está llevando en Cuba, primeramente, porque garantiza una serie de objetivos que son comunes entre ambos proyectos y también hay que destacar que implementan un ESB que es de interés, aunque no inmediato, de esta investigación. Además, como bien se plantea, integra varias plataformas y sistemas para que puedan compartir e intercambiar información, que en resumen es interactuar (interoperabilidad), objetivo primordial de la presente investigación.

## **1.5 Herramientas y estándares:**

### **1.5.1 ¿Qué es un estándar?**

Los estándares, según (Tortosa, 2006) *“son acuerdos internacionales documentados o normas establecidas por consenso mundial. Contienen las especificaciones técnicas y de calidad que deben reunir todos los productos y servicios para cumplir satisfactoriamente con las necesidades para las que han sido creados y para poder competir internacionalmente en condiciones de igualdad”*. En otras palabras, los estándares son un modelo a seguir o reglas comunes que deben cumplir los productos o servicios para poder incluirse y competir a nivel

internacional, destacando que estas reglas no son creadas por cualquier persona o institución, son reglas que están bien documentadas, reguladas y justificadas. Los estándares ofrecen una base de comparación, una medida de calidad, cantidad o nivel y un consenso de opiniones entre distintos individuos, grupos u organizaciones.

Existen varias Organizaciones que se centran en el desarrollo y proposición de estándares como son el caso de Oasis<sup>2</sup> y la W3C<sup>3</sup> que han propuesto varios estándares a disposición de la comunidad mundial tales como XML, SOAP, WSDL, SAWSDL entre otros.

La interoperabilidad se basa en estándares, éstos establecen las reglas generales del juego para el equipo, dicen a donde se quiere llegar, no cómo deben llegar. También sirven como guías aunque no son un manual de comportamiento.

Los estándares, como activadores de la capacidad de comunicación, pueden ser de dos tipos, estándares abiertos o estándares propietarios, ambos con capacidad de ser ampliamente aceptados.

#### **1.5.1.1 Estándares abiertos como pilares de interoperabilidad**

Son especificaciones técnicas disponibles públicamente, sin, restricciones de acceso o implementación por parte de ningún fabricante y desarrolladas mediante un proceso de consenso. Según la bibliografía consultada, los estándares abiertos deben permitir a todos los posibles competidores operar en base a un igual acceso a la habilidad de implementar el estándar. Además, permiten interactuar en un mercado de libre competencia entre múltiples implementaciones de hardware y software, sin estar sujetos a condicionantes de tecnologías propietarias. Hay muchas definiciones que los abordan desde diferentes perspectivas, pero en todas ellas se identifican características o aspectos comunes como que debe existir un proceso abierto alrededor del desarrollo del estándar, no debe haber propiedad intelectual asociada y la documentación debe ser de carácter público y sobre todo gratuito.

---

<sup>2</sup> Consorcio internacional sin ánimo de lucro que gestiona el desarrollo, convergencia y adopción de estándares sobre comercio electrónico.

<sup>3</sup> El consorcio World Wide Web es un consorcio internacional donde las Organizaciones miembros y el personal asociado trabajan en conjunto en el desarrollo de estándares web.

## 1.5.2 SGML

Son las siglas en inglés para Standard Generalized Markup Language (lenguaje estándar de marcado generalizado). *“Sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial”* según (Bastarrica, et al., 2009).

SGML no es más que un estándar que define los métodos para la representación de la información que abarca otros estándares como XML y HTML, entre otros, siendo la raíz de esta gran familia.

## 1.5.3 XML (Extensible Markup Language) como lenguaje de intercambio de datos

*“Es un lenguaje abierto que se ha desarrollado desde 1996 como un subconjunto de SGML y que ha sido adoptado por la W3C desde febrero de 1998. Permite describir el sentido o la semántica de los datos, pues separa el contenido de la presentación describiendo el contenido a través de etiquetas o marcas”* según (Bastarrica, et al., 2009).

Por otro lado, Ramón Montero Ayala (Ayala) define XML como “un lenguaje de marcas basado en SGML, capaz de describir cualquier tipo de información en forma personalizada, aunque también es un metalenguaje de marcado capaz de describir lenguajes de marcas adecuadas para aplicaciones concretas”, alega además que es *un conjunto de normas que permiten tratar información muy diversa desde muchos puntos de vista y sistemas diferentes, siendo el propio diseñador el encargado de decidir el proceso más adecuado a cada caso y, en consecuencia XML es un sistema complejo de descripción de información libre y rigurosa*. En resumen, XML es un lenguaje de marcas o lenguaje de modelado que se encarga más del contenido de los datos que de su presentación a través de etiquetas que marcan su estructura. Además, XML puede ser un contenedor de datos y es mucho más flexible que HTML, pues permite definir detalladamente la estructura de un documento sin necesidad de utilizar DTD.

Un documento XML tiene varias formas de representación, el hecho de que un documento esté en este formato brinda muchas ventajas como portabilidad, depuración, independencia de la plataforma y sobre todo su edición para posteriores cambios en su estructura. Además, XML es toda una familia de tecnologías con distintas especificaciones que cuenta con suficiente soporte y documentación a pesar de ser relativamente nuevo. Una de las razones para optar por XML es que no requiere de licencias, no pertenece a ninguna compañía y es un estándar internacionalmente reconocido.

Respecto a las herramientas que permiten el análisis, edición y desarrollo de XML existen variedades, algunas de las más usadas y citadas según la bibliografía consultada son XML-SPY, XML-WRITER y Visual Tools, ésta última perteneciente a IBM. Además, posee enlaces dinámicos que se pueden actualizar automáticamente siguiendo una huella de una página y es posible el uso de estilos, solo sería necesario crearlos en los DTD según (Posadas, 2000). En cuanto a la transferencia de datos XML es bastante flexible, a continuación se citan algunos ejemplos en los que se puede utilizar según (Microsoft, 2011):

- Un documento normal.
- Un registro estructurado, como un registro de citas o un pedido.
- Aplicaciones web de internet o intranet que mueven datos.
- Un objeto con datos, como el formato persistente de un objeto o un control ActiveX.
- Un registro de datos, como el conjunto de datos de una consulta.
- Una presentación gráfica, como una interfaz de usuario de la aplicación.
- Vínculos entre información y personas del web.
- Documentos de descubrimiento utilizados para buscar servicios web XML.

Existen también algunas restricciones, las que más llaman la atención son las de sintaxis, por ejemplo, no es permitida la anidación incorrecta de elementos, es obligatorio cerrar las etiquetas, los atributos deben ir entrecomillas dobles, entre muchas otras.

#### **1.5.4 XML-Schema para la estandarización de datos y documentos XML**

Según (Bastarrica, et al., 2009) “*se trata de un documento de definición estructural al estilo de los DTD, que además cumple con el estándar XML y permite expresar mayor diversidad de documentos*”. Los documentos XML Schema (usualmente con extensión XSD) se concibieron como un sustituto de los DTD, teniendo en cuenta los puntos débiles de estos y buscando mejores capacidades a la hora de definir estructuras para los documentos XML, como la declaración de los tipos de datos.

Por otro lado, la Universidad de las Ciencias Informáticas (UCI), define los documentos XSD como “*la descripción de la estructura de la información contenida en un archivo XML y de sus reglas. Se requiere para que los sistemas que transmiten o reciben archivos XML puedan*

validar la conformación de estos archivos o reglas definidas por los autores” (Universidad de las Ciencias Informáticas, 2007).

### **1.5.5 WSDL para la descripción de los contratos o interfaces de los servicios**

“Es un formato XML que describe los servicios de red como un conjunto de endpoints que procesan mensajes contenedores de información orientada tanto a documentos como a procedimientos” según (Tortosa, 2006). Otros autores como Francisco Domínguez Mateos, (Mateos, 2005), plantean que “es básicamente información XML en la que se aporta una descripción de las interfaces que intervienen en determinados servicios junto con los protocolos de invocación que soporta”.

Se puede decir que WSDL no es más que un documento XML que describe a un servicio web o sus interfaces definiendo los mecanismos de acceso a los servicios web.

### **1.5.6 Protocolo REST (Representational State Transfer)**

Según la bibliografía consultada REST, es un nuevo estilo arquitectónico para construir aplicaciones distribuidas basadas en los principios que hicieron exitosa la web modelando un servicio como la aplicación de una serie de operaciones fijas sobre un conjunto de recursos posibles de referenciar universalmente. Tiene el potencial para permitir la construcción de aplicaciones distribuidas mucho más escalables e interoperables, compuestas por cientos de servicios desacoplados entre sí. Está fuertemente basado en HTTP <sup>4</sup>1.1; aunque es independiente de este protocolo, es el único utilizado masivamente diseñado para soportar los principios REST.

Por otro lado, en el documento que contiene los lineamientos de arquitectura para los sistemas informáticos que pertenecen a la intranet de la Universidad de las Ciencias Informáticas, se plantea que REST “es un estilo arquitectónico y no un estándar definido. Se basa en las URI para identificar recursos, los cuales se retornan en XML puro, es también una manera simplificada de ejecutar métodos remotos”. (Universidad de las Ciencias Informáticas, 2007).

---

<sup>4</sup>Protocolo de Transferencia de Hipertexto: es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes web y los servidores HTTP.

### 1.5.7 Protocolo SOAP (Simple Object Access Protocol)

Según (Tortosa, 2006) SOAP *“es un estándar que define un protocolo que da soporte a la interacción (datos + funcionalidad) entre aplicaciones en entornos distribuidos y heterogéneos, es interoperable (neutral a la plataforma, lenguajes de programación, independiente del hardware y protocolos). Define cómo organizar la información de una manera estructurada para intercambiarla entre los distintos sistemas”*.

Por otra parte (Naranjo, 2007) define SOAP como *“una arquitectura conceptual que organiza funciones de negocio como servicios interoperables permitiendo reutilización deservicios para satisfacer necesidades de negocio. Está basado en estándares y con independencia de los fabricantes”*. Alejandro Botello (Botello, 2005) plantea que *“es un protocolo de mensajería para que puedan ser enviados a través de la red utilizando protocolos de transporte como HTTP, SMTP<sup>5</sup> u otros”*. En otras palabras, es una arquitectura o estilo arquitectónico para la construcción de aplicaciones donde se juntan las tecnologías de la información con los procesos de negocio.

### 1.5.8 Comparación entre SOAP y REST

La Universidad de las Ciencias Informáticas (Universidad de las Ciencias Informáticas, 2007) realiza una comparación entre SOAP y REST, la cual se cita a continuación.

#### **SOAP:**

- Los mensajes se representan en una estructura XML definida y estandarizada (“envelope”).
- Puede utilizarse sobre protocolos como HTTP, SMTP, etc.
- El acceso y la manipulación de datos es específica a la aplicación.
- La seguridad no se define a nivel de SOAP y debe ser implementada por el desarrollador.
- Esquemas XML (XML Schema) son utilizados para definir contratos o interfaces entre un cliente y el servicio.

#### **REST:**

---

<sup>5</sup> Simple Mail Transfer Protocol por sus siglas en ingles, es un servicio de correo complejo a través de servidores usando un protocolo estándar para enviar y recibir el correo.

- Los mensajes son representados en XML plano.
- Se utiliza HTTP como protocolo de transporte.
- Se utilizan comandos de HTTP para acceder a los datos (GET, POST, DELETE, PUT).
- Se utilizan URI para identificar recursos de manera única.
- Se utiliza HTTPS para brindar seguridad.
- No hay un método formal para expresar los métodos brindados (“contrato”).

Teniendo en cuenta los aspectos que se comparan, se puede apreciar que es más factible optar por SOAP, pues, REST está fuertemente arraigado al protocolo HTTP, imposibilitando la utilización de protocolos como FTP y SMTP. Por otro lado, la seguridad depende del protocolo HTTPS, por lo que si el mismo es evadido, se compromete toda la seguridad. Además, hay que tener en cuenta que la presente investigación forma parte de la propiedad de la Universidad de las Ciencias Informáticas y ésta ya tiene experiencias respecto al uso de SOAP en sus aplicaciones, lo que representa una gran ventaja para el equipo de desarrollo de la plataforma de interoperabilidad. También hay que tener en cuenta que los mensajes se encuentran en XML planos, mientras que SOAP los representa en estructuras XML bien definidas. Como factor decisivo para optar por uno de los dos protocolos, se tiene que los contratos o interfaces en SOAP son definidos a través del lenguaje WSDL utilizando esquemas XML. Sin embargo, REST no posee un método formal para expresar dichas interfaces, lo cual es una gran desventaja, pues, los contratos son la base para el desarrollo de un servicio web.

### **1.5.9 UDDI (Universal Description, Discovery and Integration)**

Muchas son los autores que hacen referencia al estándar, por ejemplo, Salvador Otón Tortosa (Tortosa, 2006) plantea que UDDI *“Es un directorio que contiene un registro o repositorio de descripciones de servicios web. Tiene 2 objetivos fundamentales, primero servir de soporte a los desarrolladores para encontrar información sobre servicios web y poder construir clientes; segundo, facilitar el enlace dinámico de servicios web, permitiendo consultar referencias y acceder a servicios de interés”*. Por otro lado, Pedro Espina Martínez, (Martínez, 2007) , plantea que UDDI está comúnmente considerado en la actualidad como la

pieza angular de las Arquitecturas Orientadas a Servicios SOA, definiendo un método estándar de publicación y descubrimientos web.

Se puede resumir que, UDDI es un estándar que permite gestionar un registro o repositorio de servicios web, en cuyo contenido se encuentran las inscripciones, interfaces o contratos de dichos servicios para su localización y acceso automático; se apoya en algunos estándares que han sido muy aceptados, tales como XML y SOAP entre otros.

Existen varias implementaciones de UDDI, entre los que se encuentra JUDDI. Este es un servidor UDDI de código abierto. Entre las características más sobresalientes se encuentran que está desarrollado en Java. Esto implica que sea independiente de la plataforma y pueda ser utilizado en la mayoría de las plataformas que soportan este lenguaje. También permite una fácil integración con los sistemas de autenticación existentes y soporta su uso con cualquier base de datos relacional<sup>6</sup> como SQL, ORACLE entre otras.

## **1.5.10 Servidores de aplicación**

### **1.5.10.1 JBOSS**

JBoss es un servidor de aplicaciones J2EE implementado en Java y es bastante popular en el mundo empresarial. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que soporte Java, por lo que es independiente de la plataforma. El producto JBOSS es únicamente un "EJB Container" y es por esto que generalmente se utiliza en conjunción con un "Web-Container". Ofrece una plataforma de alto rendimiento para aplicaciones de e-negocio y puede ser utilizado sin ninguna restricción como el caso de las licencias.

Entre las principales características que se pueden apreciar se encuentran que es de código abierto, cumple con los estándares y se dice que es muy confiable a nivel de empresa. Por otro lado, está orientado a servicios y posee una flexibilidad inmutable y sobre todo posee abundante documentación.

### **1.5.10.2 EAServer**

EAServer es un conjunto integrado de servidores de aplicación que se utilizan para desplegar aplicaciones web con soporte a tráfico de alto volumen, contenido dinámico y procesamiento

---

<sup>6</sup>Base de datos en la que los datos visibles al usuario se encuentran organizados como tablas de valores, donde todas las operaciones sobre la base de datos son sobre estas tablas.

intensivo de datos en línea. Entre sus principales características se encuentra que es un servidor configurable y de fácil administración, posee una interfaz de usuario basada en web que utiliza Java Script, aunque funciona normalmente en caso de que no esté habilitado. Además, gestiona las conexiones y tiene soporte completo para JMS<sup>7</sup>.

### **1.5.10.3 GlassFish**

GlassFish es un servidor de aplicaciones de software libre que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La base de GlassFish es el servidor Sun Java Application Server. Entre sus principales características se encuentra que posee un módulo de persistencia conocido como TopLink<sup>8</sup>.

### **1.5.10.4 Jonas**

Jonas es un servidor de aplicaciones abierto de Java, es una puesta en práctica delantera de Open Source por OW2 de la especificación de Java EE autorizado bajo LGPL. Fue el primer servidor de código abierto certificado gratuitamente. Posee entre sus características una consola de administración remota bastante clara, proporcionando herramientas accesibles para la carga e implementación remota de aplicaciones web, configuración de puertos de acceso web seguros y gestionar los conjuntos de datos (usuario, nombre, contraseña). La consola está basada en el lenguaje Java Script.

Evidentemente existen diversas opciones para decidirse por un servidor de aplicaciones, los que se mencionan anteriormente se encuentran entre los más usados en la actualidad, aunque no son todos. Sin embargo, se decide utilizar como uno de los componentes de la arquitectura a JBoss, pues, según la bibliografía consultada, se encuentra entre los preferidos de muchos hoy en día. Hay que tener en cuenta que JBoss cuenta con abundante documentación, ya sea en inglés o español, por otro lado, en la UCI se cuenta con experiencias en cuanto a su uso, representando una gran ventaja. También influye que presenta varias configuraciones que son ideales dependiendo del uso que se le va a dar al servidor de aplicaciones, posibilitando que se configure acorde a las distintas necesidades. Por otro lado, existen abundantes foros en internet a los cuales remitirse en caso de existir alguna duda que no sea erradicada a través de la documentación, y como factor muy

---

<sup>7</sup> Java Message Service, es un estándar de mensajería creado por Sun Microsystem.

<sup>8</sup> Módulo de persistencia del servidor de aplicaciones GlassFish dotado por Oracle Corporation.

importante se tiene que se mantiene actualizado, es decir, que está en constante desarrollo, mejorando así sus funcionalidades.

## **1.5.11 Herramientas y componentes de seguridad**

### **1.5.11.1 Acaxia**

Acaxia es una solución desarrollada por la Universidad de las Ciencias Informática con base en el software libre. Incorpora procesos como la administración de perfiles y usuarios, además permite la gestión dinámica multi-tema, multi-entidad, multi-escritorio, multi-sistema y multi-idioma. Entre los procesos de seguridad que posee este componente, se encuentra la administración de conexiones, autenticación, autorización, auditoría y como se menciona anteriormente la administración de perfiles. Otra de las funcionalidades que brinda es la del control de acceso a los servicios web y más importante aún, posee tres niveles de integración con otros sistemas para aprovechar sus funcionalidades (nivel de interfaz, servicios web y servicios internos).

En general, la mayoría de las soluciones para gestionar la seguridad y que implementan Single Sign-On que existen actualmente, se centran en los procesos de autenticación y autorización, los mismos juegan un papel fundamental en la seguridad de aplicaciones, permitiendo un mejor control sobre el acceso a la información. Acaxia incorpora además de estos dos procesos, como se menciona anteriormente, los procesos de administración de perfiles y usuarios, la administración de conexiones y el proceso de auditoría. [Ver anexo 2.](#)

### **1.5.11.2 Kerberos**

Kerberos es un servicio de autenticación que implementa Single Sign-On, cuyo objetivo es desempeñar el rol de un tercero confiable dentro de la infraestructura informática local. Está compuesto por una base de datos con información de autenticación centralizada, junto a varios servicios que interactúan con la base de datos para permitir que los usuarios sean autenticados sin necesidad de transferir contraseñas en texto plano por la red. Como servicio de directorio para obtener un repositorio centralizado con la información de las cuentas de usuario utiliza LDAP<sup>9</sup>, el mismo sirve como solución en caso de falla de Kerberos.

---

<sup>9</sup> Se trata de un servicio de directorio, es una base de datos que permite armar una estructura jerárquica con la información de la organización.

### **1.5.11.3 SAP NetWeaver**

Es una plataforma que es compatible con los métodos de autenticación de los proveedores de terceros, incluidas las tarjetas inteligentes y Single Sign-On que son capacidades para aplicaciones web basadas en estándares abiertos como SAML y los certificados de seguridad. Es una plataforma de tecnología integrada para aplicaciones SAP, que está orientada a servicios y a la integración. Su construcción está basada en estándares abiertos y plantea a las empresas trabajar todas sus aplicaciones en una única plataforma integrada con una estructura firme.

En resumen, se decide utilizar Acaxia como uno de los componentes de la arquitectura de interoperabilidad, de forma que se gestione la seguridad a través de dicha herramienta. Entre los factores que influyeron en la decisión se encuentran que, fue desarrollada por la UCI sobre software libre, por lo que la misma cumple con los requerimientos que rigen la universidad y el país actualmente. Esto influye positivamente en el equipo de desarrollo de la plataforma de interoperabilidad, pues, en caso de existir contratiempos o dudas es posible consultar con los especialistas de la universidad.

Por otro lado, Acaxia cuenta con suficiente experiencia, pues la misma es utilizada por varias empresas a lo largo del país. Además, como se menciona anteriormente, incorpora varios procesos que no se encuentran presentes en las demás soluciones que se encargan de gestionar la seguridad, pues las mismas solo se enfocan en los procesos de autenticación y autorización. Partiendo de lo anterior, se observa que es una herramienta bastante robusta y completa, brindando una buena seguridad a las empresas y sistemas informáticos que la utilicen.

### **1.5.12 Algunos estándares o protocolos de seguridad**

Además de las herramientas que se mencionan anteriormente, existen diversos protocolos que permiten gestionar la seguridad a distintos niveles tales como el cifrado de datos y documentos y el aseguramiento de las transacciones entre los servicios web, entre otros. A continuación se mencionan algunos de los que se decide incorporar a la arquitectura de la plataforma de interoperabilidad.

### 1.5.12.1 Web Services Security

Este es un protocolo de comunicación que provee los medios necesarios para aplicar seguridad a los servicios web, contiene especificaciones que de cierta forma garantizan la integridad y seguridad en mensajería entre servicios web. WS-Security incluye formatos de certificado como X.509 que es un estándar UIT-T para infraestructuras de claves públicas que especifica entre otros aspectos formatos estándar para certificados de claves públicas. Además, permite el uso de tecnologías de encriptación como las PKI (Public Key Infrastructure) y el protocolo Secure Socket Layer / TransportLayer Security (SSL/TLS), que son elementos claves a tener en cuenta siempre que se quiera garantizar seguridad en cualquier sistema informático.

### 1.5.12.2 XML-Encryption

Es un lenguaje cuya función principal es la de asegurar la confidencialidad de partes de documentos XML a través de la encriptación parcial o total de un documento transportado. Como ventaja tiene que se puede aplicar a cualquier recurso web, no está restringido solamente a contenido XML. También permite granular el cifrado de documentos XML haciendo posible el cifrado a varios niveles que van desde elementos básicos hasta documentos completos. Algunos de estos niveles son:

- **Cifrado de un elemento:** Se cifra solamente un elemento.
- **Cifrado de elementos hijos:** Solo se cifran datos específicos de un elemento.
- **Cifrado del contenido textual:** Se puede cifrar el contenido textual de un elemento.
- **Cifrado de documentos completos:** Se cifra todo el documento XML.
- **Súper cifrado:** Se cifran datos que ya han sido cifrados, es decir, es un doble cifrado.

### 1.5.12.3 XML Digital Signature

Es un estándar que provee mecanismos para la creación de firmas digitales a partir de XML. Asegura la integridad de partes de un documento XML transportado y define un esquema para capturar el resultado de la operación de firmas digitales y aplicarlas a los metadatos,

además proporciona la autenticación de mensajes y/o servicios de autenticación de firma para datos de cualquier tipo sin importar si se encuentran en el XML que incluye la firma o no.

En resumen, XML Digital Signature asocia claves con los datos de consulta y representa un sistema que a través de una firma digital ofrece la autenticidad de los datos y el mensaje así como su integridad.

#### **1.5.12.4 Security Assertion Markup Language (SAML)**

SAML proporciona un idioma común para el despliegue de toda la infraestructura de seguridad, además, permite compartir información de autenticación entre aplicaciones. Permite intercambiar información de identificación y está basado en XML. SAML proporciona un protocolo de mensajes tipo pregunta-respuesta para que los dominios implicados puedan gestionar las decisiones necesarias de autenticación y autorización, esto puede ser aplicado a los servicios web para proveer seguridad a las transacciones de los mismos. Una de las grandes ventajas que posee además de las ya mencionadas es que es independiente de la plataforma.

#### **1.5.13 Servicios Web (SW)**

José Javier Samper Zapater plantea que *“son un sistema de software diseñado para soportar la interacción entre dos máquinas a través de la red. Poseen una interfaz descrita en un formato que puede ser interpretado por una computadora, específicamente WSDL”*, (Zapater, 2005). Por otro lado, Pablo Valledor (Valledor Pellicer) plantea que los servicios web *“son componentes de software que permiten que varias aplicaciones se comuniquen entre sí con el objetivo de intercambiar información”*.

En otras palabras, un servicio web no es más que una publicación de algún servicio en la red por parte de algún sistema o aplicación informática al cual se accede a través de una interfaz con el objetivo de interactuar con la información que este brinda, para lo cual utiliza varios estándares y protocolos tales como WSDL que describen las interfaces de los servicios, UDDI para realizar el descubrimiento de los servicios web y SOAP o REST para la comunicación entre los servidores y los clientes que brindan y consumen los servicios.

#### **1.5.14 Razones para utilizar los Servicios Web**

Los Servicios Web cumplen una funcionalidad específica, los protocolos que los soportan normalmente se comunican por el puerto 80, por lo que su acceso es muy parecido al de una

página web, la principal diferencia es que a una página web accede cualquier usuario a menos que esté protegida por alguna forma de autenticación, sin embargo, al servicio solo accederá quien se autorice. Los servicios web fortalecen el desarrollo de aplicaciones distribuidas, pero para que tengan éxito deben cumplir ciertos requisitos como:

- Preferiblemente estar basados en estándares abiertos (XML, SOAP, WSDL, etc.), así no estarán sujetos a restricciones para quienes necesiten hacer uso de los mismos.
- Requerir la cantidad mínima de infraestructura.
- Deben ser flexibles en cuanto a la integración, pues no es objetivo que una vez que se desarrolle un servicio no se pueda integrar con las demás aplicaciones con que se quiere interactuar.
- Deben estar centrados en mensajes y documentos.

Entre las ventajas que ofrecen o limitan los servicios web, Alejandro Botello, en un informe para el 1er Simposio de tecnologías de la Información y las Tecnologías (Botello, 2005) plantea lo siguiente:

#### **Ventajas:**

- Permite crear nuevos desarrollos a un costo menor y más rápido debido a la reusabilidad.
- Se pueden reutilizar los sistemas legados.
- Integra el proceso de negocio entre proveedores y socios a un menor costo.
- Se puede ampliar el mercado de negocios al integrar los SW disponibles para nuevos desarrollos.

#### **Desventajas:**

- Para describir los SW es necesario que el registro UDDI sea confiable.
- Es una tecnología relativamente nueva, lo que puede traer problemas con la implementación.
- Los estándares que se encargan de la seguridad de los SW todavía se encuentran en desarrollo, por lo que este proceso requiere en parte de un esfuerzo y conocimiento mayor.

- La integración del proceso del negocio, la administración de las transacciones y la confiabilidad de políticas no están implementadas completamente.

La principal razón para optar por esta vía o forma de compartir información, es que la mayoría de las empresas que poseen en su dominio algún sitio o página web protegen sus redes a través de firewalls o cortafuegos, bloqueando gran parte del flujo de tráfico hacia o desde internet. Estos sitios web, generalmente se encuentran bajo el protocolo HTTP sobre TCP en el puerto 80 o 443 en caso de que se utilice SSL, siendo estos los puertos que usualmente se encuentran abiertos para estos servicios con los que se interactúa a través de los navegadores web.

### **Conclusiones del Capítulo:**

En el presente capítulo se realizó un estudio acerca de algunas de las soluciones de interoperabilidad que existen en el mundo, obteniéndose que ninguna satisface las necesidades que apremian la presente investigación, pues estas soluciones en su mayoría tienen un alto valor monetario y no están concebidas bajo las mismas condiciones que el sistema de Justicia en nuestro país. Se refleja el diseño metodológico del diagnóstico aplicado a los órganos y organismos que intervienen en el proceso de Justicia Penal, a partir de cual se obtuvieron datos que contribuyeron a la propuesta de solución. También se realiza un análisis acerca de las tecnologías, herramientas y estándares que permiten el desarrollo de una arquitectura de interoperabilidad, arribando a la conclusión de que se utilizará XML como lenguaje de intercambio de datos, por ser el estándar propuesto por la CEPAL para realizar esta función y ser un estándar aceptado y reconocido a nivel mundial.

Como servidor de aplicaciones se decide utilizar JBoss por ser multiplataforma, manejar la persistencia, poseer abundante documentación y sobre todo, porque en la Universidad de las Ciencias Informáticas existen experiencias concretas en cuanto a su uso. Por otro lado, se decide optar por Acaxia como herramienta para gestionar la seguridad de forma centralizada, pues la misma incorpora procesos que las soluciones existentes actualmente no incorporan. Como factor decisivo para su uso, destaca que es una solución propuesta por la UCI. Se utilizan además de Acaxia, estándares como SAML, WS-Security, XML Digital Signature y XML Encryption para complementar una seguridad más robusta, tanto interna como externa.

También se decide utilizar UDDI, específicamente la versión 3.0, para implementar el repositorio de servicios web que contendrá las interfaces de los mismos, así como se utiliza

JUDDI como servidor del propio repositorio y UDDIBrowser como buscador dentro del mismo por ser ambas aplicaciones desarrolladas en Java, lo que potencia el uso de los mismos en cualquier plataforma que soporte este lenguaje.

## Capítulo 2: Descripción de la solución propuesta.

### INTRODUCCIÓN

En el presente capítulo se describen los principales requerimientos de la Plataforma de Interoperabilidad, explicando brevemente en qué consisten. Se realiza además la descripción del diseño de la arquitectura de interoperabilidad del Órgano de Justicia-MININT, así como sus componentes. Se define qué variante de configuración del servidor de aplicaciones se utiliza, así como los estándares de seguridad que se tuvieron en cuenta para gestionar y asegurar la misma. Por otro lado, se definen las interfaces de los servicios web a partir de los cuales se intercambiarán y compartirán los datos por cada una de las partes involucradas en el proceso.

#### 2.1 Requerimientos de la Plataforma de Interoperabilidad para el Sistema de Justicia Penal en Cuba.

- **Autenticación a través de certificados digitales:** Cada una de las aplicaciones informáticas con que cuenta el órgano de justicia penal deberá implementar sus certificados digitales pues la plataforma solicitará los mismos cuando se intenten autenticar en la misma.
- **Solo accede las partes involucradas en el proceso de Toma de Decisiones:** Solo tendrán acceso a la plataforma en esta primera etapa o fase, los titulares de la FGR, el TSP, el MININT y demás funcionarios de la Comisión Nacional del Órgano de Justicia-MININT.
- **Seguridad, integridad, disponibilidad, consistencia y autenticidad de los datos:** Desde que un usuario o un sistema informático se autentica en la plataforma, ésta debe proveer una completa seguridad a todas las acciones que se realicen mientras se encuentren autenticados, centrándose en aspectos como la autenticidad, integridad y disponibilidad de los mismos.
- **Acceso restringido a la información:** El acceso a la información que se encuentre disponible en la plataforma será mediante los permisos o el nivel de acceso de cada usuario o sistema informático, pues éstos serán otorgados a los certificados digitales, asegurando que solo se pueda acceder a la información que le está permitida.

- **Intercambio de datos basado en XML:** Los datos que se encuentren dentro de la plataforma o que transiten a través de ésta deben estar basados en XML. Con esto, entre otras cosas, se hace posible brindar más seguridad a los datos.
- **Estandarización de los esquemas:** los esquemas deben estar estandarizados para lograr un entendimiento entre los sistemas informáticos que acceden a la plataforma asegurando mayor agilidad y eficiencia en el Proceso de Justicia Penal.
- **Uso de Software libre:** Debido a que Cuba se encuentra inmersa en el proceso de migración hacia software libre, es necesario que para el desarrollo de la plataforma se utilicen herramientas que operen bajo software libre, así como los sistemas operativos sugeridos a utilizar son Debian o Ubuntu en la versión más reciente y estable posible.
- **Uso de estándares abiertos:** Será obligatorio el uso de estándares abiertos para el desarrollo de la plataforma, pues esto conlleva a un ahorro de recursos además de brindarle portabilidad y flexibilidad a la plataforma en general.
- **Escalabilidad:** Es necesario que la plataforma sea escalable, es decir, que permita evolucionar las tecnologías, puesto que, de no ser así, puede ser que cada vez que se necesite realizar cambios, haya que construir todo partiendo del principio o realizando drásticos cambios en la estructura de la arquitectura de la plataforma.

## 2.2 Tipología estructural y tipos de datos.

Por parte del equipo de desarrollo de la plataforma de interoperabilidad, se definieron varios tipos de datos posibles a los resultados que retornan los servicios web, dichos datos tienen un formato definido y estandarizado a través de XML que no afecta la estructura de los datos en sí. De acuerdo a la estructura de los datos se identifican:

- Datos simples.
- Datos complejos.
- Datos secuenciales.
- Datos tabulados.

Los datos simples son aquellos que tienen una definición atómica y no se descomponen en otros datos. Por su parte, los datos complejos son los que se dividen o separan en datos que complementan la definición del dato general. Los datos tabulados son los que se refieren a la

combinación de varios indicadores en una representación matricial. Por último, los datos secuenciales son aquellos que se representan en forma de listas o arreglos.

### **2.3 Datos a los que responden los Servicios Web**

Los servicios web que serán implementados por cada uno de los organismos que aportan datos al proceso de Justicia Penal. Los datos que aportan estos servicios corresponden a los indicadores del Parte Mensual Informativo generados en su mayoría en la FGR a través de los sistemas de gestión de información internos de dicho organismo, así como del TSP y el MININT. En esta primera fase de la plataforma, solo se hará énfasis en los datos que corresponden a la especialidad de Procesos Penales.

**Tabla 1: Reportes por especialidad.**

<b>Especialidad</b>	<b>Reportes (Indicadores)</b>
Procesos Penales	Total EFP radicados (IND. 3.1)
Procesos Penales	Total de EFP controlados (IND. 3.7)
Procesos Penales	Total de entrevistas realizadas a detenidos (IND. 3.5.3)
Procesos Penales	Total de acciones de instrucción en las que participa el fiscal (IND. 3.7.3)

### **2.4 Procedimiento de estandarización del formato de intercambio de datos**

Para poder llevar a cabo el proceso de intercambio de datos en la plataforma de interoperabilidad, es imprescindible que estos se encuentren estandarizados, de forma que sean descifrables y legibles por todos los que harán uso de los mismos. Para ello, miembros del equipo de trabajo encargado del desarrollo de la plataforma de interoperabilidad, definieron un conjunto de actividades que conforman el diseño de una guía metodológica que permitió conducir el proceso de estandarización. A continuación se mencionan dichas actividades.

- Encuentro con las organizaciones involucradas en el proceso.
- Conciliación de las necesidades de intercambio.
- Clasificación de los datos según su naturaleza y estructura.
- Definición de los esquemas de intercambio.

## 2.5 Interfaces de los Servicios Web

Los servicios web cuentan con interfaces o contratos descritos en el lenguaje WSDL, que luego de su definición, permitirán su posterior implementación. Estos contratos dentro de su estructura, cuentan con la definición de los mensajes, binding, tipo de datos de entrada y salida, tipo de mensaje entre otras opciones.

Se han definido 4 contratos o interfaces para los servicios web que brindarán la información por parte de la Fiscalía General de la República al proceso de justicia Penal. Las interfaces definidas tienen como principal característica, que no dependen de la plataforma o alguna tecnología específica para su implementación y posterior utilización, destacando que cada una de las interfaces responde a uno de los indicadores del Parte Mensual Informativo (PMI) presentados anteriormente. En algunos casos, varias de las interfaces responden a un mismo indicador, pues algunos de los indicadores cuentan con mucha información, por lo que si se define una sola interfaz en la cual sea necesario implementar muchas operaciones, y ésta, una vez implementadas y puestas en acción falla, se pierde cualquier posibilidad de acceder a algún dato que responda a dicho indicador.

### Interfaces:

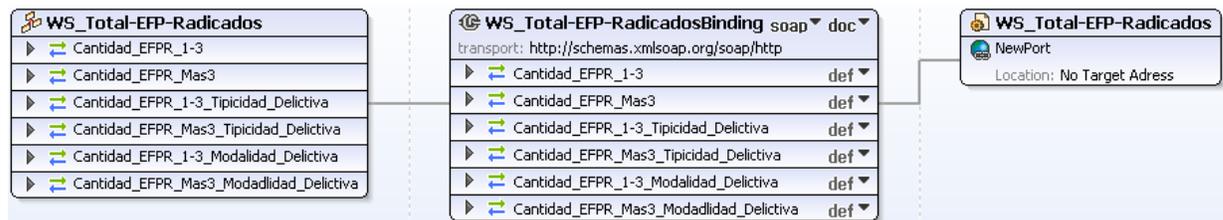


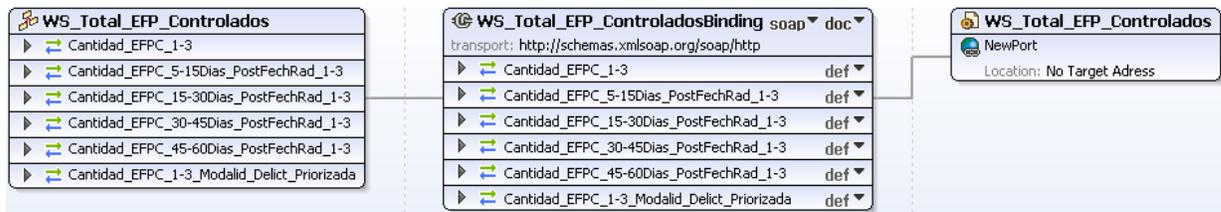
Imagen 1. Interfaz del Servicio que responde al indicador 3.1 Total de EFP Radicados.

La interfaz cuenta con 6 operaciones definidas:

- Cantidad\_EFPR\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Radicados de uno a tres años.
- Cantidad\_EFPR\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Radicados de más de tres años.
- Cantidad\_EFPR\_1-3\_Tipicidad\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo

tabla que contiene los Expedientes en Fase Preparatoria Radicados de uno a tres años clasificados por Tipicidad Delictiva Priorizada.

- Cantidad\_EFPR\_Mas3\_Tipicidad\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene los Expedientes en Fase Preparatoria Radicados de más de tres años clasificados por Tipicidad Delictiva Priorizada.
- Cantidad\_EFPR\_1-3\_Modalidad\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene los Expedientes en Fase Preparatoria Radicados de uno a tres años clasificados por Modalidad Delictiva Priorizada.
- Cantidad\_EFPR\_Mas3\_Modalidad\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene los Expedientes en Fase Preparatoria Radicados de más de tres años clasificados por Modalidad Delictiva Priorizada.



**Imagen 2. Interfaz del Servicio que responde al indicador 3.7 Total de EFP Controlados.**

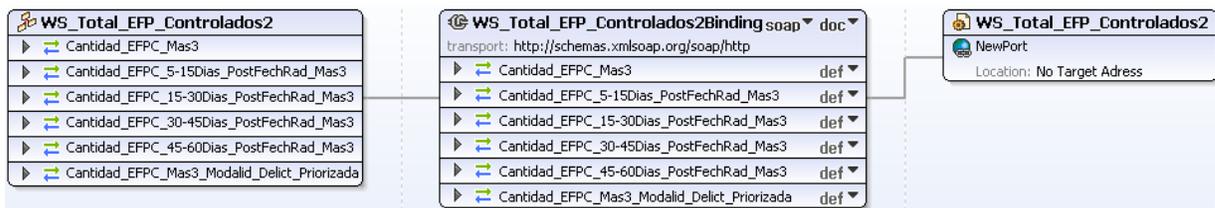
Debido a la cantidad de operaciones que posee este indicador, se definen 2 interfaces para este indicador, una primera interfaz para los Expedientes en Fase Preparatoria Controlados de 1 a 3 años y una segunda interfaz para los Expedientes en Fase Preparatoria Controlados de más de 3 años.

La interfaz cuenta con 6 operaciones definidas:

- Cantidad\_EFPC\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato numérico que indica la cantidad de Expedientes en Fase Preparatoria Controlados de uno a tres años.
- Cantidad\_EFPC\_5-15Dias\_PostFechRad\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato numérico que indica la

cantidad de Expedientes en Fase Preparatoria Controlados de uno a tres años que tengan de cinco a quince días posteriores a la fecha de radicación.

- Cantidad\_EFPC\_15-30Dias\_PostFechRad\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato numérico que indica la cantidad de Expedientes en Fase Preparatoria Controlados de uno a tres años que tengan de quince a treinta días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_30-45Dias\_PostFechRad\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de uno a tres años que tengan de treinta a cuarenta y cinco días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_45-60Dias\_PostFechRad\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de uno a tres años que tengan de cuarenta y cinco a sesenta días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_1-3\_Modalid\_Delict\_Priorizada: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene los Expedientes en Fase Preparatoria Controlados de uno a tres años clasificados por Modalidad Delictiva Priorizada.

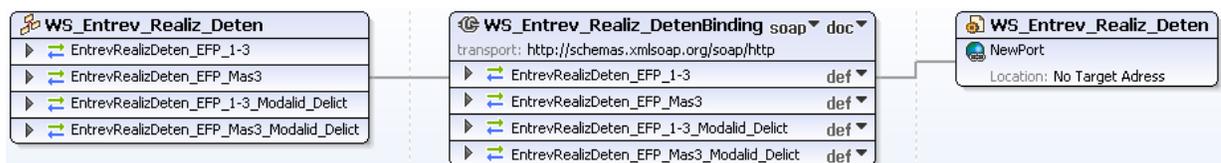


**Imagen 3. Interfaz del Servicio que responde al indicador 3.7 Total de EFP Controlados.**

La interfaz cuenta con 6 operaciones definidas:

- Cantidad\_EFPC\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato numérico que indica la cantidad de Expedientes en Fase Preparatoria Controlados de más de tres años.

- Cantidad\_EFPC\_5-15Dias\_PostFechRad\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de más de tres años que tengan de cinco a quince días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_15-30Dias\_PostFechRad\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de más de tres años que tengan de quince a treinta días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_30-45Dias\_PostFechRad\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de más de tres años que tengan de treinta a cuarenta y cinco días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_45-60Dias\_PostFechRad\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de Expedientes en Fase Preparatoria Controlados de más de tres años que tengan de cuarenta y cinco a sesenta días posteriores a la fecha de radicación.
- Cantidad\_EFPC\_1-3\_Modalid\_Delict\_Priorizada: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene los Expedientes en Fase Preparatoria Controlados de más de tres años clasificados por Modalidad Delictiva Priorizada.



**Imagen 4. Interfaz del Servicio que responde al indicador 3.5.3 Total de Entrevistas Realizadas a Detenidos.**

La interfaz cuenta con 4 operaciones definidas:

- EntrevRealizDeten\_EFP\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de entrevistas

realizadas a detenidos correspondientes a Expedientes en Fase Preparatoria de uno a tres años.

- EntrevRealizDeten\_EFP\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de entrevistas realizadas a detenidos correspondientes a Expedientes en Fase Preparatoria de más de tres años.
- EntrevRealizDeten\_EFP\_1-3\_Modalid\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene la cantidad de entrevistas realizadas a detenidos correspondientes a Expedientes en Fase Preparatoria de uno a tres años clasificados por Modalidad Delictiva Priorizada.
- EntrevRealizDeten\_EFP\_Mas3\_Modalid\_Delictiva: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene la cantidad de entrevistas realizadas a detenidos correspondientes a Expedientes en Fase Preparatoria de más de tres años clasificados por Modalidad Delictiva Priorizada.



**Imagen 5. Interfaz del Servicio que responde al indicador 3.7.3 Total de Acciones de Instrucción en las que participa el Fiscal.**

La interfaz cuenta con 4 operaciones definidas:

- AccionInstruc\_Fiscal\_EFP\_1-3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de acciones de instrucción en las que participa el Fiscal correspondientes a Expedientes en Fase Preparatoria de uno a tres años.
- AccionInstruc\_Fiscal\_EFP\_Mas3: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene el dato que indica la cantidad de acciones

de instrucción en las que participa el Fiscal correspondientes a Expedientes en Fase Preparatoria de más de tres años.

- AccionInstruc\_Fiscal\_EFP\_1-3\_Mod\_Delict: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene la cantidad de acciones de instrucción en las que participa el Fiscal correspondientes a Expedientes en Fase Preparatoria de uno a tres años clasificados por Modalidad Delictiva Priorizada.
- AccionInstruc\_Fiscal\_EFP\_EFP\_Mas3\_Mod\_Delict: Esta operación es de tipo Request-Response, devuelve un documento XML que contiene un dato tabulado o dato tipo tabla que contiene la cantidad de acciones de instrucción en las que participa el Fiscal correspondientes a Expedientes en Fase Preparatoria de más de tres años clasificados por Modalidad Delictiva Priorizada.

Además de las interfaces que se explican anteriormente, se definen los esquemas XML que deben cumplir los datos que pertenecen al Proceso de Justicia Penal y que forman parte del flujo de información que viaja a través de la plataforma de interoperabilidad, así como también se definen los esquemas de los tipos de datos definidos.

## **2.6 Componentes de la Arquitectura de la Plataforma de Interoperabilidad del Órgano de Justicia en Cuba**

### **2.6.1 Portal Corporativo**

La plataforma contará con un portal corporativo a través del cual los funcionarios del Órgano de Justicia-MININT podrán gestionar e interactuar con la misma y realizar las acciones necesarias que contribuyan positivamente en el proceso de toma de decisiones, y éste a la vez contribuya a la eficiencia, agilidad y eficacia del Proceso de Justicia Penal.

El portal permitirá acceder a la plataforma de interoperabilidad a través de peticiones por el protocolo HTTP, que pueden ser realizadas desde cualquier aplicación o dispositivo que soporte dicho protocolo. En general, el portal será el encargado de mostrar toda la información que se encuentra en la plataforma a partir de los permisos o el acceso que se posean dentro de la plataforma. Permitirá además, siempre que el usuario que accede tenga los permisos suficientes, administrar algún componente de la plataforma como el repositorio

de servicios web, el repositorio de esquemas y metadatos, el componente Acaxia o el servidor de aplicaciones.

#### **Requerimientos del Portal Corporativo:**

- **Autenticación:** El portal será la puerta de entrada a la plataforma, es donde se identificarán las aplicaciones y los usuarios para acceder a la información que se encuentre dentro del sistema.
- **Sencillez:** El portal será la primera vista o la vista principal de la plataforma por lo que el entorno debe ser lo más sencillo y amigable posible, pues se ha de tener en cuenta que las personas que tienen acceso a la plataforma son Jueces, Fiscales y demás funcionarios encargados de velar por la justicia en general en el país y que en conjunto conforman la Comisión Nacional del Órgano de Justicia-MININT.

#### **2.6.2 Repositorio de Servicios Web basado en el estándar UDDI**

La plataforma contará con un repositorio de servicios web que estará basado en el estándar UDDI en su versión 3.0. En dicho repositorio serán registrados cada uno de los servicios web y sus interfaces correspondientes a los órganos y organismos que intervienen en el Proceso de Justicia Penal. Existen varias tipologías para la implementación de un registro de UDDI tales como:

- **Registro de Negocios UDDI (UBR):** Es una red de registros UDDI públicos en Internet.
- **E-Mercado UDDI:** Registro UDDI dedicado a un sector de la industria dado.
- **Nodo Privado UDDI:** Registro UDDI sobre una intranet usado para proveer la funcionalidad de UDDI a usuarios de una compañía o empresa.
- **Nodo Extranet UDDI:** Registro UDDI sobre una extranet usado para proveer la funcionalidad de UDDI entre compañías relacionadas en el negocio.

Se pretende el uso de los servicios web en la Plataforma de Interoperabilidad que responderá al Órgano de Justicia-MININT en Cuba, para compartir la información entre los distintos organismos que intervienen en el Proceso de Justicia Penal, dichos organismos, se encuentran dispersos en distintas redes. Por esta razón, se decide optar por una tipología tipo Nodo Extranet UDDI.

El repositorio de servicios web contará con varios procesos, tales como el proceso de Administración, el proceso de Publicación y el proceso de Búsquedas. El proceso de Administración tendrá como objetivo el control y la administración de la información y las funcionalidades del sistema (el repositorio en este caso), por lo que debe existir un Administrador, que será el que debe velar porque se cumpla todo lo que esté establecido. Por otro lado, el proceso de publicación consistirá en el registro de la información, así como de los proveedores de servicios web y las especificaciones técnicas. El proceso de Búsqueda por su parte, permitirá obtener información para el desarrollo de los servicios que brindarán la información a través de la plataforma, así como listar los mismos una vez que se implementen. Además, el repositorio debe contar con un browser o buscador, que no será más que una interfaz a través de la cual se podrá administrar e interactuar con el repositorio a través de sus funcionalidades, en este caso UDDIBrowser. Por otro lado, se define como servidor UDDI a JUDDI, que está desarrollado en el lenguaje Java y por esa razón puede ser utilizado en cualquier plataforma que soporte este lenguaje.

Conceptos asociados al Repositorio de Servicios Web.

- **Usuario:** será aquel usuario que solo podrá acceder a buscar información acerca de los servicios que se encuentran registrados y publicados sin la previa autenticación.
- **Publicador:** será el usuario que tiene los permisos para poder gestionar algunos datos dentro del repositorio.
- **Administrador:** será el usuario que tiene todos los permisos y acceso total para gestionar todos los datos del repositorio.
- **Entidad:** Serán todos y cada uno de los órganos y organismos que intervienen en el proceso de justicia.

### **2.6.2.1 Sistema de Administración del repositorio de servicios web**

La plataforma debe contar con un sistema de administración que permita administrar el repositorio de servicios web. Este, será el mecanismo a través del cual se administrará el repositorio de servicios web, brindando todas las funcionalidades que permitan el proceso de Gestión del repositorio, ya sea agregar o eliminar interfaces al repositorio o modificar las existentes. La vía de acceso será a través de peticiones HTTP desde cualquier dispositivo

que soporte este protocolo, permitiendo acceder a los contratos de los servicios web definidos anteriormente.

### **2.6.3 Repositorio de esquemas y metadatos**

El objetivo de este componente es almacenar todos los esquemas XML que definan el formato de intercambio de los datos que serán intercambiados en la plataforma. Este sistema implementa un procedimiento de registro de esquemas y metadatos que garantiza una guía mediante la cual los organismos interesados en intercambiar datos, están obligados a cumplir una serie de requerimientos que contribuyen al registro de esquemas válidos y bien formados desde el punto de vista técnico.

El repositorio permite acceder a los esquemas del formato de intercambio mediante peticiones HTTP, que pueden realizarse desde cualquier aplicación que implemente el protocolo. En este repositorio estarán registrados todos los esquemas de la tipología de datos estructurales definidos a partir del procedimiento de estandarización de formato de intercambio de datos en la plataforma de interoperabilidad, contribuyendo a institucionalizar la definición de estándares de intercambio de datos, además de documentos, archivos multimedia, archivos binarios, y cualquier otro objeto digital que se vaya a intercambiar en la plataforma.

Las principales funcionalidades que brinda este componente, son el acceso a esquemas XSD que definen la forma de intercambio de datos, así como al archivo de transformación XSLT asociado que garantiza la visualización por defecto de los datos en formato XHTML. Otras funcionalidades significativas son aquellas que garantizan la gestión y administración de los usuarios y la asignación de niveles de acceso de estos; así como también las funcionalidades que implementan el Procedimiento de Registro de Esquemas y Metadatos.

### **2.6.4 Requerimientos de los sistemas de administración de los repositorios de registros web y de esquemas y metadatos.**

- **Autenticación:** El portal debe permitir la autenticación de usuarios para acceder a los repositorios por parte de los administradores, publicadores, proveedores de servicios y demás usuarios definidos para mantener y administrar dichos repositorios.
- **Acceso restringido:** El portal debe ser capaz de una vez autenticados los usuarios proveerles solamente la información y acceso que les está permitido.

- **Funcional:** Debe brindar todas las funcionalidades del Proceso de Gestión de los repositorios posibles a los usuarios una vez que se autenticuen para facilitar su trabajo.

## 2.6.5 Servidor de Aplicaciones

Se utiliza como servidor de aplicaciones JBoss, el mismo permite la integración con apache que es el servidor web que más abunda entre las aplicaciones con que cuenta el Órgano de Justicia-MININT. JBoss cuenta con JBssPortal, que es una plataforma para servir y albergar portales web y soporta varios estándares como JMX<sup>10</sup>, además de contar con otras funciones internas para manejar la mensajería, persistencia, manejo de log, entre otras.

### 2.6.5.1 Configuración del servidor

El servidor JBoss puede ser configurado de tres maneras diferentes, configuración *minimal*, configuración *default* y configuración *all*. La configuración *minimal* tiene el mínimo de servicios para levantar JBoss, en este caso el servicio de *log*, el server JNDI y el URL deployment scanner. Generalmente, esta configuración se utiliza para iniciar MX/JBoss sin ninguna otra tecnología Java EE, siendo el server base debido a que no está presente ningún web container ni existe soporte para EJB o JMS.

La configuración *all* como su nombre lo indica, inicia todas las operaciones y servicios disponibles incluyendo RMI/IIOP, y los servicios de clúster que no son cargados en las demás configuraciones. Por último, la configuración *default* es un perfil básico de servidor Java EE que contiene un conjunto predeterminado de servicios. Cuenta con los servicios que más se utilizan y que son necesarios para implementa una aplicación Java EE y como se menciona anteriormente, no incluye los servicios de clúster (clustering services).

La configuración que se utilizará en la plataforma de interoperabilidad será la configuración *default*, pues el caso de *all* implica que se estarán levantando servicios que no serán utilizados, entorpeciendo el rendimiento del servidor en general, por otro lado, la configuración *minimal* carece de muchos servicios que son indispensables como son el caso de JMS y los web container.

---

<sup>10</sup> Java management eXtensions

### 2.6.5.2 Persistencia del servidor en producción

El servidor JBoss implementa la persistencia por defecto basada en JPA<sup>11</sup>. Este modelo de persistencia es recomendable para entornos de desarrollo pero no para entornos de producción. La razón radica en que durante el desarrollo de aplicaciones, el despliegue (*deploy*) de éstas en el servidor configurado para persistencia interna, contribuye a la eficiencia en la compilación y funcionamiento de las aplicaciones. Sin embargo, en un entorno de producción no se recomienda usar la configuración de persistencia interna porque reduce la capacidad de adaptación del servidor, además de que se corre el riesgo de alcanzar grandes índices de estrés producto a la convergencia de peticiones. De manera que se sugiere el uso de bases de datos externas desplegadas en Sistemas Gestores de Bases de Datos (DBMS) a los que se le pueda delegar la responsabilidad de optimizar y gestionar las consultas.

En este sentido, se propone la definición de tantos *datasource* como servidores de bases de datos se necesiten para externalizar la persistencia del servidor de aplicación. Esto significa que todas las tablas en las que se almacenan todos los datos del servidor de aplicación, se desplegarán en servidores de bases de datos externos, mejorando el rendimiento a partir del empleo de los mecanismos de optimización de consultas de los propios servidores.

En el fichero *hsqldb-ds.xml* del directorio *deploy* del servidor, se definen los *datasource* a emplear, y en cada uno se inicializan algunos parámetros como son:

- Parámetros de conexión: usuario y contraseña del servidor de bases de datos.
- Mínimo (*min-pool-size*) y máximo (*max-pool-size*) de conexiones en cada *pool*
- El tiempo máximo de espera para el establecimiento de una conexión (*idle-timeout-minutes*)

Una vez que se define los *datasources* de las bases de datos a donde se desplegarán las tablas del JBoss, se configuran los *Java Beans* de administración de conexiones en el fichero de forma que se les especifique que los *datasources* que utilizaran para configurar la persistencia no serán los propios internos, sino que se les debe dar la dirección de los *datasources* que se han definido externamente.

---

<sup>11</sup> Java Persistence API, es la API persistencia que fue desarrollada por Java EE. Es un Framework del lenguaje de programación Java que se encarga de manejar los datos relacionales en las aplicaciones.

## 2.7 Seguridad en la Plataforma

La seguridad de la información es un tema delicado que no está al alcance de todos, en ocasiones, no se le da la importancia que necesita dicha información y no se diferencia lo privado de lo público, no por cuestiones de conocimiento, sino por la propia ignorancia.

La seguridad, es un requisito indispensable cuando se desea garantizar que la comunicación entre las aplicaciones, los servicios web, el flujo de información y demás componentes se realice de forma segura y confiable tanto como sea posible. Para garantizar o al menos tener esta posibilidad es necesario contar con al menos alguno de los siguientes requisitos:

- **Confidencialidad:** significa que la información no sea visible a extraños.
- **Autenticación:** la autenticación garantiza que el acceso a la información solo será por quienes posean la identidad adecuada para acceder al sistema.
- **Integridad:** permite estar seguros de que la información no ha sido modificada, ya sea de forma accidental o no.
- **No repudio:** con esto se asegura que el que envíe el mensaje no pueda negar haberlo enviado.

### 2.7.1 Elementos de seguridad del intercambio de datos y el acceso a los sistemas de la plataforma

#### 2.7.1.1 Estándares que aportan seguridad a la plataforma

Existen varias formas de atribuirle seguridad a los procesos de intercambio de datos y las transacciones entre los servicios web. Entre ellos se encuentra Web Service Security, que es un mecanismo que permite el uso de tecnologías de encriptación como son SSL y las PKI o infraestructuras de llaves públicas, esto permite que se le pueda brindar seguridad en general a la mensajería entre los servicios web con que contará la plataforma de interoperabilidad. También se encuentra XML-Encryption que es un lenguaje que permite la encriptación de documentos y datos XML y que además posee varios niveles de encriptación, que van desde un elemento hasta un documento completo. Con esta tecnología se le puede brindar un poco más de seguridad a el proceso de intercambio de datos puesto que los mismos pueden ser cifrados y aportan más calidad y confiabilidad en el origen de los mismos.

Además se cuenta con el estándar XML-encryption que dota de mecanismos para la generación de firmas digitales basadas en XML, teniendo en cuenta que a través de esta

variante se puede fortalecer aun más la seguridad en la plataforma, pues se gana en aspectos como la autenticidad e integridad de los datos. Otro de los estándares a tener en cuenta es SAML para fortalecer los procesos de autenticación en la plataforma. Como característica y ventaja que tienen en común estos estándares se encuentra que todos son estándares abiertos desarrollados por organizaciones como la W3C y Oasis, además de ser independientes de la plataforma.

### **2.7.1.2 Acaxia como componente de seguridad centralizada**

Los estándares mencionados anteriormente, proveen seguridad a las transacciones de los servicios web y combinándolas junto a otros estándares se logra una potente seguridad de forma general.

Pero no basta con estas soluciones para proveer la completa seguridad a la plataforma, pues quedan cuestiones como la autenticación y la autorización. La Universidad de las Ciencias Informáticas ha desarrollado un componente llamado Acaxia que gestiona la seguridad de forma centralizada y que concibe factores como multi-entidad y multi-sistema, cuestiones que casi ninguna otra solución incorpora, lo que la hace una solución bastante robusta.

Acaxia será el componente encargado de verificar los permisos que poseen los usuarios que se autentican en la plataforma a través de certificados digitales que estos deben poseer, así como también velará por la seguridad en el repositorio de esquemas y metadatos, el registro UDDI y el servidor de aplicaciones JBoss, pues, una vez que se introduce el usuario y contraseña para interactuar con la plataforma, Acaxia es el componente que estará presente en cada acción que se realice por parte de estos usuarios.

Hay que tener en cuenta que esta solución propuesta por la UCI posee o simula un registro UDDI interno que no es lo suficientemente robusto como para su uso en la plataforma. El problema radica en que Acaxia no se integra con un registro UDDI que no sea el que trae por defecto, siendo este el principal obstáculo a la hora de decidirse por esta herramienta. Sin embargo, el problema puede ser resuelto mediante la implementación de algún mecanismo que permita la integración de Acaxia con un registro UDDI.

## **2.7.2 Elementos de seguridad del entorno de despliegue de los componentes de la plataforma**

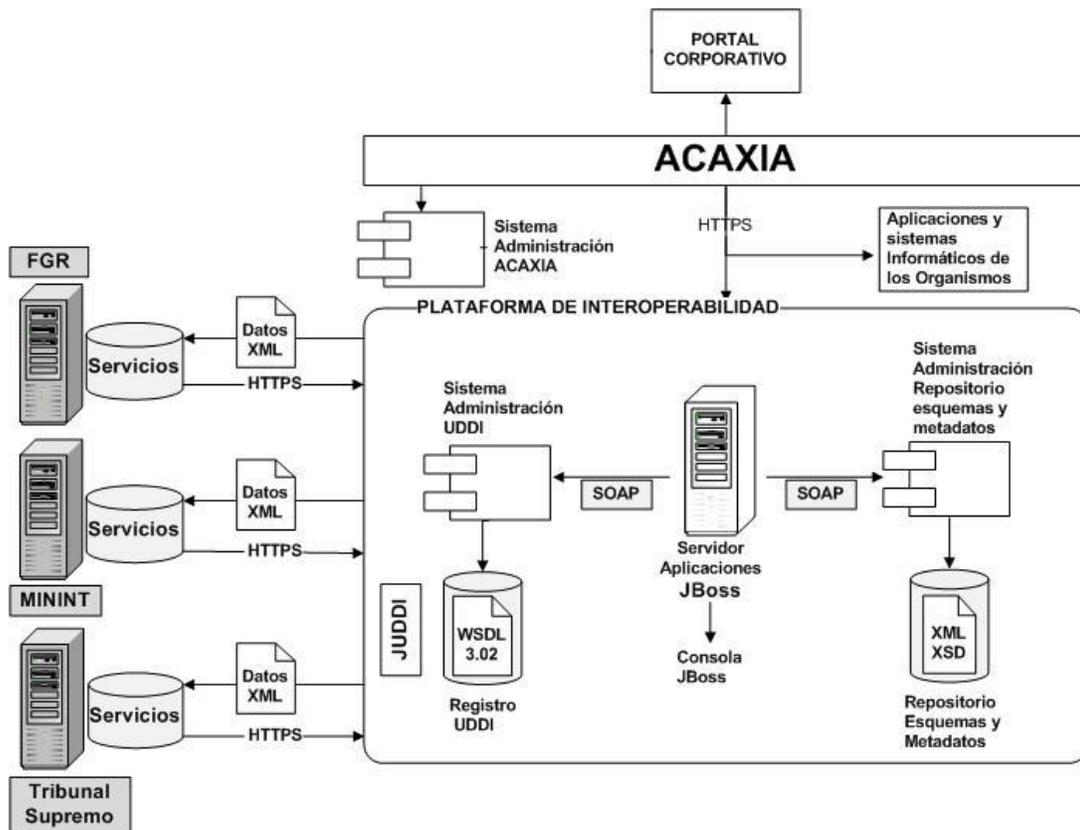
### **2.7.2.1 Firewall como forma protección de los servidores**

Además de los componentes de seguridad mencionados, se necesita la protección de todos los servidores y demás ordenadores que brindan servicio a la plataforma. Una de las formas para lograr esto es a través de un firewall o cortafuegos, que no es más que una parte de la red que a través de uno o varios dispositivos se encarga de bloquear el acceso no autorizado y permitir solamente entrar a quien está autorizado, utilizando software, hardware o la combinación de ambos. En el ámbito de la seguridad informática son un punto clave, pues todo lo que entra, sale o pasa a través de los sistemas y redes a los que pertenecen, son examinados por los cortafuegos a través de las reglas y políticas que se definan.

Iptables es un firewall que es fácil de configuración dependiendo de lo que se quiera lograr, consiste en configurar un script Shell añadiendo las reglas que se quieren establecer aunque existen herramientas que permiten realizar la configuración de forma gráfica (Firestarter, IpKungfu, entre otros) haciendo posible la configuración de las reglas sin necesidad de tener conocimientos profundos del tema. Se decide incorporar esta poderosa herramienta con vistas a fortalecer más la seguridad en la plataforma de interoperabilidad, de manera que los servidores no solo estén protegidos por Acaxia y los estándares definidos, sino que el acceso a los mismos esté monitorizado por Iptables.

## **2.8 Vista de despliegue de los componentes de la arquitectura de interoperabilidad**

Con la definición de los estándares analizados, así como las tecnologías y herramientas que permiten llegar a lo que se conoce como interoperabilidad, se propone la siguiente vista de despliegue, que soluciona el problema de la interacción entre los sistemas informáticos con que cuenta actualmente el Órgano de Justicia-MININT en Cuba, sistemas que son la fuente de datos para el proceso de toma de decisiones que influye directa y positivamente en el proceso de Justicia Penal.



**Imagen 7** Vista de despliegue de los componentes de la arquitectura de interoperabilidad.

### CONCLUSIONES DEL CAPÍTULO:

En el presente capítulo se elaboraron los contratos de los servicios web que brindarán información al proceso de Justicia Penal, así como se definieron los requerimientos que debe cumplir la Plataforma de interoperabilidad. Se describió la arquitectura de interoperabilidad del Órgano de justicia-MININT, donde se explican las tecnologías y estándares seleccionados para el desarrollo de la misma y finalmente se expone la propuesta de solución al problema de la investigación.

## Capítulo 3: Análisis y validación de la arquitectura

### INTRODUCCIÓN

La calidad de la plataforma de interoperabilidad depende directamente de la evaluación de la arquitectura, pues en este caso, es determinante para la selección entre las diferentes tecnologías que existen para dar solución al problema de la investigación. Existen varias técnicas y métodos para evaluar una arquitectura, generalmente se enfocan en el análisis y prueba de los atributos de calidad o los casos de uso arquitectónicamente significativos. Para realizar la evaluación de la arquitectura propuesta se tendrán en cuenta varios métodos y técnicas que permiten validar el diseño contra las cualidades que se desean tener en la arquitectura, enfocándose en la evaluación cualitativa de la misma.

#### 3.1 Objetivos.

Los objetivos que se persiguen con la evaluación de la arquitectura, son principalmente mejorar o corregir el diseño la arquitectura en caso de detectarse algún error, fundamentar las decisiones arquitectónicas que se han tomado y aplicado, así como validar los requerimientos definidos. En general, se pretende medir y evaluar cualitativamente el potencial de la arquitectura diseñada para alcanzar los atributos de calidad que requiere y demanda la Plataforma de Interoperabilidad del Órgano de Justicia-MININT cubano.

#### 3.2 Necesidad de evaluar una arquitectura

La arquitectura es como la columna vertebral en las personas para cualquier proyecto o sistema informático, su evaluación y análisis permitirán arrojar errores que pueden resultar incómodos y sobre todo costosos en el futuro en cuanto a tiempo y recursos se refiere. También será posible verificar el cumplimiento de los requisitos previamente definidos con que debe cumplir el diseño de la arquitectura, y los atributos de calidad deseados para el sistema o la plataforma en este caso.

En caso de que se detecten errores en la arquitectura, es más eficiente solucionarlos antes de que se comience a desarrollar el sistema, incidiendo positivamente en factores como tiempo, recursos, planificación, alcance y presupuesto entre otros que son más engorrosos de solucionar una vez desarrollado el sistema.

### **3.3 Resultados que se obtienen a través de la evaluación de la arquitectura**

Generalmente, cuando se realiza la evaluación de una arquitectura se les da respuesta a varias interrogantes que se fueron formulando a medida que se trabaja en la propuesta de la misma. Ejemplos de estas preguntas son:

- ¿Es adecuada para el sistema la arquitectura que se diseñó?
- ¿Realmente cumple con los requerimientos o requisitos definidos?

Evidentemente, estos resultados varían según el método o la técnica que se utilice para la evaluación. Una arquitectura resulta adecuada cuando en general cumple con los requisitos de calidad, cuando es posible llevar a cabo el desarrollo del sistema con los recursos disponibles y sobre todo, si cumple con los requerimientos previamente definidos.

Por lo general, los resultados del análisis y evaluación de la arquitectura no son medibles cuantitativamente, sino cualitativamente, identificando las principales debilidades y acto seguido, proponiendo las soluciones a las mismas.

### **3.4 ¿Cuándo evaluar la arquitectura?**

La evaluación de la arquitectura se puede realizar en el momento que se estime conveniente, pero varios autores hacen referencia a dos etapas distintas, una etapa temprana y una tarde. La etapa temprana es cuando para realizar la evaluación no es necesario que la arquitectura esté completamente terminada o implementada. La etapa tarde consiste en realizar la evaluación una vez que la arquitectura ya se encuentra establecida e implementada. Generalmente, para determinar o definir el momento en que se va a realizar la evaluación se debe tener en cuenta si el equipo de desarrollo comienza a tomar decisiones que afectan directa o indirectamente a la arquitectura del sistema. Por otro lado, es sumamente importante determinar si el costo de no tomar las decisiones por parte del equipo de desarrollo sería mayor que el costo de realizar una evaluación.

### **3.5 Algunos métodos que permiten evaluar arquitectura**

Existen varios métodos en la actualidad que permiten la realización de evaluaciones arquitectónicas, todos y cada uno de ellos enfocados en distintos aspectos y atributos de calidad, incluso hay algunos que dependiendo de las condiciones y los atributos definidos evalúan mejor y más eficiente una arquitectura que otros.

**SAAM (Software Architecture Analysis Method):** Por sus siglas en ingles, fue el primer método que surgió basado en escenarios, que originalmente fue creado para el análisis del grado de modificación de la arquitectura. El mismo consiste en la enumeración de escenarios que representarán los probables cambios a los que el sistema estará sometido en el futuro. La evaluación del método está definida en 6 pasos:

1. Desarrollo de escenarios.
2. Descripción de la arquitectura.
3. Clasificación y asignación de la prioridad de los escenarios.
4. Evaluación individual de los escenarios indirectos.
5. Evaluación de la interacción de los escenarios.
6. Creación de la evaluación global.

**ATAM (Architecture Trade-off Analysis Method):** El método de análisis de acuerdos de arquitectura se encarga de revelar la forma en la arquitectura satisface ciertos atributos de calidad y provee una visión de cómo estos atributos interactúan con otros. Este método se centra en 3 áreas, los estilos arquitectónicos, el análisis de atributos de calidad y el método de evaluación SAAM. Consta con 9 pasos para realizar la evaluación que están divididos y agrupados en nueve fases (Presentación, Investigación y análisis, Pruebas y Reportes).

**ARID (Active Reviews for Intermediate Design):** Es un método de bajo costo y gran beneficio, el mismo es conveniente para realizar la evaluación de diseños parciales en las etapas tempranas del desarrollo. ARID es un híbrido entre ADR y ATAM. Se basa en ensamblar el diseño de los stakeholders para articular los escenarios de usos importantes o significativos, y probar el diseño para ver si satisface los escenarios. Como resultado de la aplicación de dicho procedimiento se obtiene un diseño de alta fidelidad acompañado de una alta familiarización con el diseño de los stakeholders.

**Losavio (2003):** Es un método para evaluar y comparar arquitecturas de software candidatas, que hace uso del modelo de especificación de atributos e calidad adaptado del modelo ISO/IEC 9126. La especificación de los atributos de calidad haciendo uso de un modelo basado en estándares internacionales ofrece una vista amplia y global de los atributos de

calidad, tanto a usuarios como arquitectos del sistema, para efectos de la evaluación. El método contempla siete actividades.

### **3.6 Definición de los atributos de calidad a evaluar.**

Teniendo en cuenta que la calidad del software es la medida en la que se logra la combinación deseada de atributos en los procesos (interoperabilidad, integridad, confiabilidad, etc.) del software según la IEEE, y que muchos de estos son responsabilidad de la arquitectura, se definen los siguientes como objetivos inmediatos de la primera fase de la arquitectura de la Plataforma de Interoperabilidad del Órgano de Justicia-MININT.

**Seguridad:** Se puede entender por seguridad la capacidad de que un sistema pueda mantener la información libre de riesgos, para lo cual es necesario que cuente con aspectos como la integridad, confidencialidad, disponibilidad y no repudio.

**Flexibilidad:** Se puede entender como la capacidad de que un sistema pueda adaptarse a los cambios, diferentes entornos y condiciones. Se puede decir que existe la flexibilidad en un sistema, cuando este puede ser fácilmente modificado para dar respuesta a nuevos requerimientos.

**Escalabilidad:** Es la facilidad con que un sistema puede modificarse para aumentar su capacidad funcional, el diseño arquitectónico, de datos o procedimental.

**Integridad:** Se puede entender como la medida o capacidad en la que componentes del sistema desarrollados por separados, pueden trabajar correctamente una vez que se integran.

**Reusabilidad:** Es la capacidad de que los componentes y estructura de un sistema puedan ser reutilizados en o por aplicaciones futuras.

### **3.7 Evaluación de la arquitectura de Interoperabilidad**

No es objetivo de la presente investigación realizar una comparación de los métodos y técnicas de evaluación que existen actualmente en una escala de mejor o peor. Teniendo en cuenta que la arquitectura es un producto temprano del diseño de un sistema y que esta, es el soporte principal del mismo, se decide utilizar ARID como método para realizar la evaluación de la arquitectura, pues el mismo permite realizar evaluaciones a diseños parciales en etapas tempranas a un costo muy bajo. Además aprovecha las mejores características de otros métodos puesto que es un combinado entre ATAM y ADR.

Para la evaluación se seleccionan los principales escenarios para la arquitectura, aunque existen muchos más que también aportan cualidades a la arquitectura, pero que serán refinados en etapas posteriores. A continuación se describen y explican las relaciones de los atributos de calidad definidos con los escenarios seleccionados.

**Tabla 2.** Relación entre el atributo Seguridad y el escenario de Interacción con la plataforma.

Atributo de calidad	Perfil	Escenario
Seguridad	Seguridad externa	Interacción con la plataforma
<b>Relación Atributo-Escenario</b>		
<p>Acaxia, es el componente que controla la seguridad de forma general cuando se comienza a interactuar con la plataforma. Es el encargado de cuando un usuario o sistema informático quiere acceder, tomar sus credenciales (usuario, contraseña, certificado digital) una vez autenticados y verificar los permisos que poseen, a partir de ahí, todas las acciones sucederán solo si posee los permisos para realizarlas. Además, se cuenta con estándares como WS Security que se encarga de la seguridad en las transacciones de los servicios web con que cuenta la plataforma, SSL (Secure Socket Layer) es el protocolo de comunicación con la plataforma, haciendo que todas las conexiones basadas en el estándar HTTP sean seguras y confiables.</p>		

**Tabla 3.** Relación entre el atributo Seguridad y el escenario de Acciones en la plataforma.

Atributo de calidad	Perfil	Escenario
Seguridad	Seguridad interna	Acciones en la plataforma
<b>Relación Atributo-Escenario</b>		
<p>Todas las acciones en la plataforma serán controladas inicialmente por el componente Acaxia y las conexiones internas en la plataforma serán basadas en SSL de forma que sean seguras. En el directorio conf/ login-config.xml del servidor de aplicaciones JBoss se puede configurar una lista de nombres de dominios de seguridad, además de que se puede personalizar la seguridad en cuanto a la integración con sistemas de autenticación y bases de datos existentes. Por otro lado, se cuenta con los estándares SAML, XML Digital Signature, WS-Security y XML Encryption que fortalecen la seguridad en cuanto a la</p>		

encriptación de los datos, las transacciones entre servicios web y los protocolos de comunicación entre los mismos, brindando integridad, confiabilidad y no repudio de los datos en general.

**Tabla 3.** Relación entre el atributo Flexibilidad y el escenario de Ajuste a los cambios.

Atributo de calidad	Perfil	Escenario
Flexibilidad	Flexibilidad	Ajuste a los cambios
<b>Relación Atributo-Escenario</b>		
<p>Debido a que toda la arquitectura está basada en estándares abiertos, posibilita que estos puedan ser modificados sin necesidad de realizar cambios que afecten la estructura general de la arquitectura. Por ejemplo, si se decide en el futuro redefinir el lenguaje de intercambio de datos, solo sería necesario definir los esquemas de los datos en el nuevo lenguaje, lo que no afectará a los demás componentes de la arquitectura. Lo mismo sucede en el caso del servidor de aplicaciones, pues solo sería necesario establecer las configuraciones del nuevo servidor.</p>		

**Tabla 4.** Relación entre el atributo Escalabilidad y el escenario de Diseño de la arquitectura.

Atributo de calidad	Perfil	Escenario
Escalabilidad	Ampliación	Diseño de la arquitectura
<b>Relación Atributo-Escenario</b>		
<p>Gracias a que se decide utilizar los estándares abiertos, se garantiza que la arquitectura es escalable en su totalidad, posibilitando la evolución de su estructura. Se pueden agregar nuevos componentes sin necesidad de realizar muchos cambios en los existentes, incluso, en una segunda fase de la plataforma, se puede orientar la arquitectura actual a una arquitectura orientada a servicios.</p>		

**Tabla 5.** Relación entre el atributo Integridad y el escenario de Integración entre componentes.

Atributo de calidad	Perfil	Escenario
Integridad	Integridad	Integración entre componentes
Relación Atributo-Escenario		
<p>La integración entre los componentes definidos en la arquitectura está garantizada por el hecho de que todos están basados en estándares ampliamente aceptados mundialmente, además de ser de libre uso y contar con comunidades de desarrollo como soporte. Por otro lado, los componentes con que cuenta la arquitectura son configurables respecto a la integración, por ejemplo, el servidor de aplicaciones JBoss posee la facilidad de integrarse con Apache, ampliando sus funcionalidades en cuanto a rendimiento se refiere. Sin embargo, el registro UDDI interno que implementa el componente Acaxia, no es lo suficientemente robusto y se hace necesaria la integración con un registro UDDI externo a dicho componente.</p>		

**Tabla 6.** Relación entre el atributo Reusabilidad y el escenario de Intercambio de datos.

Atributo de calidad	Perfil	Escenario
Reusabilidad	Reusabilidad	Intercambio de datos
Relación Atributo-Escenario		
<p>El intercambio de datos a través de la plataforma se encuentra estandarizado y definido a través de XML. Gracias a esto es posible que los datos puedan ser utilizados por todos sin necesidad de transformaciones, además, se permite que sistemas o aplicaciones informáticas que se incorporen en el futuro a la plataforma utilicen el mismo formato de datos que ya estaba definido. Por otro lado, los servicios web que se brindan en lo interno de la plataforma, al estar implementados a partir de las interfaces WSDL definidas, permiten que los mismos sean independientes al lenguaje y plataforma facilitando que sean accesibles y utilizados por todos sin dificultades.</p>		

Como resultado de la evaluación cualitativa de la arquitectura de interoperabilidad propuesta, se evidenció que se gestiona la seguridad eficientemente, tanto en lo externo como en lo interno de la plataforma, pues, se logra a través de Acaxia, el protocolo SSL y los estándares XML Encriptyon, SAML, WS-Security y XML Digital Signature. Además, se demostró que la arquitectura es completamente escalable y flexible respecto a los cambios en su estructura debido a que está basada en su totalidad en estándares abiertos.

Por otro lado, se comprueba la reusabilidad de la estandarización del lenguaje de intercambio de datos, pues XML es el estándar por excelencia para el intercambio de datos a nivel mundial. En general, la arquitectura proporciona satisfactoriamente los atributos de calidad definidos. Se debe tener en cuenta, que en el éxito de la propuesta, influyen factores ajenos a la arquitectura que se ha propuesto en el presente trabajo, tales como el hardware y las condiciones bajo las que se implemente la solución.

#### **CONCLUSIONES DEL CAPÍTULO:**

En el presente capítulo se expusieron varias de las técnicas utilizadas para evaluar una arquitectura. Se definieron los atributos de calidad a evaluar y se realizó la evaluación del diseño de la arquitectura de Interoperabilidad del Órgano de Justicia-MININT de manera parcial, demostrando que la arquitectura condiciona en gran medida los atributos de calidad deseados.

## CONCLUSIONES GENERALES

A partir del acuerdo por parte de los funcionarios del Órgano de Justicia-MININT y la Universidad de las Ciencias Informáticas (UCI) de crear un grupo de trabajo encargado del desarrollo de una plataforma informática que posibilite la interacción de los sistemas informáticos para lograr un intercambio de datos surge la necesidad de proponer una arquitectura de interoperabilidad para dicha plataforma lo más confiable, flexible y escalable posible. Con vistas a lograr esto tuvo especial significación el estudio del estado del arte de algunas de las soluciones existentes en el mundo, resultando los siguientes conceptos como los más importantes:

- Interoperabilidad.
- Niveles de interoperabilidad.
- Arquitectura de interoperabilidad.
- Herramientas y estándares que posibilitan el desarrollo de una arquitectura de interoperabilidad.

La representación de la arquitectura de interoperabilidad permitió llegar a un entendimiento entre todos los involucrados en la tarea de desarrollarla Plataforma de interoperabilidad para el Órgano de Justicia-MININT, determinado entre otros factores por las siguientes decisiones:

- Se decide el uso de estándares abiertos para el desarrollo de la plataforma.
- Se selecciona como lenguaje de estandarización e intercambio de documentos XML.
- Se selecciona como servidor de aplicaciones JBoss.
- Se selecciona como estándar para el repositorio de servicios web UDDI.

A raíz del estudio realizado se demostró que el diseño de la arquitectura de interoperabilidad cumple con los objetivos propuestos, mediante la evaluación de los atributos de calidad más deseados en los principales escenarios definidos a través del método ARID.

## RECOMENDACIONES

- Desplegar los componentes de la arquitectura en un entorno piloto para implementar pruebas de intercambio de datos a partir de los esquemas definidos en la primera etapa de desarrollo de la plataforma de interoperabilidad.
- Implementar integración del servidor UDDI con el componente de seguridad ACAXIA
- Implementar el portal corporativo e integrarlo con el resto de los componentes de la arquitectura propuesta.
- Implementar los servicios web correspondientes a los datos seleccionados en la primera etapa del desarrollo de la plataforma de interoperabilidad, a partir de los contratos WSDL definidos en el presente trabajo.
- Proponer un método que permita la evaluación de una arquitectura de interoperabilidad.
- Evolucionar la arquitectura actual a una Arquitectura Orientada a Servicios SOA.

## Bibliografía

**ALCTS.** [En línea] <http://www.libraries.psu.edu/tas/jca/ccda/tf-meta3.html>.

**AMELLER, David y FRANCH, Xavier.** *Asignación de Tratamientos a Responsabilidades en el contexto del Diseño Arquitectónico Dirigido por Modelos*. Departamento de Lenguajes y Sistemas Informáticos, Universidad Politècnica de Catalunya. Barcelona : s.n. pág. 10.

**AMENDOLA, L. 2003.** *La Confiabilidad Desde el Diseño. Proyectos de Mantenimiento*. Departamento de Proyectos de Ingeniería, PMM Institute for Learning. Universidad Politécnica de Valencia. España : Asociación Española de Mantenimiento, 2003. pág. 10, Artículo.

**Angles, Renzo, y otros. 2009.** *Documentación electrónica e interoperabilidad de la información*. [ed.] Sergio Ochoa, Cecilia Bastarrica y Claudio Gutiérrez. Primera Edición. Santiago de Chile : Maval Ltda, 2009. pág. 220. La Url de esta tesis es

[http://www.aem.gob.cl/archivos/Documentacion\\_Electronica\\_e\\_Interoperabilidad\\_de\\_la\\_Informacion\\_-\\_Libro\\_2009.pdf](http://www.aem.gob.cl/archivos/Documentacion_Electronica_e_Interoperabilidad_de_la_Informacion_-_Libro_2009.pdf) .

**Ayala, Ramón Montero.** XML inicios y referencias.

—. *XML inicios y referencias*. pág. 274.

**Baena, Cecilia García. 2010.** *Cloud Computing*. s.l. : IBM, 2010. pág. 4, Informa de prensa.

**BARBACCI, Mario. 2005.** IEEE Argentina. *IEEE*. [En línea] IEEE, 22 de Julio de 2005. [Citado el: 9 de Febrero de 2008.] <http://www.ieee.org.ar/noticiasdetalle.asp?IDNoticia=115>. CS-08.

**Bastarrica, Maria Cecilia, Gutiérrez, Claudio y Ochoa, Sergio. 2009.** *Documentación electrónica e interoperabilidad de la información*. 2009. pág. 220.

**BOHORQUEZ, José Yván. 2008.** Factorías de Software. Implementación de una Factoría en la empresa E4GS. [entrev.] Jorge Luis VALDÉS GONZÁLEZ. *Factorías de Software*. Los Cortijos, 29 de Abril de 2008.

**2006.** Boletín Trimestral del instituto Tecnológico de Informática. . [En línea] Noviembre de 2006. <http://www.iti.es>.

**BOOCH, Grady. 1998.** *Object-Oriented Analysis and Design whith applications*. Segunda Edición. Santa Clara : Addison-Wesley, 1998. 0-8053-5340-2.

**BOOCH, GRADY y RUMBAUGH, JAMES. 1995.** *Unified Method for Object-Oriented Development*. s.l. : Rational Software Corporation, 1995. Overview.

**Botello, Alejandro. 2005.** *Arquitectura Orientada a Servicios (SOA)*. s.l. : Universidad Politécnica de San Luis Potosí, 2005. pág. 18, Exposición. Informa presentado por Alejandro Botello el 25 de mayo de 2005 en el 1er Simposio de Tecnologías de la Información y Telemática.

**CABRERO, Daniel, GARZÁS, Javier y PIATTINI, Mario. 2008.** *Técnica de Mejora del Mantenimiento Software Basada en Valor*. España : s.n., 2008.

**Canarias., Consejería de Justicia y Seguridad del Gobierno de.** *Proyectos de Interoperabilidad en la Administración de Justicia en Canarias*.

**CANÓS, José H., LETELIER, Patricio y PENADÉS, Carmen. 2003.** *Metodologías Ágiles en el Desarrollo de Software*. Universidad Politécnica de Valencia (DSIC). España : Ingeniería del Software y Sistemas de Información (ISSI), 2003. Informe. Artículo perteneciente al Informe del Taller Internacional sobre las Metodologías Ágiles en el Desarrollo de Software..

**CARLES AMBROJO, Joan. 2005.** Las fábricas de 'software' buscan especialización y costes laborales más bajos. *El País*. [En línea] 12 de Mayo de 2005. [Citado el: 4 de Abril de 2008.] [http://www.elpais.com/articulo/tecnologia/fabricas/software/buscan/especializacion/costes/laborales/bajos/elpcibtec/20050512elpcibtec\\_3/Tes](http://www.elpais.com/articulo/tecnologia/fabricas/software/buscan/especializacion/costes/laborales/bajos/elpcibtec/20050512elpcibtec_3/Tes).

**Castells, Pablo. 2005.** *Web Semántica*. 2005.

**Cornejo., Jose E. Gonzales.** [En línea] [Citado el: 10 de 12 de 2010.]

**Criado, J. Ignacio, Gascó, Mila y Jiménez, Carlos E. 2010.** *Bases para una estrategia Iberoamericana de Interoperabilidad*. 2010. pág. 55.

**De la Rosa Triviño, José Luis, Triviño Cabrera, Alicia y Aldana Montes, José F.** *Servicios Web Semánticos: OWL-S y WSMO*. Málaga : Escuela Técnica Superior de Ingeniería Informática. pág. 12. Esto está en D:\Hogguit\Para tesis\Tesis\Documentacion Tesis\Elieta bibliografia\Bibliografía nueva\Semantic Web Services (SWS)\Papers\ponencia76\_2.pdf.

**Diario El País. 2008.** Ciberpaís. *El País*. [En línea] 4 de Abril de 2008. [Citado el: 4 de Abril de 2008.] <http://www.elpais.com>.

**Empresa de Software © DAEDALUS. 2007.** DAEDALUS. [En línea] Data, Decisions and Language, S. A., 2007. [Citado el: 8 de Enero de 2008.] <http://www.daedalus.es/AreasISDiseno-E.php>.

**FENTON, Norman E. y NEIL, Martin. 2005.** *Software Metrics: Roadmap*. London : s.n., 2005.

**FENTON, Norman E. y PFLEEGER, Shari Lawrence. 1997.** *Software metrics : a rigorous approach*. 2nd. London : Boston : PWS Pub., 1997. 0534954251.

**GALVÁN, Blas J., y otros. 2005.** *Técnicas de análisis de fiabilidad y métodos evolutivos de optimización global aplicados al diseño de software*. Instituto de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería, Universidad de Las Palmas de Gran Canaria. Islas Canarias : s.n., 2005. pág. 17, Artículo de Investigación.

**GAMMA, Erich, y otros. 1998.** *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l. : Addison Wesley Longman, 1998. 0-201-63498-8.

**GARCÍA PEÑALVO, Francisco José, y otros. 2004.** *Modelo de Cualificación de Assets del repositorio GIRO*. Valladolid : s.n., 2004.

**GARCIA, Joaquín. 2006.** Gestión de proyectos con SCRUM. *IngenieroSoftware*. [En línea] 4 de Septiembre de 2006. [Citado el: 18 de Marzo de 2008.] <http://www.ingenierosoftware.com/equipos/scrump.php>.

**GONZÁLEZ ARECHAVALA, Yolanda y GARCÍA, Fernando de Cuadra. 2001.** *Calidad del Software*. Instituto de Investigación, Escuela Técnica Superior de Ingeniería (ICAI). s.l. : País Vasco, 2001. pág. 54.

**GRACIA, Joaquín. 2004.** Podredumbre del software. *IngenieroSoftware*. [En línea] 22 de Agosto de 2004. [Citado el: 18 de Marzo de 2008.] <http://www.ingenierosoftware.com/analisisydiseno/podredumbre.php>.

**GREENFIELD, Jack y SHORT, Keith. 2003.** *Software Factories. Assembling Applications with Patterns, Models, Frameworks and Tools*. Microsoft. California, USA : s.n., 2003. 13.

**HUMPHREY, WATTS S. 2001.** *Introducción al Proceso de Software Personal*. [ed.] Andrés Otero. Primera Edición. Madrid : PEARSON EDUCATION S.A., 2001. pág. 328. Edición en Español. 84-7829-052-4.

- IEEE. 2004.** *Guide to the Software Engineering Body of Knowledge*. s.l. : IEEE Computer Society, 2004. 0-7695-2330-7.
- . **1990.** *IEEE Standard Glossary of Software Engineering Terminology*. Institute of Electrical and Electronics Engineers (IEEE). s.l. : IEEE, 1990. pág. 84. 155937067X.
- . **2006.** IEEE, The world's leading professional association. [En línea] IEEE, 2006. [Citado el: 3 de Febrero de 2008.]
- I-Sol S.A. Intelligent Solutions. 2008.** *Software Factory. Inserción de I-Sol S.A. como empresa argentina, en el mercado mundial*. Capital Federal de Argentina : s.n., 2008.
- Iván, Ing. 2008.** Factorías de Software. Implementación de una Factoría en la empresa E4GS. [entrev.] Jorge Luis VALDÉS GONZÁLEZ. *Factorías de Software*. Los Cortijos, 29 de Abril de 2008.
- IZQUIERDO HERRERA, Raykenler y LAZO OCHOA, René. 2007.** *El modelo de diseño del sistema HyperWeb. Módulos de Tratamiento Farmacológico y Configuración*. Ciudad de la Habana : s.n., 2007.
- JACOBSON, IVAR, BOOCH, GRADY y RUMBAUGH, JAMES. 2000.** *El Proceso Unificado de Desarrollo de Software*. [ed.] Andrés Otero. [trad.] Salvador Sánches. Madrid : PEARSON EDUCACIÓN S.A., 2000. pág. 464. Edición en Español. 0-201-57169-2.
- . **2000.** *El Proceso Unificado de Desarrollo de Software*. [trad.] Salvador Sánchez, y otros. 1. Madrid : PEARSON EDUCACIÓN S.A., 2000. pág. 464. Edición en Español. 84-7829-036-2.
- JAUHAR, ALI y TANAKA, JIRO. 1998.** *An Object Oriented Approach to Generate Executable Code from the OMT-base Dynamic Model*. Instituto de Ciencias Informáticas y Electrónica, Universidad de Tsukuba. Japón : s.n., 1998. Informe de Investigación.
- JETTER, Andreas. 2006.** *Assessing Software Quality Attributes. With Source Code Metrics*. Department of Informatics, University of Zurich. Zurich : Software Evolution & Architecture Lab (SEAL), 2006. pág. 56, Tesis de Diploma.
- Jiménez, Gloria Matilde Carrión. 2010.** *Tecnologías de Web Semántica orientada al desarrollo de Servicios Web*. Ciencias de la Computación, Universidad Técnica Particular de Loja. 2010. pág. 165, Tesis.
- . **2010.** *Tecnologías de Web Semántica orientada al desarrollo de Servicios Web*. Ciencias de la Computación, Universidad Técnica Particular de Loja. 2010. pág. 165, Tesis.
- KNIBERG, Henrik. 2007.** *Una historia de guerra ágil. SCRUM y XP desde las trincheras. Cómo hacemos SCRUM*. [ed.] Diana PLESA. [trad.] Ángel MEDINILLA. EE.UU. : Enterprise Software Development Series (InfoQ), 2007. Enterprise Software Development Series. 978-1-4303-2264-1.
- LARMAN, Craig. 2004.** *Applying UML And Patterns*. s.l. : Prentice Hall, 2004. 9780131489066.
- . **2003.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Segunda Edición. México : PEARSON EDUCACIÓN , 2003. pág. 624. Vol. 1, Edición en Español. 8420534382 .
- LÓPEZ, Oscar, LAGUNA, Miguel Ángel y MARQUÉS, José Manuel. 2004.** *Reutilización del Software a partir de Requisitos Funcionales en el Modelo de Mecano: Comparación de Escenarios*. Universidad de Valladolid. 2004. pág. 12.
- 2010.** *Los servicios públicos europeos de Administración electrónica*. 2010.

- Manso, M.A., y otros. 2007.** *Modelo de Interoperabilidad Basado en Metadatos*. Ingeniería Topográfica y Cartográfica, Universidad Politécnica de Madrid. Madrid : s.n., 2007. pág. 20, Tesis Doctoral.
- . **2007.** *Modelo de Interoperabilidad Basado en Metadatos*. 2007. pág. 20.
- MARÍN, Beatriz, CONDORY-FERNÁNDEZ, Nelly y PASTOR, Oscar. 2007.** Calidad de Modelos Conceptuales. Un análisis multidimensional de modelos cuantitativos basados en la ISO 9126. [ed.] Dr. D. José C. VERDÚN y Dr. D. José Antonio GUTIERREZ. *Revista de Procesos y Métricas de las Tecnologías de la Información*. Edición Especial, Octubre de 2007, 4, pág. 92. Número Especial con motivo de la VIII Conferencia Anual de la Asociación Española de Métricas de Sistemas Informáticos. III Encuentro Internacional ISBSG-AEMES.
- Martínez, Pedro Espina. 2007.** *Extensibilidad de UDDI*. 2007. pág. 83.
- Mateos, Francisco Domínguez. 2005.** *LIIBUS: Arquitectura de Sistemas Interoperables entre Middlewares Heterogéneos usando Máquinas Abstractas Reflectivas Orientadas a Objetos*. Departamento de Informática, Universidad de Oviedo. s.l. : Universidad de Oviedo, 2005. pág. 389, Tesis Doctoral. Tesis doctoral presentada por Francisco Domínguez Mateos y dirigida por el Dr. Juan M Cuevas en mayo de 2005.
- MEYER, Bertrand. 1997.** *Object-Oriented Software Construction*. ISE, Santa Barbara : Prentice Hall, 1997. 84-8322-040-7.
- MGI V1.0. [En línea] [Citado el: 9 de febrero de 2011.]  
[http://www.cidge.gob.mx/doc/MG\\_Interop\\_Dic08.pdf](http://www.cidge.gob.mx/doc/MG_Interop_Dic08.pdf).
- Microsoft. 2011.** Información básica sobre tecnología XML (Diseñador XML). *8kfywf4(v=vs.80).aspx.htm*. [En línea] 2011. [Citado el: 15 de marzo de 2011.]  
<http://msdn.microsoft.com/es-es/library/8kfywf4%28v=vs.80%29.aspx>.
- MILLER, Robert y TRIPATHI, Anand. 2003.** *Primitives and Mechanisms of the Guardian Model for Exception Handling in Distributed Systems*. Computer Science Department, University of Minnesota. Minneapolis : s.n., 2003. Artículo de Conferencia. Conferencia Internacional ECOOP'2003 (Exception Handling in Object Oriented Systems: Towards Emerging Application Areas and New Programming Paradigms).
- MONASOR, Miguel Jiménez. 2008.** *Calidad y Medición de Sistemas de Información. Calidad de Proceso. Una orientación hacia el Desarrollo Distribuido de Software*. ALARCOS. Ciudad Real : s.n., 2008. pág. 21.
- Moreno Escobar, Hernan, y otros. 2007.** *Libro blanco de interoperabilidad de gobierno electrónico para América Latina y el Caribe*. 2007. pág. 39.
- MUÑOZ, Javier y PELECHANO, Vicente. 2007.** *MDA vs Factorías de Software*. Dept. de Sistemas Informáticos y Computadores, Universidad Politécnica de Valencia. Valencia : s.n., 2007. pág. 10, Artículo.
- Naranjo, Mauricio. 2007.** *Lineamientos para adopción de arquitectura orientada a servicios para las empresas*. Bogotá D.C. : LUCASIAN LABS, 2007. pág. 44.
- Object Management Group (OMG). 2003.** *MDA Guide Version 1.0.1*. s.l. : OMG, 2003. pág. 62.
- OLMEDILLA ARREGUI, Juan José. 2005.** *Revisión Sistemática de Métricas de Diseño Orientado a Objetos*. Madrid : Universidad Politécnica de Madrid, 2005.
- ortíz, Ana María. EJIS Esquema judicial de ineteroperabilidad y seguridad.**

**PALACIO BAÑARES, Juan. 2007.** *Flexibilidad con Scrum. Principios de diseño e implementación de campos de Scrum.* 2007. Adaptando los procesos a la empresa.. 0710210187520.

—. **2008.** *Navegápolis 2008.* Edición Febrero 2008. s.l. : Navegápolis®, 2008. 0802020409715.

**PALACIO, Juan. 2007.** *Flexibilidad con Scrum. Principios de diseño e implementación de campos de Scrum.* 2007. Adaptando los procesos a la empresa.. 0710210187520.

*Panorámica y situación del estándar EBXML.* **Triguero, Mario Antonio y Barchino, Roberto. 2007.** Madrid : Universidad de Alcalá, 2007. Conferência IADIS Ibero-Americana WWW/Internet 2007. pág. 5.

**PÉREZ MIRIÑÁN, Martín. 2002.** Asociación JavaHispano. *Asociación JavaHispano.* [En línea] Asociación JavaHispano, 17 de Diciembre de 2002. [Citado el: 9 de Febrero de 2008.] [http://www.javahispano.org/contenidos/es/el\\_peligro\\_de\\_los\\_patrones/](http://www.javahispano.org/contenidos/es/el_peligro_de_los_patrones/).

**Pérez, Javier Eguíluz. 2008.** *Introducción a XHTML.* 2008.

**Posadas, Marino. 2000.** *XML. Introducción al lenguaje.* s.l. : Grupo EIDOS, 2000. pág. 85.

**PRESSMAN, ROGER S. 2001.** *Ingeniería del Software. Un enfoque práctico.* [trad.] Darrel Ince. Quinta Edición. Madrid : McGraw Hill/Interamericana de España, 2001. Adaptación al Español..

—. **2001.** *Ingeniería del Software. Un enfoque práctico.* [trad.] Darrel Ince. Quinta Edición. Madrid : McGraw Hill, 2001. Adaptación al Español..

**PUIG, Vicenç, y otros. 2004.** *Control tolerante a fallos. Fundamento y Diagnóstico de fallos.* Departament Enginyeria de Sistemes, Automàtica e Informàtica Industrial (ESAI), Universidad Politècnica de Catalunya. España : s.n., 2004. pág. 34.

**Quintero, Héctor Ramón Peñaranda.** Monografías.com. [En línea] <http://www.monografias.com/trabajos23/informatica-juridica/informatica-juridica.shtml>.

**R. CHIDAMBER, Shyam y F. KEMERER, Chris. 1994.** A Metrics Suite for Object Oriented Design. [aut. libro] IEEE. *IEEE Transactions on Software Engineering.* EE.UU. : IEEE Press Piscataway, NJ, USA, 1994, Vol. 20, págs. 476 - 493.

**Rational Software Corporation. 2003.** Ayuda extendida de RUP (Rational Unified Process). Versión 2003.06.00.65 *Rational Unified Process.* s.l. : Rational Software Corporation, 2003.

**Redondo, Peis, Montero, Hassan y Herrera, Viedma.** *Ontologías, metadatos y agentes: recuperación "semántica" de la información.* pág. 10.

**REIMER, Darrell y SRINIVASAN, Harini. 2003.** *Analyzing Exception Usage in Large Java Applications.* IBM Research, IBM. New York : s.n., 2003. Artículo de Conferencia. Conferencia Internacional ECOOP'2003 (Exception Handling in Object Oriented Systems: Towards Emerging Application Areas and New Programming Paradigms).

**RENZEL, Klaus. 2003.** *Error Handling for Business Information Systems. A Pattern Language.* Primera Edición. s.l. : software design & management (sd&m), 2003. pág. 132.

**Romero, Alfredo Reino.** Introducción a xml en castellano. [En línea] <http://www.ibium.com/alf/xml/colima2000.pdf>.

**ROSANIGO, Zulema Beatriz. 2000.** *Maximizando reuso en software para Ingeniería Estructural. Modelos y Patrones.* Argentina : s.n., 2000. pág. 113. Tesis de Maestría en Ingeniería de Software.

- SCHMULLER, Joseph. 2000.** *Aprendiendo UML en 24 horas.* [ed.] Oscar MADRIGAL MUÑIZ. Primera Edición. México : Pearson Educación, 2000. pág. 448. 968-444-463-X.
- SHAW, Mary, y otros. 2007.** *First Steps toward a Unified Theory for Early Prediction of Software Value.* Proc. IEEE Equity, IEEE. Amsterdam : s.n., 2007. Artículo de Conferencia. Conferencia Internacional IEEE EQUITY 2007.
- SIEDERDLENBEN, Johannes. 2003.** *Errors and Exceptions – Rights and Responsibilities.* SD&M Research. Munich : s.n., 2003. Artículo de Conferencia. Conferencia Internacional ECOOP'2003 (Exception Handling in Object Oriented Systems: Towards Emerging Application Areas and New Programming Paradigms).
- Tascón, Andrés Guzman.** *"Interoperabilidad de los Documentos Electrónicos (DE)". Estado del Arte.*
- Tortosa, Salvador Otón. 2006.** *Propuesta de una arquitectura de software basada en servicios para la implementación de Repositorios Objetos de Aprendizaje Distribuidos.* 2006. pág. 313.
- Universidad de las Ciencias Informáticas. 2007.** *Arquitectura para los sistemas que conforman la Intranet Universitaria.* Habana : s.n., 2007. pág. 25.
- Valledor Pellicer, Pablo.** *Servicios Web Semánticos.*
- VÉLEZ SERRANO, , José F., y otros. 2005.** *Técnicas avanzadas de diseño de software: Orientación a objetos, UML, patrones de diseño y Java.* Madrid : Universidad Rey Juan Carlos, 2005. pág. 171. Vol. 1.
- W. COOPER, James. 1998.** *The Design Patterns .* s.l. : Addison-Wesley, Design Patterns Series, 1998.
- WIPER, M.P. y WILSON, S. 2003.** *Comprobación de Softwrae. Un análisis Bayesiano.* Departamento de Estadística y Econometría, Universidad Carlos III de Madrid. España : s.n., 2003. Artículo de Investigación. 27 Congreso Nacional de Estadística e Investigación Operativa.
- Zapater, José Javier Samper. 2005.** *Ontologías para servicios web semánticos de información de tráfico: Descripción y herramientas de explotación.* 2005. pág. 366.
- ZARAZAGA, F.J., y otros. 2005.** *Automatizando el paso de Diseño Orientado a Objeto a Codificación mediante el uso de Metainformación basada en facets.* Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza. 2005. pág. 12.