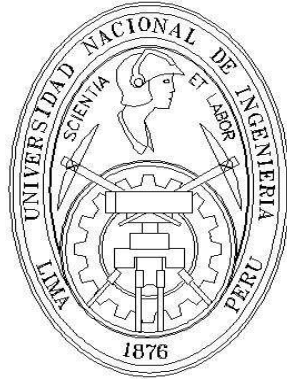


**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**



**DISEÑO E IMPLEMENTACIÓN DE UNA
BIBLIOTECA DIGITAL DISTRIBUIDA BASADA EN
WEB SERVICES PARA EL SECTOR EDUCACIÓN
TESIS**

**PARA OPTAR EL GRADO DE MAESTRO EN CIENCIAS
MENCIÓN: TELEMÁTICA**

PRESENTADA POR:

ALAND BRAVO VECORENA

LIMA - PERÚ

2011

A mis padres: Oliver y Rosy, y a mis hermanos Evelyn y Darwin

Agradecimiento

Quiero expresar mi agradecimiento a los docentes de la sección de Posgrado de la Universidad Nacional de Ingeniería por el apoyo prestado en el desarrollo de esta Tesis; y, en particular, a mi asesor, MSc. Arturo Vilca Román, por su constante guía y aliento. Deseo también expresar mi agradecimiento a la Sección de Posgrado de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Ingeniería, por permitirme utilizar sus instalaciones para diseñar e implementar el prototipo del sistema de biblioteca de esta tesis, y al CONCYTEC por su valioso apoyo directo o indirecto en la culminación de este trabajo de investigación.

Extracto

El presente trabajo está referido a la creación de una Arquitectura Orientada a Regulaciones para el diseño e implementación de una Biblioteca Digital Distribuida basada en servicios web, para lo cual se ha diseñado e implementado un prototipo base como prueba de concepto, con lo cual se espera servir como referente de mejora metodológica y tecnológica en el sector educativo. ROA es una arquitectura creada por el autor de esta tesis, que se basa en cuatro componentes: Metodologías de dirección del proyecto; Tecnologías de apoyo ROA; Infraestructuras de regulación, y; Líneas de desarrollo base. El gran valor de la propuesta arquitectura ROA es alto: debido a que integra las mejores metodologías y tecnologías actuales para el desarrollo de aplicaciones empresariales. Con la masificación de Internet se acrecienta la necesidad de integrar sistemas que de manera sistemática den soporte a diferentes procesos de una organización. La decisión de implementar una aplicación de biblioteca digital distribuida mediante el uso de la arquitectura ROA, como prueba de concepto, nació del interés de comprobar la viabilidad metodológica y técnica al momento de resolver posibles dificultades en su aplicación real. Sin embargo, la arquitectura ROA propuesta no se limita al sector educativo y los frutos de esta investigación muy fácilmente podrán ser aplicados a otros contextos: como el Sector Salud, Sector de la Pymes, etc., es decir, en todos aquellos sectores donde la integración de sistemas de información a bajo costo ameriten un valor notorio para la nación. En el presente proyecto se construye un prototipo de biblioteca digital, distribuida con arquitectura ROA, que utiliza la infraestructura compuesta por un servidor de procesos que decodifica el lenguaje BPEL, generado desde el "lenguaje de modelado de procesos BPMN" de los procesos convencionales modelados de una biblioteca tanto a nivel de flujos de información; así como de interfaces de usuario. Dicho servidor de procesos se acopla con el servidor de aplicaciones: vía los WSDL, generados a partir de los casos de uso UML e implementados vía componentes empresariales J2EE-JBoss. La ventaja de utilizar ROA se materializa en una metodología unificada para la dirección del proyecto que integra principalmente el PMBOK-PMI con el RUP, como parte de las técnicas metodológicas para la ingeniería de software. Con el componente Tecnológico de Soporte ROA, se utilizan tecnologías sobre código abierto tales como BPMS-Intalio,

MySQL, Red Hat Linux, JBoss y Ajax para la capa de presentación; Axis para la capa de servicios, los cuales permiten asegurar la inversión del proyecto manteniendo un costo-beneficio aceptable. Con las Infraestructuras de Regulación ROA se articula el cumplimiento regulatorio de estándares internacionales mediante el uso de las librerías de infraestructura regulatoria; en nuestro caso se utiliza ITIL y 6-Sigma. Con la línea de Desarrollo ROA se podrá evaluar resultados a escala nacional, considerando las ventajas y desventajas, de la propuesta arquitectura ROA versus la arquitectura SOA. Finalmente se detallan los resultados en cuanto se refiere a: tiempo de desarrollo, interface de usuario e interoperabilidad.

Abstract

This work refers to the Regulation Oriented Architecture (ROA) for a design and implementation of a distributed digital library, for which a base prototype has been designed to implement proof of concept, which is expected to be a reference for methodological and technological improvements in education sector. ROA is an architecture created by the thesis's author that is based on four components: project management methodologies, ROA support technologies, infrastructure regulations and development base lines. The great value of the proposed architecture ROA has been high because it integrates the best current methodologies and technologies for the development of business applications. With the Internet's explosion there is a growing need to integrate systems that support an organization's processes in a systemic manner. The decision to implement a distributed digital library application, using ROA architecture as a proof of concept, has born from interest to test the feasibility and technical methodology in resolving any difficulties in actual implementation. But the proposed ROA architecture has not been confined to the education sector and the fruits of this investigation can be apply very easily to other contexts such as health sector, SME Sector, etc. it means all those areas where information systems integration at low cost has a remarkable value for the nation. This project builds a digital library prototype using ROA distributed architecture over SOA infrastructure, which consists of a server process that decodes the language from the BPEL language generated from BPMN conventional processes modelled from a library, both level information flows and user interfaces. The process server is coupled to the application server via the WSDL generated from UML use cases implemented via JBoss J2EE-business components. The advantage of using ROA is that it uses a unified methodology for project management which includes mainly the PMBOK-PMI with the RUP as part of the methodological techniques for software engineering. With the Support Technology ROA component we used open source technologies such as BPMS-Intalio, MySQL, Red Hat Linux, JBoss, Ajax for the presentation layer and Axis to be the service layer, which can ensure the project investment maintaining an acceptable cost benefit. With Regulatory Infrastructure ROA enables regulatory compliance of international standards through the use of regulatory infrastructure libraries, in our case we use ITIL and 6-Sixma. With ROA Development line base we directly assess results at country level, considering the advantages and disadvantages of the proposed ROA versus SOA architecture. Finally the results are detailed in terms of development time, user interface and interoperability concerns.

Tabla de contenido

Capítulo 1.	1
Introducción	1
1.1. Cuestiones preliminares	5
1.2. Planteamiento del problema	14
1.3. Antecedentes	17
1.4. Solución al problema	18
1.5. Objetivos de la tesis	27
1.6. Organización de la tesis	30
Capítulo 2.	32
Consideraciones preliminares	32
2.1. La Informática Educativa en el País	36
2.1.1 La Educación Virtual	36
2.1.2 Desarrollo de Software Educativo mediante la Ingeniería de Software	40
2.2. Aplicaciones Telemáticas para innovar metodologías de enseñanza - aprendizaje	42
2.2.1. Notación para Modelado de Procesos de Negocio: BPMN	45
2.3. Evolución y Crecimiento de los Web Services	57
2.3.1. Conceptos fundamentales	68
2.3.2. Introducción a los Web Services	76
2.3.3. Tecnologías y Arquitecturas de Desarrollo de los Servicios Web	77
2.3.4. Estándares y Seguridad en Web Services	82
2.3.5. Estrategias y Soluciones basadas en Web Services	89
2.4. Aplicación de los Web Services en las Telecomunicaciones	91
2.4.1. Arquitectura NGN	91
Capítulo 3.	106
Diseño e Implementación del Sistema Automatizado de la Biblioteca Digital	106
3.1. Análisis de la Biblioteca Digital Distribuida	111
3.1.1. Consulta a fuentes de información apropiada e identificación	

de problemas	111
3.1.2. Análisis de posibles causas de los problemas detectados en el sistema convencional	111
3.1.3. Análisis de alternativas de solución	117
3.1.4. Establecimiento del papel del computador	119
3.1.5. Planificación del desarrollo de la Biblioteca Digital Distribuida	120
3.2. Diseño de la Biblioteca Digital Distribuida	138
3.2.1. Especificaciones de Diseño	142
3.2.2. Diseño Educativo	148
3.2.3. Diseño de Comunicación	149
3.2.4. Diseño Computacional	151
3.2.5. Preparación y revisión de un prototipo	152
3.3. Desarrollo de la Biblioteca Digital Distribuida	153
3.3.1. Definición de Estrategia	153
3.3.2. Detalle del Lenguaje SOA-UML versus ROA-BPM en el Diseño del Sistema Dexter's Lib	155
3.3.3. Documentación del Proyecto	164
Capítulo 4.	185
Revisión de la Solución, Pruebas del Prototipo y Resultados	
4.1. Revisión de la solución	185
4.2. Prueba Piloto	191
4.2.1. Preparación de la prueba piloto	192
4.2.2. Desarrollo de la prueba piloto	194
4.2.3. Análisis de resultados de la prueba piloto	194
4.2.4. Toma de decisiones	197
Capítulo 5.	199
Conclusiones	
5.1. Resumen de los resultados y sus implicancias	199
5.2. Recomendaciones para trabajos futuros	206

Apéndice A.	209
Proyecto Huascarán	209
Apéndice B.	217
Glosario de Términos	217
Apéndice C.	220
Código Fuente	220
Bibliografía.	229

Listado de Figuras

Figuras del Capítulo 1.

1.1.	Propuesta de una Arquitectura Orientada a Regulaciones ROA para el Sector Educación	2
1.2.	Esquema típico de una Arquitectura orientada a servicios SOA convencional	3
1.3.	Detalles de la Plataforma de la Arquitectura Orientada a Regulaciones ROA	4
1.4.	Pirámide de Infraestructura Regulatoria ROA que consta de cuatro capas y dividido en siete escalones	5
1.5.	Esquema de un Escenario Bus de Servicio Regulado ROA-ESB	16
1.6.	Plataforma de Bus Educativo Dexter's Lib	16
1.7.	Paradigma de enfoque centralizado de ROA-BPM	21

Figuras del Capítulo 2.

2.1.	Macro Esquemas de la Arquitectura Orientada a Regulaciones ROA	33
2.2.	Ciclo de Vida Tradicional SOA	33
2.3.	Pirámide de Seguridad Telemática ROA	34
2.4.	Ejemplo de un Escenario de Seguridad ROA para una Plataforma de Código Abierto	35
2.5.	Etapas de un Modelado BPM Intalio	45
2.6.	Proceso automatizado concentrado bajo el enfoque de Arriba hacia Abajo y de Abajo hacia Arriba	46
2.7.	Notaciones gráficas elementales del BPMN	48
2.8.	Listado de Elementos de Notación Gráfica BPMN	48
2.9.	Notaciones gráficas de Conectores BPMN	49
2.10.	Gráfico de Pools y Lanes BPMN	49
2.11.	Notaciones gráficas de Artefactos BPMN	49
2.12.	Notaciones gráficas de Eventos BPMN	50
2.13.	Notaciones gráficas de Compuertas BPMN	51
2.14.	Notaciones gráficas a detalle de Eventos BPMN	52
2.15.	Notación de Compuerta Exclusiva BPMN	53
2.16.	Notación de Compuerta Paralela BPMN	54

2.17.	Notación de Compuerta Exclusiva BPMN con Eventos	54
2.18.	Notación de Compuerta Inclusiva BPMN	55
2.19.	Notación de Compuerta Paralela Compuesta BPMN	55
2.20.	Notación de Tareas y Sub Procesos BPMN	56
2.21.	Notación de Flujos en Serie y Paralelo BPMN	56
2.22.	Operaciones y cometidos de los Web Services	79
2.23.	Componentes Lógicos de los Web Services	83
2.24.	Diagrama de Secuencia de los Web Services	99
2.25.	Caso de Uso de los Web Services	100
2.26.	Protocolo de Acceso a objetos sencillo	101
2.27.	Lenguaje de Descripción de los Web Services	102
2.28.	Descripción, detección e integración universales	103
2.29.	Lenguaje de inspección de los Web Services	104
2.30.	API de Java para llamadas a procedimientos remotos basadas en XML	104
2.31.	Esquema de despliegue de la Plataforma UNICA que utiliza una arquitectura basada en Web Services	105

Figuras del Capítulo 3.

3.1.	Tasa de Asistencia de Educación Inicial de la población de 3 a 5 años de edad, 1997-2001	112
3.2.	Evolución de la Población en el tiempo	113
3.3.	Arquitectura básica de un nodo aislado de la Biblioteca Digital Distribuida	118
3.4.	Arquitectura de un Nodo BPM de la Biblioteca Digital Distribuida	119
3.5.	Etapas de la Simplificación Administrativa basado en la Metodología GTZ	127
3.6.	Tipos de Procesos y su interacción con el Cliente	129
3.7.	Interacción Interna entre los elementos de un Proceso	129
3.8.	Nivel de resolución de los Objetivos Estratégicos a los Resultados	130
3.9.	Diagrama de Bloques utilizado en la Metodología GTZ	133
3.10.	Diagrama de Actividades del Rediseño del Proceso	137
3.11.	Diagrama de Bloques utilizado en el modelo ROA-BPM a partir de la Metodología GTZ	139
3.12.	Diagrama de Componentes SOA-J2EE	143
3.13.	Esquema de comunicación entre los nodos del Sistema de Biblioteca Digital Distribuida basado en SOA-J2EE	145

3.14.	Diagrama de Capas que compone el Sistema de Biblioteca Digital Distribuida basado en ROA-BPM	146
3.15.	Esquema de Comunicación entre los nodos del Sistema de Biblioteca Digital Distribuida basado en ROA-BPM	147
3.16.	Roles de los Actores ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	155
3.17.	Modelado de la Capa de Dominio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	159
3.18.	Modelado de la Capa de Persistencia ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	160
3.19.	Modelado de la Capa de Presentación ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	160
3.20.	Modelado de la Capa de Servicio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	161
3.21.	Modelado de la Capa de Servicio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida	162
3.22.	Vista Lógica de la arquitectura de los componentes	167
3.23.	Actores del sistema de biblioteca	168
3.24.	Proveedores y Consumidores del sistema de biblioteca	168
3.25.	Participantes de Búsqueda de Recurso del Sistema de Biblioteca	169
3.26.	Diagrama de Casos de Uso	170
3.27.	Vista Global de Actores y Casos de Uso del sistema de biblioteca	171
3.28.	Diagrama de Clases que se mapean a los Entity Beans, y estos a su vez se mapean a una tabla almacenada en la base de datos	172
3.29.	Diagrama de Secuencia para Caso de Uso Búsqueda de Recursos	173
3.30.	Diagrama de Secuencia para Caso de Uso Registrar Recurso	174
3.31.	Diagrama de Colaboración para Caso de Uso Registrar Recurso	175
3.32.	Diagrama de Colaboración para Caso de Uso Búsqueda de Recurso	176
3.33.	Diagrama de Interacción entre los Componentes J2EE	177
3.34.	Flujo Normal del Caso de Uso: Registrar Recursos del Sistema de Biblioteca	178
3.35.	Detalle del bloque del Nodo Cliente en el Diagrama de Interacción entre los Componentes de la Figura 3.25	179
3.36.	Detalle de Descomposición de las Fases 1, 2, 3 y 4 del Proceso de	

Gestión de Servicios de Biblioteca	180
3.37. Detalle de Descomposición de las Fases 5, 6 y 7 del Proceso de Gestión de Servicios de Biblioteca	180
3.38. Detalle de Descomposición de la Fase 8 del Proceso de Gestión de Servicios de Biblioteca	181
3.39. Modelado BPMN que invoca el Web Service de Registro de Recurso Bibliográfico con su instancia Libro	181
3.40. Detalle interno del Manejo de Excepciones BPMN para Inserción del Recurso Bibliográfico Libro	182
3.41. Detalle del código BPMN que implementa gráficamente la transacción del Registro del Recurso Bibliográfico Libro (Parte 1/2)	183
3.42. Detalle del código BPMN que implementa gráficamente la transacción del Registro del Recurso Bibliográfico Libro (Parte 2/2)	183
3.43. Detalle del bloque BPMN que implementa una subrutina que llama a un Web Service en forma gráfica	184
3.44. Detalle del Modelo de Base de Datos utilizado en el Proyecto BPMN de la Biblioteca Digital Distribuida	184
Figuras del Capítulo 4.	
4.1. Utilitario de Mapeo XML de la interface de Diseño BPM Intalio	186
4.2. Diagrama BPMN base de Comunicación Asíncrona utilizando la Vista de Correlación y Plantillas (1/2)	187
4.3. Diagrama BPMN base de Comunicación Asíncrona utilizando la Vista de Correlación y Plantillas (2/2)	187
4.4. Escenario de invocación a un Procedimiento Almacenado Externo	188
4.5. Detalle de implementación del Procedimiento Almacenado Externo	189
4.6. Escenario de invocación a un SQL creado directamente en la herramienta de Diseño BPM Intalio	189
4.7. Detalle del Esquema de Datos XSD utilizado en la Capa de Servicio	190
4.8. Detalle del XSLT utilizado en la Capa de Servicio	190
4.9. Configuración de despliegue de las Capas de Dominio y Persistencia del Proyecto de Biblioteca Digital Distribuida	191
4.10. Configuración previa de despliegue de las Capas de Persistencia y Servicio del Proyecto de Biblioteca Digital Distribuida	193

4.11. Configuración Exitosa de Despliegue de las Capas de Persistencia y Servicio del Proyecto de Biblioteca Digital Distribuida	193
4.12. Listado de Procesos implementados para el Despliegue del Proyecto de Biblioteca Digital Distribuida	195

Figuras del Capítulo 5.

5.1. Interfaz Gráfica de Usuario: Registro de Usuario	200
5.2. Interfaz Gráfica de Usuario: Registro de Libro	200
5.3. Interfaz Gráfica de Usuario: Búsqueda de contenidos	201
5.4. Validación en Base de Datos de la Interfaz Gráfica de Usuario: Registro de Recurso Bibliográfico	203
5.5. Validación en Base de Datos de la Interface Gráfica de Usuario: Registro de Nodo	204
5.6. Validación en Base de Datos de la Interfaz Gráfica de Usuario: Registro de Usuario	204

Listado de Cuadros

Cuadros del Capítulo 1.

1.1.	Características del Conocimiento Científico	9
1.2.	Lenguajes de Procesos más importantes	20
1.3.	Resumen de la Semántica Ontológica BPMN	22

Cuadros del Capítulo 2.

2.1.	Tabla de Perfiles ROA	32
2.2.	Lenguajes de Procesos	46
2.3.	Semántica de la Notación BPMN ROA	47
2.4.	Historia de la Tecnología de los Web Services	78
2.5.	Evolución de la Tecnología de los Web Services	81
2.6.	Comparación de diversas arquitecturas de Web Services	84

Cuadros del Capítulo 3.

3.1.	Niveles de Modelado ROA-BPM	109
3.2.	Matriz de Riesgos ROA-BPM	125
3.3.	Matriz de Riesgos versus Impacto ROA-BPM	126
3.4.	Matriz de Priorización ROA-GTZ	130
3.5.	Matriz de Actores ROA-GTZ	131
3.6.	Símbolos ASME usados en la adecuación de la Metodología ROA-GTZ para identificar los tipos de actividad	131
3.7.	Calificación del valor de actividad ROA-GTZ	132
3.8.	Matriz de avance de un Proyecto ROA-BPM utilizando GTZ y PMBOOK	134
3.9.	Matriz de Habilidades ROA-GTZ	135
3.10.	Matriz ASME del Proceso con Procedimiento Rediseñado con ROA-GTZ	136
3.11.	Comparación de indicadores de desempeño entre el Procedimiento Actual y el Procedimiento Propuesto con ROA-GTZ	138
3.12.	Comparación de Tecnologías y Metodologías utilizadas en las arquitecturas SOA-J2EE y ROA-BPM	147
3.13.	Comparación entre DCOM, CORBA y los Web Services	152
3.14.	Comparación de la Arquitectura Orientada a Servicios SOA con Arquitectura Orientada a Regulaciones ROA	163
3.15.	Descripción de las definiciones usadas en el Prototipo de la Biblioteca Digital Distribuida	166

Listado de Pie de Página

Pie de Página del Capítulo 3.

- 1) Fuente: Guía Metodológica complementaria para el análisis de las matrices de delimitación de competencias y distribución de funciones sectoriales 128
- 2) Cabe aclarar que los casilleros marcados en rojo no descalifican la labor que los funcionarios vienen desempeñando, sino que únicamente indican áreas de mejora que pueden ser abordadas en futuros programas de capacitación 135

Capítulo 1

Introducción

Actualmente las instituciones tanto públicas como privadas cuentan con procesos que requieren un cambio dinámico en el tiempo, de modo que puedan adaptar sus procesos encapsulados en servicios a estándares internacionales con el objetivo de empalmar su organización a un medio globalizado. En este contexto surge la necesidad de automatizar los procesos organizacionales con tecnologías emergentes que permitan una rápida adaptación de las mejores prácticas telemáticas. Bajo esta realidad, surge la necesidad de concebir una Arquitectura Orientada a Regulaciones (ROA), que permita guiar a las áreas de tecnologías de información de las instituciones académicas y empresariales; de modo que, con un presupuesto y en tiempos reducidos, se concrete la adaptación ROA como estándar de facto al momento de implementar proyectos de tecnología de información.

La ROA está basada en cuatro pilares, cada uno de los cuales cumple un rol distinto para lograr el objetivo propuesto en concreto:

1. Metodologías de Dirección ROA
2. Tecnologías de Soporte ROA
3. Infraestructura de Regulación ROA
4. Líneas de Desarrollo ROA

Para valorar la importancia de la arquitectura propuesta, se muestra el enfoque tradicional de la Arquitectura Orientada a Servicios SOA, en el cual se aprecia que el componente regulatorio es una deficiencia intrínseca de esta arquitectura.

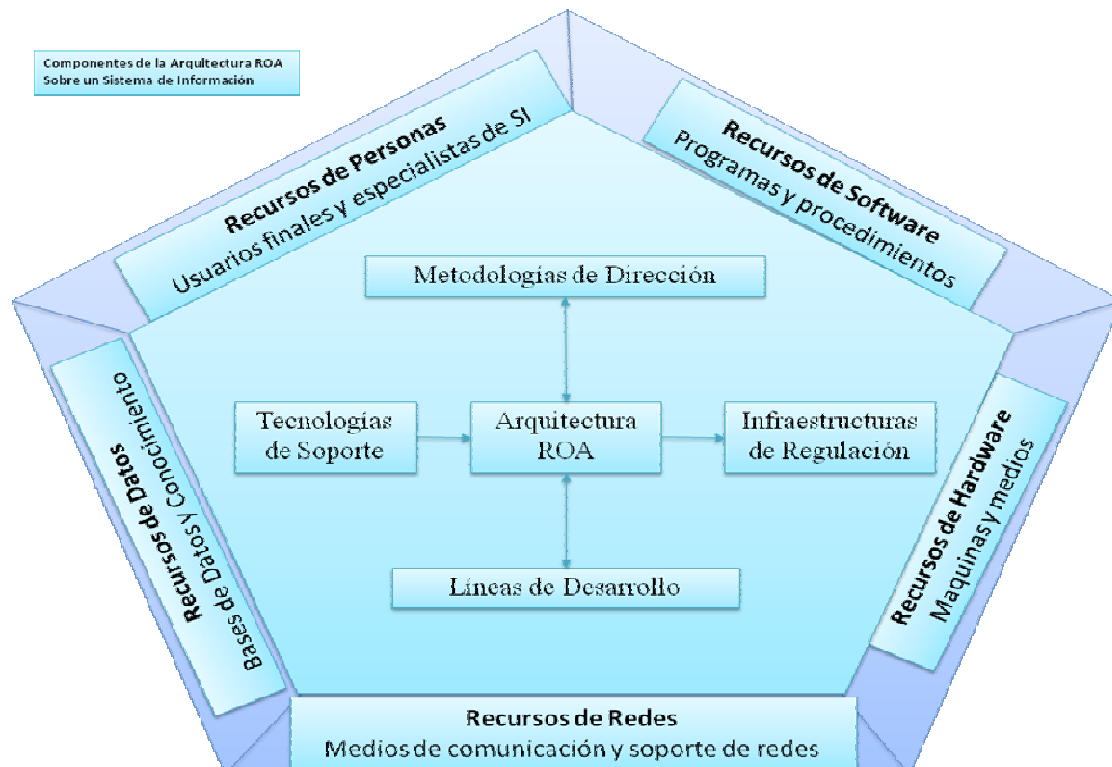


Figura 1.1: Propuesta de una Arquitectura Orientada a Regulaciones (ROA) para el Sector Educación

La **Arquitectura Orientada a Regulaciones (ROA)** es un conjunto de metodologías, tecnologías y regulaciones que cumplen estándares internacionales y/o nacionales; orientadas a integrar las soluciones de infraestructura digital al momento de automatizar procesos dentro de una organización en un marco de ciudades digitales contemporáneas.

La ROA permite dotar de una arquitectura integral sobre el cual se desarrolla un sistema de información; es decir, permite recoger, procesar, almacenar, analizar y difundir información para cumplir un propósito definido por una determinada organización.



Figura 1.2: Esquema típico de una Arquitectura Orientada a Servicios (SOA) convencional

Mientras que en la arquitectura ROA se consideran aspectos metodológicos para gestionar todo un proyecto con un equipo de analistas, arquitectos y desarrolladores. También se consideran las tecnologías para el despliegue del proyecto, luego se enfatiza la infraestructura regulatoria en cuanto a estándares internacionales se refiere, para que finalmente se concrete a las líneas de desarrollo ROA con énfasis en los patrones de procesos empresariales.

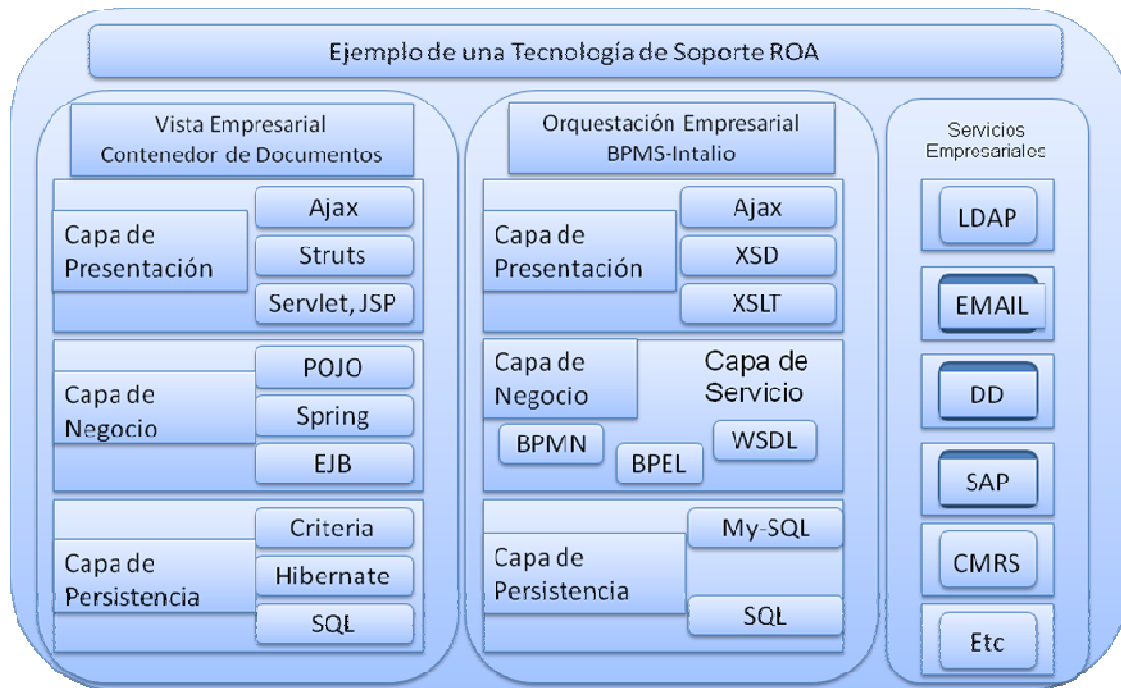


Figura 1.3: Detalles de la Plataforma de la Arquitectura Orientada a Regulaciones ROA

La infraestructura de regulación en seguridad está compuesta de cuatro capas, las cuales están divididas en siete peldaños que conforman la Pirámide de Infraestructura de Regulación ROA.

Dichos niveles son los siguientes:

- Capa 1: Regulación a Nivel de Usuario
- Capa 2: Regulación a Nivel de Red de Acceso
- Capa 3: Regulación a Nivel de Procesos Regulados
- Capa 4: Regulación a Nivel de Servicios

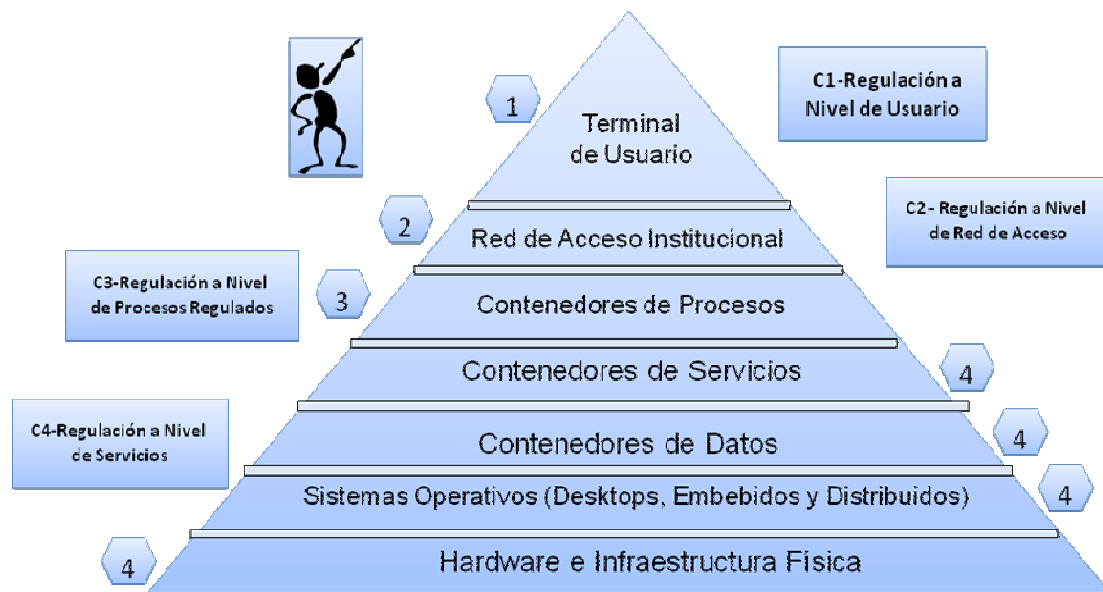


Figura 1.4: Pirámide de Infraestructura Regulatoria en Seguridad ROA que consta de cuatro capas y está dividida en siete escalones

1.1. Cuestiones preliminares

El presente trabajo de tesis fue concretado el año 2002 bajo el título inicial “Biblioteca Digital Distribuida basada en Web Services para el Proyecto Huascarán”, de modo que en el presente año el autor se ha permitido actualizar el enfoque de la tesis, utilizando las tecnologías más recientes y modificando el contenido y alcance. Lo más resaltante es la migración de la notación UML por el del BPMN; el reemplazo de los patrones de diseño con el de los patrones de procesos, así como el reemplazo de la metodología RUP por una arquitectura ROA integral que ha permitido concretar el presente proyecto de investigación aplicada.

Es importante recalcar que BPMN no reemplaza o sustituye a UML, sino que complementa las vistas de modelado. La ROA fue concebida debido a que BPMN aún no tiene una arquitectura propiamente definida, de modo que se

propone una arquitectura orientada a regulaciones orientada hacia meta lenguajes de modelado gráfico, a fin de que sea factible su evolución en el tiempo. Asimismo, el modelado de una biblioteca digital contempla diversos casos de uso. En el presente proyecto nos centraremos al caso de uso de registro de usuario de biblioteca como prototipo de prueba de concepto. En lo que respecta a las interfaces de usuario se enfatiza en el Workflow Tempo de Intalio, descartando la inserción de un Portal, como Liferay, por motivos de tiempo. También la ROA permite unificar diversas metodologías, tecnologías, marcos regulatorios y líneas de desarrollo tales como BPMN, inteligencia de negocios, regla de negocios, entre otras. La importancia de utilizar BPMN es que nos permite simular los procesos modelados para validar la consistencia de los mismos, y luego desplegarlos en un entorno de producción a través de un servidor de procesos.

Existen varios niveles de conocimiento aprendidos y desarrollados en el presente trabajo de investigación de tesis, a citar algunos:

Conocimiento empírico: No se intentó averiguar nuevas técnicas de sistemas distribuidos, sino que se comprobaron varias de las técnicas ya utilizadas en el contexto internacional, tales como la plataforma J2EE y .NET a través de la comprobación experimental aplicada en base al proyecto educativo Huascarán. Sobre este, el aporte consistió en adecuar estas técnicas y presentar una propuesta de un prototipo de una Biblioteca Digital Distribuida basada en Web Services como un propuesta del Sistema de Información para el Proyecto Huascarán; esto porque se tendría previamente que demostrar la superioridad de estas técnicas con respecto a otras, pero que posteriormente se demuestra que es eficiente en ciertos casos: en los cuales se debe cumplir con los estándares en el sector educación.

Conocimiento artístico: En este ámbito se trató de sensibilizar a los actores mediáticos y de decisión para que, a través de las instituciones pertinentes, se pueda plantear una propuesta integral de una plataforma distribuida para implementar una biblioteca digital a escala nacional, teniendo en cuenta la idiosincrasia de nuestras comunidades de la costa, sierra y selva. Si bien es

cierto, se trató de acervar la sensibilidad poniendo una dosis de arte digital, usando para ello un personaje de arte cómica Dexter; pero aún queda muchos temas por resolver.

Conocimiento filosófico: La importancia de este trabajo de investigación no es un mero conocimiento utilitario: el de conocer por conocer, sino el conocimiento por el conocimiento mismo; que ha permitido, de algún modo, indagar sobre las esencias de las cosas; enmarcando lo aprendido a un conocimiento general de los aspectos culturales y políticos de nuestra historia, para dejar los temas algorítmicos computacionales a la CIENCIA que se encarga de los conocimientos parciales a detalle. En este contexto: puede decirse que el proyecto Huascarán fue un fracaso desde el punto de vista de un sistema de información integral; ya que solamente se limitó a equipar conexión a Internet y computadoras, pero es justamente que debemos aprender de nuestros errores para no cometerlos más adelante. En contrapartida, se ha podido desarrollar todo una arquitectura robusta, basada en regulaciones, como una propuesta que intenta institucionalizar métodos basados en mejores prácticas.

Conocimiento científico: Durante el desarrollo de este trabajo daremos énfasis a la ciencia: que es el conjunto de conocimientos racionales, objetivos, metodológicos y sobre todo provisionales, ya que están en permanente evolución para sí mismos y para nuestro entorno actual; tal como el tema de LICENCIA de AUTOR que está siendo desplazada por la LICENCIA de USO COMÚN en un tenor de sociedad digital, en donde el conocimiento es adaptado a una realidad diferente para crear una nueva. En ROA hemos utilizado versiones condensadas y adaptadas a un escenario específico; por ejemplo, se ha creado una metodología para direccionar los proyectos entre una mezcla de PMBOK y RUP adaptado para manejar la notación BPMN en lugar de UML.

En este trabajo de tesis vamos a usar las dos clases de ciencias:

- 1) **Ciencia empírica o fáctica:** Ya que vamos a usar la metodología RUP, que pertenece a una familia dentro del universo de las metodologías de construcción de software y no cuestionaremos ésta, sino que la usaremos

en forma fáctica; que a su vez utiliza un lenguaje para procesar el método que tienen a través del UML: basadas en hechos, experiencias de casos de uso y contrastada con los patrones de diseño tanto a nivel de lógica de negocios como a nivel de implementación en código; lo que nos lleva a la conceptualización de una arquitectura basada en modelos ampliamente aceptados por organizaciones internacionales propulsoras de los estándares.

Dentro de esta ciencia se dividen tres sub categorías a decir:

Ciencias Naturales: Estudia la naturaleza.

Física, Botánica, Zoología.

Ciencias Sociales: Estudia la sociedad.

Historia, Psicología, Antropología, Contabilidad, Economía, etc.

Ciencias Normativas: Busca ubicarse dentro del espectro de las ciencias.

Derecho, Moral.

- 2) **Ciencia no empírica o formal:** Dado que vamos a usar algoritmos computacionales, haremos uso de la lógica y las matemáticas; por ello el objeto de estudio de por sí es abstracto, y se basa en un contexto temporal que es la propuesta de un Sistema de Información enmarcado en un prototipo de una Biblioteca Digital Distribuida como una prueba de concepto al Proyecto Huascarán del sector educativo Peruano.

A continuación, vamos a listar las características del conocimiento científico que validará nuestra hipótesis y finalmente los resultados obtenidos:

Característica	Descripción
Racional	Basado en la razón, juicios, normas y pautas. Es demostrable
Objetivo	No se basa en sensaciones, intereses, procura que se base en evidencia, limpia todo aquello que sea subjetivo
Especializado	A veces por probar la aplicación real de una técnica computacional, se deja de lado el enfoque de un todo de los

	usuarios finales del sistema tales como Alumno, Docente y Personal Administrativo, el cual podría perder el rumbo de la solución óptima
Lenguaje propio	La producción de nuevo conocimiento genera nuevo léxico especializado en el tema
Predictivo	Es demostrable, ya que se sabe con anterioridad lo que va a pasar
Metódico	Establece procedimientos repetitivos para producir nuevo conocimiento, en contraste con lo sistemático que utiliza algo probado que funciona y, por ende, no produce nada nuevo

Cuadro 1.1: Características del conocimiento científico

Entonces podemos concluir que la ciencia es técnica (reglas complejas) más arte (carece de reglas), tiene intuición inmediata y exige disposición intelectual, que a través del método de investigación vamos a plantear un tema de investigación (diseño e implementación de una biblioteca digital distribuida basada en Web Services) y surge del problema de plantear una metodología sistémica basada en una propuesta de Arquitectura Orientada a Regulaciones que permita desarrollar seriamente un proyecto de trascendencia nacional en el sector educación, para lo cual se plantea una Hipótesis que es una posible respuesta a solucionar el problema planteado usando técnicas avanzadas de ingeniería de software, así como una plataforma de un sistema de información BPMN: sobre el que se despliega un Web Service para la comunicación distribuida entre nodos remotos; dicha hipótesis será demostrada a través del desarrollo del prototipo a fin de que posteriormente sea aplicado a un universo más amplio.

En el presente trabajo de tesis se hace uso de la investigación científica en sus tres etapas:

- 1) Etapa de Planteamiento: A continuación los aspectos más sobresalientes:
 - a. Título del Proyecto: Diseño e Implementación de una Biblioteca Digital Distribuida basada en Web Services para el Sector

Educación.

- b. Problema: Falta de Integración entre los alumnos, docentes y administrativos del Sector Educación del Estado Peruano, de modo que permitan utilizar técnicas computacionales avanzadas para autoevaluación y asistencia en la toma de decisiones tanto de alumnos, docentes, así como del personal administrativo.
- c. Justificación del Problema: Como un antecedente podemos citar el fracaso rotundo del Proyecto Huascarán, que por falta de una asesoría especializada solo se limitó a instalar computadoras y acceso a Internet a los colegios, y un mero portal de educación donde no existe la integración real entre los actores del sector educación. En la actualidad el Gobierno planea ejecutar una mayor cobertura del servicio de educación, motivo por el cual se hace necesario el ahorro de costos operativos y la experiencia de los errores del pasado para no volverlos a cometer; y que permita reutilizar eficientemente los recursos del personal docente y administrativo. Asimismo demostrar que un prototipo de viabilidad tecnológica de dicha solución, permitiría comprobar lo acertado de las técnicas computacionales de integración vía Web Services.
- d. Objetivos: Toda vez que el alcance de este proyecto de tesis es amplio, he decidido agrupar los objetivos en:
 - i. Objetivo de índole Telemático:
 - 1. Desarrollo de una Arquitectura Basada en Regulaciones ROA para el Desarrollo del Prototipo de una Biblioteca Digital Distribuida basada en Web Services utilizando BPMN.
 - 2. Estudio de la Integración entre la metodología de construcción de software RUP con la metodología de gestión de proyecto PMBOK, para la creación de indicadores metodológicos en un proyecto ROA.
 - ii. Objetivo de índole Tecnológico-Computacional:

1. Obtener habilidades en las herramientas usadas en la construcción del prototipo:
 - a. Requerimiento y Análisis: Diseñador de Procesos de Negocios Intalio-BPMS.
 - b. Diseño y Construcción: Servidor de Procesos de Negocios Intalio - BPMN.
 - c. Gestión del Proyecto: Metodologías de Dirección de la Arquitectura Orientada a Regulaciones ROA.
 2. Desarrollo del sustento computacional sobre las que se basan las herramientas de ingeniería de software tales como los patrones de procesos, patrones de diseño, modelado de las entidades que en última instancia se enlazan a tablas relacionales.
- iii. Objetivo de índole Social:
1. Presentar los resultados a la comunidad académica.
- e. Finalidad: Demostrar la viabilidad técnica y teórica de la aplicación de la Arquitectura Orientada a Regulaciones para el diseño e implementación de los Web Services y automatizar los procesos de atención del sector de Educación a través de la implementación de un prototipo de una Biblioteca Digital Distribuida basada en patrones de negocios y patrones de diseño de software; en la que se emplea herramientas profesionales de construcción de software empresarial. Demostrar la necesidad de contar con una plataforma digital distribuida de Bibliotecas en donde participen alumnos, docentes y personal administrativo a nivel nacional. Utilizar BPMN para implementar el prototipo de modo que permita validar la metodología al momento de implementar tecnológicamente los patrones de procesos: desde macro procesos hacia procesos a nivel de resolución de detalle que nos permita analizar plazos y riesgos del proyecto.

- f. Hipótesis: Carencia de un sistema digital automatizado de integración en el sistema de Educación a nivel nacional, que impide la aplicación de técnicas computacionales para apoyo en auto evaluación y toma de decisiones a nivel operativo (cantidad de libros digitales, pizarras por región o ciudad, etc.). Al efecto de limitar los alcances de la tesis se circunscribe el trabajo de tesis al sub conjunto de digitalización de una Biblioteca Digital Distribuida, debido a que no existe un sistema real parecido en el Perú. Se ha escogido un referente más cercano: el Proyecto Huascarán, el cual se va a utilizar como un comparador al prototipo propuesto para comprobar las ventajas de la plataforma distribuida.
- g. Variables Dependientes:
 - Metodología y Herramientas para Gestión del Proyecto:
 - i. Metodología: Variante de PMBOK y RUP con BPMN.
 - ii. Herramienta: Intalio Designer.
 - Técnicas y Herramientas de Requerimientos y Análisis:
 - iii. Técnicas: ROA, MDA, SOA, SOAP, BPEL.
 - iv. Herramientas: Intalio Server.
 - Técnicas y Herramientas de Diseño y Construcción:
 - v. Técnicas: Patrones de Diseño de Procesos.
 - vi. Herramientas: Intalio BPMS.
- h. Variables Independientes:
 - 1) Prototipo de Biblioteca Digital Distribuida basada en BPMN.
 - 2) Aplicabilidad al Proyecto Huascarán del Gobierno Peruano.
- i. Medición de la efectividad de las variables Independientes:

Usaremos los siguientes indicadores para las variables independientes:

 - i. Cobertura
 - ii. Escalabilidad
 - iii. Robustez
 - iv. Disponibilidad

- v. Interoperabilidad
 - j. **Ámbito de la Investigación:** Sistema de Educación del Perú
 - k. **Población o Universo:** Proyecto Huascarán.
 - l. **Técnicas de Acopio de Información:** Las técnicas a aplicar serán:
 - i. Tesis de Maestría y Doctorado relacionados al tema.
 - ii. Publicaciones de investigaciones realizadas por la IEEE.
 - iii. Material relacionado al tema tomado de Internet.
- 2) **Etapa de Ejecución:** Este trabajo de tesis se desarrolló en el Laboratorio de la Facultad de Ingeniería Eléctrica y Electrónica de la UNI durante los años 2002 al 2003, periodo durante el cual el autor de la tesis llevó a la práctica las técnicas de sistemas distribuidos, aplicados al sector de educación utilizando J2EE y BPMN con énfasis en la automatización de procesos.
- 3) **Etapa de Presentación de Resultados:** Orientado a comunicar los resultados:
- a. Oral: Exposición de la tesis.
 - b. Escrita: Redacción del volumen de tesis.
 - c. Publicación de la Tesis en plataformas virtuales de tesis digitales.

Se prevé que el éxito de las aplicaciones basadas en web tomará un papel importante en la fase evolutiva de la computación distribuida, la cual está basada en protocolos de Internet y estándares XML. Los web services proveen un prometedor mecanismo para comunicación y colaboración entre aplicaciones de negocios; los cuales son construidos empleando varios recursos que permiten el trabajo en conjunto, a pesar de sus diferencias en la capa de implementación. Sin embargo, actualmente hay una carencia nacional del empleo de esta tecnología en el sector educativo. Hoy en día surgen nuevos dilemas al momento de definir estándares de formatos para el intercambio de información educativa en el País, más aún cuando se pretende comprender la organización del sistema educativo nacional, como es el caso del programa Huascarán [ver referencia 21]. Saltan a la vista algunos puntos relevantes:

- Ineficiente uso de los recursos informáticos
- Diversidad de proveedores de sistemas de información
- Baja cooperación entre sistemas de información
- Diversidad de plataformas y modelos de programación

En la última década se han propuesto numerosos esquemas en la literatura para soslayar estos inconvenientes, entre ellos están los Web Services empleando lenguajes java y xml, que definen una técnica de acoplamiento débil entre los diferentes sistemas a integrar. Esta tecnología implica tres aspectos importantes:

- Primero: la introducción de la tecnología empresarial java, la cual separa el software de la plataforma y permite concentrar nuestros esfuerzos en la creación de la lógica de negocios, empleando técnicas de patrones de diseño, implícitamente insertado en el framework Struts.
- Segundo: la introducción del lenguaje extensible de marcas (XML), el cual separa los datos del software, permitiendo de esta forma que diversos sistemas de software puedan compartir datos sin ser afectados por los cambios en las estructuras de los datos; permitiendo utilizar diversos motores de bases de datos (mysql, oracle, informix, etc.) para poder almacenar dicha información.
- Tercero: la introducción de los Web Services, los cuales separan la colaboración de los sistemas de cómputo conectados en redes, permitiendo la delegación de procesos sin estar acoplados fuertemente.

1.2. Planteamiento del problema

En el marco global, dentro del contexto educativo nacional, se ve la necesidad de contar con un sistema nacional virtual educativo (plataforma de software), inter-conectado a través de una plataforma única. Dicha plataforma debería permitir solucionar los problemas sistémicos de manejo educativo para

los diversos roles a través de los actores involucrados. Desde un punto de vista de enfoque sistémico, podemos plantear la necesidad de contar con un Sistema Interconectado Nacional Educativo de Administración y Control, el cual constaría de los siguientes servicios:

- Servicio de Información Académica (SIA)
- Servicio de Administración de Recursos (SAR)
- Servicio de Medición Educativa (SME)
- Servicio de Información de Centros Educativos (SICE)
- Servicio de Información Estudiantil (SIE)
- Servicio de Información de Docentes (SID)
- Servicio de Información de Medios Educativos (SIME)

En el marco específico, el contexto del alcance del presente proyecto está limitado a un módulo que forma parte del Servicio de Información de Medios Educativos del Sistema Interconectado Educativo de Administración y Control; en concreto la interconexión de Bibliotecas Digitales, donde se aplica una instancia del prototipo para acondicionar la solución al Sector Educación, pudiendo ser una propuesta complementaria al Programa Huascarán que pretende fortalecer la educación en el Perú, a través de las tecnologías de información y comunicación. [fig 1.5 y fig 1.6]

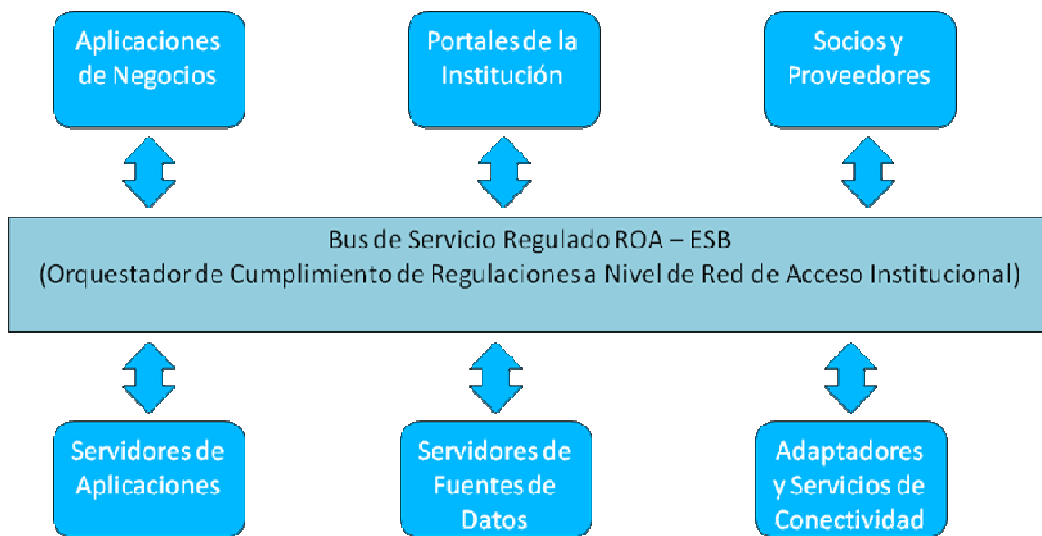


Figura 1.5: Esquema de un Escenario Bus de Servicio Regulado ROA-ESB

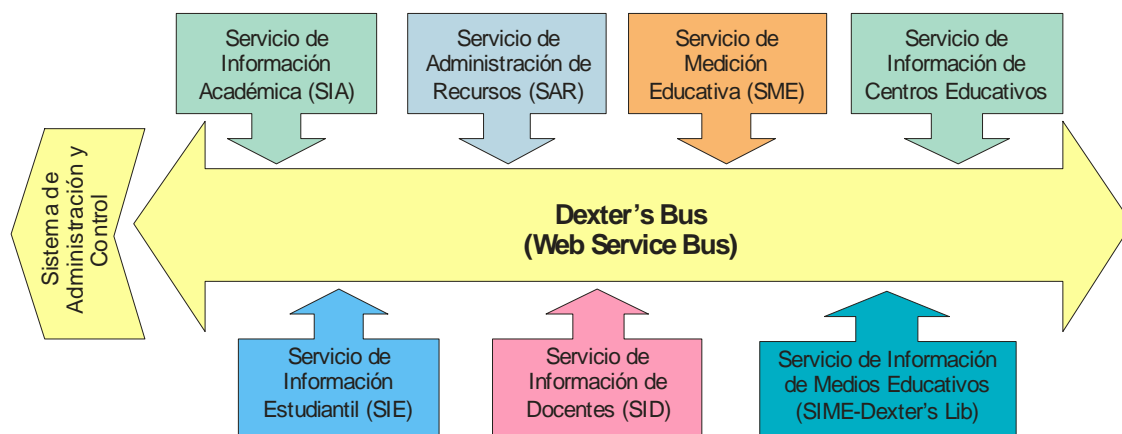


Figura 1.6: Plataforma de Bus Educativo Dexter's Lib

Está claro que dicha solución tecnológica puede ser adaptada a otros entornos: llámese empresarial o telecomunicaciones, en donde el factor de integración juega un rol decisivo.

Si suponemos un panorama donde la mayoría de colegios estuvieran conectados a Internet e intentando compartir información; podríamos dar revista a ciertos problemas que se presentarían:

- Diversidad de proveedores de sistemas de información. Esto significa que muchas compañías podrían desarrollar aplicaciones a medida para cubrir ciertas necesidades de información en las escuelas y en la administración

- educativa; tales como Sistemas de Evaluación Educativa, Sistemas de Bibliotecas, Librerías multimedia en línea, etc. Esto en realidad es bueno pues fomenta la competencia; pero se convierte en un desastre si no hay una clara estandarización en los formatos de información. Una empresa predominante podría someter a otras imponiendo su propio formato, e impidiendo la libre participación de otras empresas en el mercado.
- Diversidad de Plataformas y Modelos de Programación. Este problema se deriva del anterior. Cada proveedor de sistemas de información es libre de generar sus aplicaciones bajo una determinada plataforma soportada por cada centro educativo; como por ejemplo Linux, Windows, o UNIX y escribirlo en un lenguaje determinado, digamos como: Visual Basic, Java, C++, o Cobol.
 - Baja cooperación entre sistemas de información. Si cada aplicación que se desarrollara estuviera concebida como autosuficiente, entonces se sacrificaría el verdadero valor de Internet que es el intercambio de información entre sistemas. Se llegaría a un punto en donde habría mucha información idéntica replicada en múltiples bases de datos, dando lugar a continuos problemas de actualización de datos.
 - Ineficiente uso de los recursos informáticos. Una aplicación que pretenda hacerlo todo derivaría en una alta demanda por sus servicios, lo que se traduciría en un alto costo de uso de la red y sobrecarga del servidor donde reside la aplicación.

1.3. Antecedentes

Este tema de Tesis es fruto de una investigación iniciada en noviembre de 2002, lográndose en ese entonces los siguientes avances más representativos:

- Se revisaron artículos y publicaciones relacionadas con el tema. Se compararon los Web Services con otras alternativas de integración como *CORBA (Common Object Request Broker Architecture)*, que se centra en el protocolo llamado *Internet Inter-ORB Protocol (IIOP)* para objetos remotos; *DCOM (Distributed Component Object Model)* que soporta

objetos remotos que se ejecutan sobre un protocolo llamado: Llamada de Procedimiento Remoto de Objetos ORPC; Java/RMI (*Remote Method Invocation*), que se basa en un protocolo llamado *Protocolo de Método Remoto Java (JRMP)*, quedando fortalecido el interés por ahondar en Web Services. Se compararon alternativas de implementación de Web Services, entre ellas: J2EE (Sun Microsystems) y .NET (Microsoft), resultando J2EE como plataforma preferida de desarrollo.

- Se revisaron un conjunto de estándares XML para el intercambio de información educativa, entre ellas: IMS (Instructional Management Systems) y OASIS Education XML TC (Comité Técnico de XML en la educación). También se investigaron iniciativas sobre integración de sistemas educativos como SIF (Schools Interoperability Framework).
- Se investigó a fondo el conjunto de tecnologías J2EE como JSP/Servlets y EJB. Se logró realizar prototipos de implementación de Web Services usando herramientas de Software Libre como: Apache para dar soporte al servidor web; Axis para dar soporte SOAP; JBoss para dar soporte a los EJB; Tomcat para dar soporte en la capa de presentación a los JSP y Servlets; MySQL como servidor de la base de datos, y; Windows como posible sistema operativo.
- Se investigaron a fondo las técnicas de ingeniería de software con énfasis en los patrones de diseño, a través del framework Struts y arquitecturas de construcción de plataformas empresariales orientadas a modelos.

1.4. Solución al problema

En este trabajo de tesis se propone una alternativa de solución al problema de la integración en el sector educativo; en concreto, de los servicios de biblioteca a nivel nacional, utilizando para ello un Sistema de Gestión de Procesos BPMS sobre plataforma de Código Abierto Intalio. Si bien es cierto se plantea un prototipo para una aplicación puntual de gestión de bibliotecas como una propuesta de mejora al proyecto de educación denominado Huascarán; este mismo prototipo puede acondicionarse a otros proyectos educativos de esta

índole.

Por su capacidad de integración se han escogido, para implementar la aplicación de la Biblioteca Digital Distribuida, sistemas distribuidos heterogéneos basados en Software Libre, de manera que su desarrollo pueda evolucionar en el tiempo. Es decir, obtener una aplicación a bajo costo y alta calidad en base a las características de la arquitectura seleccionada.

Las características intrínsecas necesarias para validar un lenguaje de proceso son:

1. Sintaxis con alta capacidad de expresión
2. La garantía de que no haya ambigüedad
3. Nivel de detalle por resolución, desde niveles macros hacia niveles granulares
4. Sintaxis que pueda ser compartido por diversos actores dentro de la organización
5. Sintaxis que pueda ser utilizado por las personas para procesar las aplicaciones, permitiendo manejar la complejidad de los procesos

Lenguaje	Descripción	Detalle
BPEL	Lenguaje de Ejecución de Procesos de Negocio	Orientado a la ejecución de procesos Aún no tiene una sintaxis gráfica estándar
XPDL	Lenguaje de Definición de Procesos XML	Lenguaje orientado a la descripción de los procesos. Permite intercambio entre diferentes herramientas. Sin semántica definida. Soporta BPMN
jBPM	Lenguaje de Modelado de Procesos de negocios Java	Soporta la ejecución de BPEL Pensado como lenguaje intermedio para implementar sobre él otros lenguajes de procesos
BPMN	Notación de Modelado de Procesos de Negocio	Lenguaje visual estándar, incluye una traducción hacia BPEL Se ha convertido en un estándar de facto

		Existen herramientas de código abierto en el mercado (tal como Intalio Designer)
--	--	--

Cuadro 1.2: Lenguajes de Procesos más importantes

En BPMN los procesos de negocio involucran capturar la secuencia de las actividades en forma ordenada. Los niveles de modelado de procesos son:

Mapas de Procesos—Simples diagramas de flujo de las actividades.

Descripción de Procesos—Proporcionan información más extensa acerca del proceso, como las personas involucradas en llevarlo a cabo (roles), los datos, información, etc.

Modelos de Procesos—Diagramas de flujo detallados, con suficiente información como para poder analizar el proceso y simularlo.

El modelado de procesos con BPMN está dirigido a los Analistas de Negocios y a los Diseñadores de Procesos; mientras que la ejecución de los procesos con BPEL está dirigida a los Arquitectos de Software e Ingenieros de Software.

Las principales características del BPMN son:

1. Refleja la experiencia y las mejores prácticas de la comunidad de modelados de negocios.
2. Sólo incluye elementos fundamentales para modelar procesos de negocio, no maneja:
 - a. Estructura Organizacional y Recursos
 - b. Modelos de Datos
 - c. Reglas de Negocios
 - d. Inteligencia de Negocios
3. Define reglas claras de extensibilidad.
4. Maneja diversos niveles de abstracción desde macro niveles hacia grano niveles.

En la arquitectura ROA, el BPMN posee una semántica enmarcada dentro del contexto organizacional, tal como se describe en el siguiente gráfico:

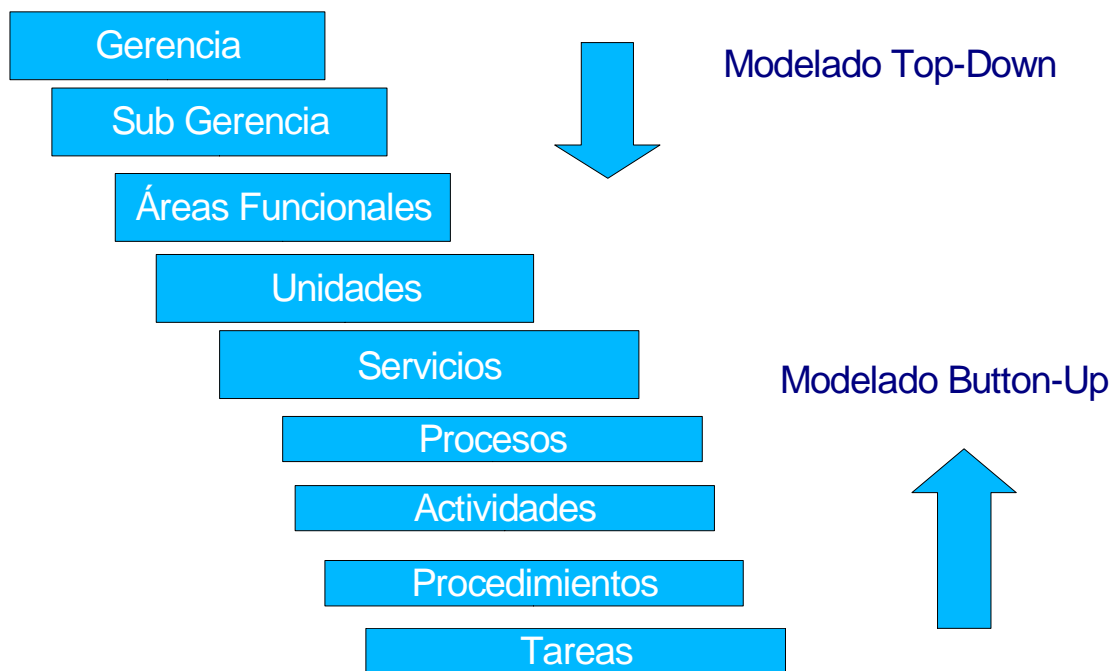


Figura 1.7: Paradigma de enfoque centralizado ROA-BPM

A continuación se muestra una tabla que resume la semántica ontológica del lenguaje BPMN:

Léxico de la Semántica Ontológica del Lenguaje BPMN		
Léxico natural	Descripción	Léxico BPMN
En qué orden	Se define la secuencia luego de haber realizado el análisis	Flujo de Secuencia Asociaciones
Cuándo	Cada qué intervalo se produce el proceso	Eventos Compuertas
Qué recursos necesita	Se obtiene a partir del documento de requerimientos funcionales	Objetos de Datos Asociaciones
Qué recursos produce	Crea un documento de revisión de la arquitectura	Flujo de Mensajes Actividades Objetos de Datos
Quién lo hace	Un arquitecto de software define los roles y responsabilidades	Flujo de Mensajes Actividades
Qué reacción produce	Define el flujo de escalamiento del proceso	Objetos de Datos Grupos Anotaciones

Hacer qué	Revisa la arquitectura final	Pools Lanes

Cuadro 1.3: Resumen de la semántica ontológica BPMN

Entonces: podemos concluir que los sistemas de gestión de procesos BPMS soportan los procesos educativos. Algunos de estos procesos se encaminan hacia sistemas de e-learning. Dentro de la industria del e-learning existen cuatro organizaciones que han creado estándares a nivel mundial: ADL, AICC, IEEE e IMS [ver referencia 28]

ADL: *Advanced Distributed Learning.* Es un organismo que investiga y desarrolla especificaciones para motivar la adopción y el avance del e-learning; su propósito es ayudar a que los materiales de aprendizaje se adecuen a las necesidades de entrenamiento y que estos estén disponibles de manera general. El estándar propuesto por ADL se llama SCORM (Modelo de Referencia de Objetos de Contenido Compartido).

AICC: *Comité de CBT de la Industria de Aviación.* Es una de las organizaciones más antiguas en la proposición de estándares de e-learning. Desde 1988 se plantearon lineamientos para el desarrollo de sistemas para la aviación, pionera en el entrenamiento basado en simuladores y tecnología. AICC propone un entrenamiento efectivo en costos, eficiente y sostenible; para esto publica una serie de recomendaciones que incluyen configuraciones de hardware y software.

El CMI (Instrucción Administrada por Computadora). Plantea lineamientos para el desarrollo de contenido que podría comunicarse con la mayoría de los sistemas de administración del aprendizaje (Learning Management Systems) utilizando para ello una notación de modelado de procesos de negocios empresariales BPMN, de modo que cualquier modificación en el flujo del proceso, éste se realice en forma inmediata.

IEEE: El Comité de estándares de tecnologías del aprendizaje del Instituto de Ingenieros Eléctricos y Electrónicos también ha desarrollado recomendaciones y estándares técnicos que recalcan las mejores prácticas. El estándar de uso más extendido de esta organización es LOM (Metadatos de Objetos de Aprendizaje), que define las características de los elementos y estructuras de aprendizaje.

IMS: El consorcio IMS reúne a vendedores e implantadores que se enfocan en el desarrollo de especificaciones basadas en XML y describen las características clave de cursos, lecciones, evaluaciones, aprendices y grupos. Los estándares más importantes de IMS son:

- **IMS Meta Data**, que plantea recursos para el procesamiento de datos de aprendizaje.
- **IMS Content Parking**, que describe la manera de empaquetar contenidos.
- **IMS QTI**, que describe formas de intercambio de preguntas y evaluaciones.

Existen tres aspectos principales en los estándares e-Learning acerca de los datos del proceso:

1. La interfaz de comunicación, es decir, la forma en la que los recursos pueden comunicarse con otros sistemas; y esto ha sido recalado por los estándares de AICC y ADL.
2. Los metadatos, que son la descripción consistente de recursos entre aplicaciones, lo cual ha sido enfatizado por el estándar LOM de IEEE, así como por SCORM de AICC.
3. El empaquetamiento, o sea como organizar los recursos en grupos útiles,

enfanzado por los estándares de IMS, así como por el estándar SCORM de AICC.

Desde que la solución planteada se basa en el uso de componentes (es decir se pueden ir agregando módulos sin afectar a los ya existentes basados en posteriores requerimientos) y además emplea XML; esto nos permite cumplir con los estándares mencionados, a la vez que nos brinda ventajas en concreto:

- Permite funcionalidad Web semántica modelando metadata (entorno de descripción del recurso, para describir mejor los contenidos y las relaciones entre datos) e inteligencia hacia la web.
- Los documentos XML pueden ser estructurados para identificar cada pieza importante de información, así como las relaciones que existen entre ellas, es decir, es posible escribir código que puede procesar documentos XML sin intervención humana.
- Simplifica el intercambio de datos debido a que diferentes organizaciones, o aún partes de la misma organización, raramente usan un mismo conjunto de herramientas tomando bastante tiempo para poder comunicarse entre sí.
- Otra ventaja, es que los vendedores de software ya proveen herramientas para transformar sus archivos de la base de datos: directorios LDAP, u órdenes de compra, hacia y desde XML.
- Permite búsquedas inteligentes; un caso puntual es que Oracle 9i R2 provee BD XML nativo.

En esto es importante distinguir el entorno de las aplicaciones de los Web Services, empleando SOAP, teniendo como medio de transporte HTTP y XML desde el punto de vista de la arquitectura de software, donde sobresalen dos tipos de arquitecturas:

1. POP: Presentation-oriented publishing

2. MOM: Message-oriented middleware

En el esquema POP, la información estructurada es presentada en varios tipos de medios (papel, pantallas, mobiles, etc.); mientras que en el esquema MOM la nueva arquitectura esta basada en un entorno cliente/servidor, donde el software middleware maneja la base de datos (servidor) y envía la data en XML al cliente. En este esquema algunos clientes pueden leer diferentes bases de datos mientras estos estén en XML.

La esencia de los estándares se enfoca en las posibilidades de comunicación de contenidos, los metadatos y el empaquetamiento; por ello en el presente proyecto se toma como referencia estas premisas para resolver diversos problemas a través de los Web Services.

Un panorama alternativo con el uso de esta familia de tecnologías sería:

- Diversidad de proveedores de sistemas de información. Varias compañías podrían ofrecer sus productos informáticos de manera que ofrezcan más valor para su mercado, pero siempre rigiéndose bajo estándares que permitan el intercambio de información con otros sistemas. De esta manera se promueve una competencia en igualdad de condiciones a través de la versatilidad de la arquitectura.
- Diversidad de plataformas y modelos de programación. Cada proveedor de sistemas de información estaría libre de crear sus productos bajo la plataforma y modelos de programación que le resulte adecuado, ya que tendría la seguridad que su sistema podrá cooperar con otros sistemas creados bajo otra plataforma y modelo de programación.
- Alta cooperación entre sistemas de información. Este nuevo esquema de integración promueve el desarrollo de sistemas de información

cooperativos fomentando el intercambio de información. Es decir, permite diseñar sistemas de información como si se tratasen de Servicios (Arquitectura Basada en Servicios), ofreciendo el soporte necesario a otros sistemas para cumplir con un determinado proceso que dé valor al usuario. De esta manera, se reduciría la duplicación de datos entre múltiples sistemas de información. A manera de ejemplo, un Servicio de Información Estudiantil (SIE) se encargaría de la administración de datos básicos de estudiantes, tales como código, nombres, dirección, familiares; mientras que un Servicio de Biblioteca (SB) se encargaría de la administración de datos de una biblioteca, tales como códigos de libro, títulos, autores, ejemplares; el SB entonces utilizaría los servicios del Servicio de Información Estudiantil para completar el proceso de préstamos de libros hacia los alumnos.

- Eficiente uso de los recursos informáticos. Web Services son un nuevo paradigma en el desarrollo de Sistemas Distribuidos. Su efectivo uso permitirá generar aplicaciones que residan en distintos lugares y servidores, balanceando la carga de uso del sistema y dando la sensación que se está usando un único gran sistema.
- Colaboración en la administración del contenido. Múltiples usuarios podrían colaborar en un único contenido con controles de versiones sobre las publicaciones del material educativo, permitiendo involucrar a los actores del sistema.
- Esquema educativo multicanal. Convirtiendo el contenido a XML este mismo contenido podría ser publicado en varios formatos para diferentes usos tales como la web, dispositivos móviles, documentos pdf, etc.

1.5. Objetivos de la tesis

El prototipo propuesto es un Sistema de Información de Bibliotecas Digitales Distribuidas basado en la propuesta de una Arquitectura Orientada a Regulaciones, que propone una manera distinta de desarrollar un sistema de información orientado al manejo de procesos. Para efectos de validación sólo se ha desarrollado el escenario de la entidad que representa al Registro de Usuario Bibliográfico, lo cual nos ayudará a probar la viabilidad técnica de la solución cuando se requiera adecuarla a estándares internacionales.

El prototipo pretende ser un sistema de información alternativo a los sistemas de bibliotecas tradicionales, ya que será distribuido e interoperable con otras aplicaciones, pero de todas formas se necesita un desarrollo adicional para adecuar la integración en un contexto local o nacional; debido a que cada institución educativa posee sus propias peculiaridades. En él se presentan nuevos conceptos de interoperabilidad, entre sistemas multiplataforma, a través del uso de estándares de Internet (BPMN, BPEL, XSD, XML, Ajax) y tecnologías como *Web Services*. Estas características fundamentales permitirán crear un entorno que estimule y promueva la comunicación entre lectores, autores, empresas editoras y bibliotecas a nivel local y nacional; pudiendo ser extendida a un entorno internacional. Asimismo, es importante recalcar que se puede extender esta solución tecnológica para solucionar problemas similares en otros ámbitos de la educación peruana como el de la capacitación docente, evaluaciones, etc. En esto último, el hecho que la base de datos sea distribuida y modelada a través de objetos mediante el lenguaje SQL nos permite una transparencia del tipo de la base de datos a emplearse; así como la utilización óptima de recursos tecnológicos por zonas geográficas. Sin embargo, cabe recalcar que la arquitectura ROA es más ligera y flexible y permite adecuar estándares para un proyecto en particular. A todo ello se debe sumar la complejidad de la tecnología, pues requiere un conocimiento profundo en técnicas de ingeniería de software.

Objetivos Generales

- Propuesta de una Arquitectura Orientada a Regulaciones para el Desarrollo de Sistemas de Información orientada al Sector Educación.
- Diseñar e implementar un prototipo ROA de una Biblioteca Digital Distribuida que demuestre el uso de Web Services en el sector educación, y la posible mejora que podría generar al Programa Educativo Peruano.

Objetivos Específicos

- Definir a detalle los componentes de la Arquitectura Orientada a Regulaciones; enfatizando en las Metodologías de Dirección, Tecnologías de Soporte, Infraestructuras de Regulación, así como las líneas de Desarrollo.
- Revisar los estándares de formatos para el intercambio de información educativa para su uso efectivo y contextualizado en el sistema educativo peruano.
- Entender la tecnología de Web Services bajo la plataforma BPMN Intalio y su arquitectura interna. Contrastarla con otras alternativas de integración.
- Proponer el uso de una plataforma tecnológica de un Sistema de Información basada en Web Services en el Sistema Educativo, y en especial como una continuación del Programa Huascarán; estableciendo las ventajas y desventajas con dicha solución.
- Facilitar la apertura de canales de comunicación entre los potenciales interesados en el desarrollo de este proyecto, entre ellos: el Ministerio de Educación, el Programa Huascarán como referente, y la Comunidad

Educativa en general; de manera que los frutos de esta investigación sea coherente con las necesidades y los recursos disponibles, además de permitir derivar los requerimientos para la aplicación seleccionada.

- Esbozar la propuesta de una plataforma de desarrollo de servicio de información de material educativo digital denominada Dexter Lib, que brinde facilidad de acceso, versatilidad en las búsquedas, entre otras características.
- Analizar la viabilidad técnica de integrar los distintos sistemas asociados al consumo y al suministro de material educativo que permita la interconexión educativa a nivel Nacional.

Con la finalidad de restringir el alcance del prototipo de la Biblioteca Digital Distribuida que he denominado Dexter Lib, a continuación se detallan los considerandos que no cubre el presente desarrollo:

- En el presente proyecto de Tesis, se desarrolla el Sistema de Información de Bibliotecas Digitales Distribuidas del Servicio de Información de Medios Educativos (SIME – Dexter'sLib) sobre BPMN, a efectos de validar un prototipo de prueba de concepto.
- No se pretende cubrir todos los casos de uso de un sistema de biblioteca, solo se implementan los casos de uso de registro de usuario de biblioteca sobre BPMN.
- No se pretende construir un servidor de procesos BPMS, por el contrario se utiliza el servidor Intalio-JBoss para orientar el desarrollo a la parte del dominio del problema, y no dedicar tiempo a la complejidad de la arquitectura BPEL-J2EE; más bien a definir el flujo del proceso mismo.

- No se pretende solucionar todos los problemas de integración en el sector educativo, sólo se desea demostrar la viabilidad técnica, así como las ventajas y desventajas de utilizar BPMN.
- No pretende reemplazar lo existente del Programa Huascarán (que se ha limitado a desplegar una infraestructura de hardware con un portal centralizado), sino que se postula una propuesta de creación de una plataforma de software complementaria relacionado a un Sistema de Información que permita la integración sobre sistemas distribuidos en el sector educación; dando como ejemplo el prototipo Dexters Lib propuesto.

1.6. Organización de la tesis

El capítulo 2 es una discusión sobre la informática educativa en el País; se comenta la implicación de las Aplicaciones Telemáticas para innovar metodologías de enseñanza aprendizaje, así como la evolución y crecimiento de los sistemas de información desde los Web Services hasta los Sistemas de Gestión de Modelado de Procesos de Negocios, con ejemplos concretos en el sector educativo así como de las telecomunicaciones. Se presenta también la propuesta de una Arquitectura Orientada a Regulaciones ROA utilizando una variante del método unificado racional para el desarrollo de software, que incluye las técnicas de patrones de procesos y su integración con metodologías GTZ, Seis Sigma y PMBOK. En el capítulo 3 se propone y sustenta el diseño de un prototipo de una biblioteca digital distribuida sobre ROA para demostrar los conceptos propuestos, asimismo se describe el diseño usando BPMN poniendo énfasis en la documentación del proyecto.

En el capítulo 4 se revisan y validan los modelos de componentes construidos con resultados analíticos previamente presentados. El estudio está basado en resultados de simulación a nivel de usuario final y en explicaciones intuitivas de las observaciones. Finalmente, debido a lo reciente del tema, actualmente existe poca investigación nacional sobre la implementación de web services en el

sector educativo sobre BPMN y BPEL. Con la finalidad de aportar conocimiento en este campo, se analiza el posible efecto que tendría el rendimiento del modelo de web services a escala nacional, y los resultados se comparan y analizan con un breve resumen en el capítulo 5 con resultados relevantes y recomendaciones para futuros trabajos de investigación. Asimismo, en los apéndices se detallan aspectos complementarios del prototipo, así como del proyecto Huascarán, y un glosario de terminologías utilizadas.

Capítulo 2

Consideraciones preliminares

En este capítulo se presentan alcances del proyecto, se enuncian formalmente los conceptos de la arquitectura orientada a regulaciones ROA, las tecnologías de los sistemas distribuidos y se discuten algunas metodologías de construcción de software. ROA permite dotar de una arquitectura integral sobre la cual se desarrolla un sistema de información, es decir, permite recoger, procesar, almacenar, analizar y difundir información para cumplir un propósito definido por una determinada organización. La ventaja de utilizar ROA es que permite adaptar, para cada escenario, las mejores prácticas tanto a nivel metodológico como a nivel tecnológico indistinto de los fabricantes. Como prueba de concepto vamos a desarrollar un entorno ROA para diversos escenarios de configuración.

En el siguiente cuadro se muestran algunos perfiles validados:

Perfil ROA	Metodologías de Dirección	Tecnologías de Soporte	Infraestructuras de Regulación	Líneas de Desarrollo	Tipo de Organización
ROA Básico	PMBOOK RUP-BPM	BPMS	ITIL NTP	Automatización y Monitoreo de Procesos	PYMES, Gobiernos Locales
ROA Intermedio	PMBOOK RUP-BPM 6 SIGMA	BPMS BI BR	ITIL NTP CMMI	Automatización de Procesos, Inteligencia de Negocios, Portal	Empresas, Gobiernos Regionales
ROA Avanzado	PMBOOK RUP-BPM 6 SIGMA GTZ	BPMS BI BR ESB	ITIL CMMI COBIT ISO	Bus de Servicio Empresarial SOA	Grandes Empresas, Gobierno Central

Cuadro 2.1: Tabla de Perfiles ROA

En la Figura 2.1 se resume la forma en que interaccionan los tres cometidos.

La ROA cubre las capas arquitectónicas mínimas que se necesita como pre requisito para ser concebida como una supra arquitectura que contenga a las arquitecturas convencionales.

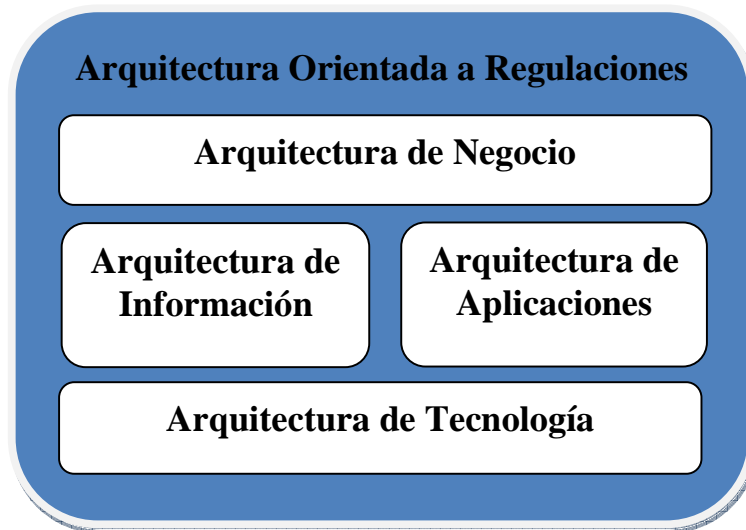


Figura 2.1: Macro Esquemas de la Arquitectura Orientada a Regulaciones ROA

La ROA es adaptable en el tiempo en cuanto a las arquitecturas de tecnologías se refiere; actualmente se utiliza bastante SOA a este nivel. En el siguiente gráfico se muestra a detalle el ciclo de vida SOA:

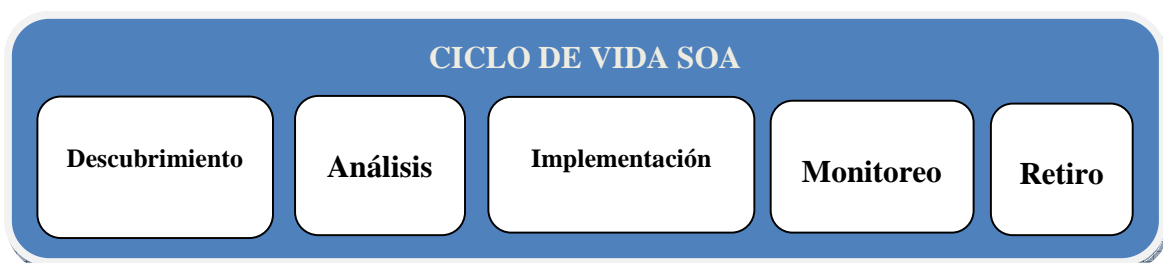


Figura 2.2: Ciclo de Vida Tradicional SOA

Descubrimiento:

- Se define los requerimientos
- Se identifican los procesos de negocios

Análisis:

- Identificación de servicios e interfaces
- Modelado de procesos

Implementación:

- Implementación del servicio
- Implementación del proceso
- Pruebas

Monitoreo:

- Control de configuraciones y versionamiento
- Configuración del sistema y el servicio
- Registro de servicio

Retiro:

- Modificar servicios y procesos dependientes

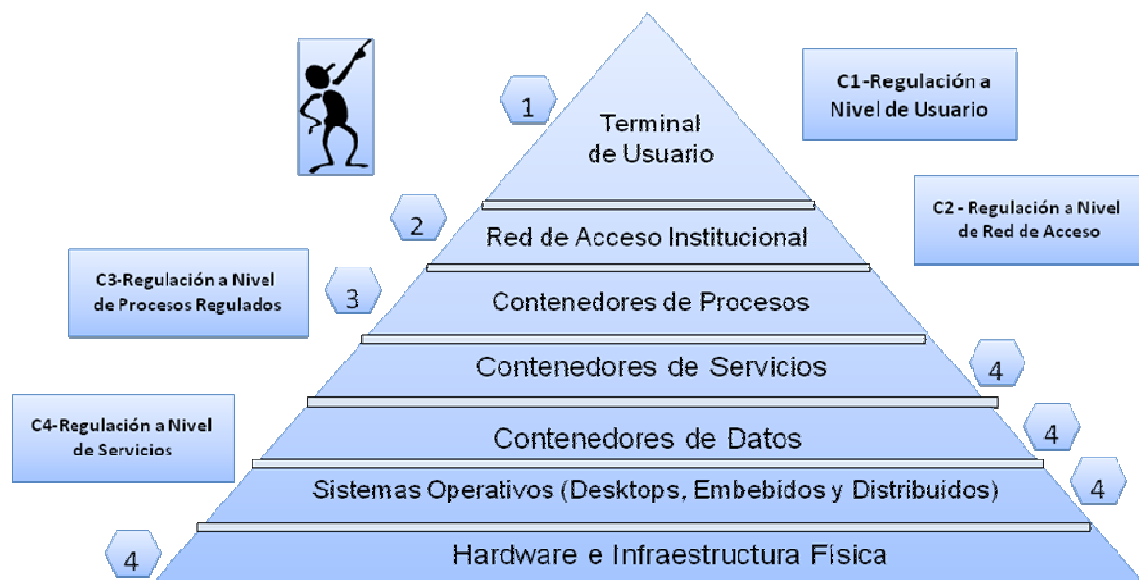


Figura 2.3: Pirámide de Seguridad Telemática ROA

Un aspecto importante de ROA es que su capa de nivel de seguridad está basada en cuatro capas y subdividida en 7 niveles:

- Capa 1: Regulación de Seguridad a nivel de Usuario
- Capa 2: Regulación de Seguridad a nivel de Red de Acceso
- Capa 3: Regulación de Seguridad a nivel de Proceso Regulado
- Capa 4: Regulación de Seguridad a nivel de Servicio

En la figura 2.3, se muestra la pirámide de seguridad telemática ROA. Debido a que la normativa peruana vigente en el sector de gobierno da prioridad a las soluciones de TI orientado a plataformas de código abierto; se ha desarrollado la ROA utilizando soluciones Open Source. La aplicación de la pirámide de seguridad de siete niveles a un escenario de Código Abierto se muestra a continuación:

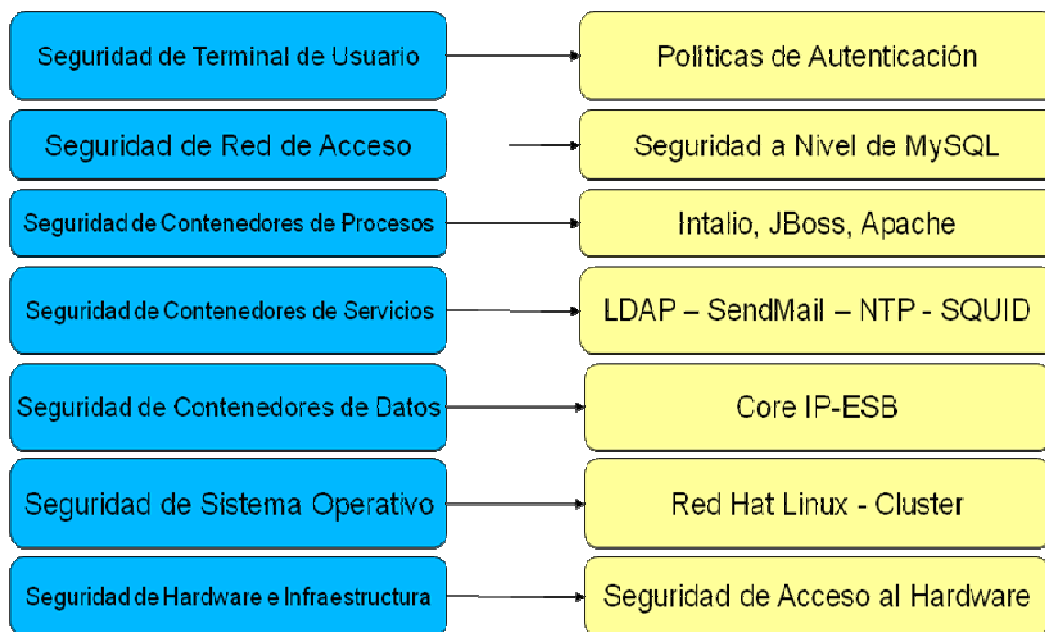


Figura 2.4: Ejemplo de un Escenario de Seguridad ROA para una Plataforma de Código Abierto

Debido a que la ROA es independiente del lenguaje de programación a utilizar, se ha escogido una configuración basada en un bus de servicio empresarial con tecnologías de código abierto.

2.1. La Informática Educativa en el País

Recientemente la informática está incidiendo en la formación educativa en el país, por ejemplo, a través del Programa Huascarán. En el marco de la interculturalidad, de acuerdo a normas y estándares internacionales, es imprescindible facilitar el cambio cualitativo en la forma de educar; permitiendo la construcción del conocimiento significativo e incorporando los valores postulados en la política educativa peruana a través de las tecnologías de la información.

En el Programa Huascarán: las tecnologías de la información y los magnéticos, fotónicos y digitales facilitan el desarrollo de un modelo pedagógico que permite a la comunidad educativa el acceso a más y mejor educación, en todo el territorio nacional, todo el tiempo. [ver referencia 19]

2.1.1. La Educación Virtual

Existen diferentes concepciones de enseñanza a distancia, entre las que se encuentra la enseñanza virtual, también denominada e-learning, on-line learning, formación on line, formación por Internet; donde estamos ante una forma de enseñanza a distancia con un uso predominante de Internet como medio tecnológico.

Otras concepciones de enseñanza a distancia más tradicionales, son la educación por correspondencia, la clase a distancia, la tele formación o la enseñanza semi presencial.

La educación virtual se lleva a cabo a través de las llamadas plataformas de difusión o gestión del conocimiento, también denominadas en inglés Learning Management System (LMS), las cuales consisten en un paquete integrado de software alojado en un servidor, al cual se accede desde los navegadores de Internet convencionales sin que el usuario deba instalar en su ordenador ningún

programa, y que incluyen todas las herramientas necesarias para ofrecer cursos a través de Internet o de una Intranet. La educación virtual está siendo cada vez más completa y sofisticada y son ya muchos los elementos que necesitan instrucción por parte de los expertos y que han venido a enriquecer sobremanera las posibilidades de esta modalidad de aprendizaje, como puede ser, por ejemplo, el libro electrónico o eBook que por sus enormes posibilidades didácticas, se está abriendo camino con fuerza en este nuevo escenario de enseñar y aprender. [ver referencia 20]

El término biblioteca digital es un concepto que ha sido manejado desde que se desarrolló la primera computadora, una biblioteca computarizada que complementa, añade, e inclusive, reemplaza a las bibliotecas tradicionales. En un contexto internacional, podría decirse que el término de Biblioteca Digital tomó énfasis en el año 1994 en cuya fecha se estableció una iniciativa de Bibliotecas Digitales por la Fundación Nacional de Ciencias, DARPA y NASA en los Estados Unidos, basados en el inesperado crecimiento del Internet.

Dentro de la Comunidad de Bibliotecas Digitales se han desarrollado con el tiempo diversas áreas de investigación, siendo tres de estas áreas las más relevantes:

- ◆ Repositorios digitales
- ◆ Museos digitales
- ◆ Manejo de activos digitales

La diferencia sustantiva entre estas áreas de investigación, dentro de la comunidad de Bibliotecas Digitales, radica en lo siguiente:

Los repositorios digitales han evolucionado: de ser lugares de almacenamiento de equipos hacia una organización responsable que mantiene el contenido digital. Estos repositorios digitales son sitios activos, dinámicos y colaborativos, como un sitio de proyecto de software de fuente abierta. Otros, tienen una inclinación distinta, más histórica, viéndolo como lugares donde el

contenido digitalizado físico y análogo y el contenido digital original son preservados por siempre para la humanidad.

Los museos digitales que en esencia a través de imágenes 3-D, con voz y video en tiempo real inclusive, permiten captar las grandes esculturas o centros históricos o arqueológicos del mundo; dando las facilidades de la ubicuidad digital.

El manejo de los activos digitales está orientado a definir arquitecturas de referencia end-to-end para almacenar datos y la forma de transformar estos datos a cualquier tipo de medio; permitiendo entregar un activo digital a cualquier cliente o usuario, ya sea interno o externo, en cualquier formato y en cualquier momento.

Comparación de plataformas de aprendizaje virtual

Las siguientes plataformas son utilizadas en Internet con propósitos educativos. El aprendizaje basado en Internet utiliza múltiples opciones en plataformas, estas opciones son:

CD ROM: Abreviación de Compact Disk Read-Only Memory, es un CD-ROM que sirve para almacenar datos digitales y puede contener 650 Mb o 720 Mb. Son programas de información independientes no conectados a Internet u otros sistemas de comunicación.

Sitios Web: Son páginas virtuales enlazadas en Internet o en la intranet. Pueden ser comparadas a un manual de referencia o con un libro. Suministran información pasiva.

Comunicación Asíncrona por Internet (AIC): Se refiere a las herramientas comunicacionales propias de Internet, como el correo electrónico y el foro que usualmente tienen una base de datos u otro medio de almacenamiento de información accesible a los estudiantes e instructores. Los usuarios pueden escribirse entre ellos a destiempo.

Comunicación Síncrona por Internet (SIC): Es una plataforma que permite la comunicación en tiempo real entre los participantes a través de chats, video conferencia y voice chat. El Chat es un sitio virtual común para todos los participantes en donde los mensajes mandados son expuestos para todos. Cada persona puede mandar su propio mensaje.

Aprendizaje Basado en la Web (WBT): Es una plataforma de aprendizaje en línea que contiene herramientas de comunicación y gerencia de curso en una Intranet o en Internet y puede combinar cualquiera de las plataformas anteriores.

Híbridos: Son cualquier combinación de las plataformas anteriores para la enseñanza persona a persona en un aula de clases, adiestramiento o facilitación de grupo.

Ventajas y Desventajas del Aprendizaje Virtual

Son muchas las ventajas que ofrece; las más importantes son las siguientes:

- Formación disponible las 24 horas y 365 días del año en cualquier lugar
- Disminución de los costos de desplazamiento y de tiempo
- Disminución de los costos de formación
- Comunicación en tiempo real y asíncrono
- Proporciona mayor autonomía geográfica y temporal
- Facilidad de realizar cursos ofertados por universidades de todo el mundo
- Oportunidad de trabajar a su propio ritmo y de autodisciplinarse
- Soluciones de formación inmediatas
- Aplicable a cualquier área de conocimiento
- Múltiples herramientas de comunicación e interacción
- Desarrollo del aprendizaje colaborativo

Las dos desventajas principales son las siguientes:

- Todavía existen muchas personas que no conocen las nuevas tecnologías,

y se resisten a hacerlo

- Limitaciones tecnológicas, tanto de equipos como de las conexiones a Internet

2.1.2. Desarrollo de Software Educativo mediante la Ingeniería de Software

Según la definición del IEEE, **software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo. Un producto de software es un producto diseñado para un usuario.** En este contexto, la Ingeniería de Software (SE del inglés Software Engineering) es un enfoque sistémico del desarrollo, operación, mantenimiento y retiro del software.

El proceso de ingeniería de software se define como **un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad.** El proceso de desarrollo de software **es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código; el código es probado, documentado y certificado para su uso operativo.** Concretamente **define: quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.** [ver referencia 17]

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios. La ingeniería de software concierne un conjunto de actividades de concepción y

de puesta en aplicación de los productos y procedimientos para racionalizar la producción de software y su mantenimiento.

Se preocupa de la realización, de la gestión, del mantenimiento, de la calidad y de la utilización de un producto de programación por un número importante de personas y para una larga duración (múltiples versiones).

Es una necesidad debida a la poca fiabilidad del software y a las grandes dificultades de producir un producto que satisfaga condiciones de contrato a su debido tiempo.

La ingeniería de software se interesa en:

- Diferentes teorías
- Metodologías (técnicas y métodos)
- Herramientas
- Organización general
- Gestión de proyectos

La ingeniería de software se propone ayudar para:

- Organizar proyectos (planeamiento y comunicación para el trabajo en equipo)
- Pronosticar los costes y los plazos
- Manejar los recursos (humanos, materiales, de software)
- Seguir la evolución del software (mantenimiento, ciclo de vida)
- Documentar, definir y respetar normas

Uno de los aspectos importantes en el Programa Huascarán, es la calidad de software educativo [ver referencia 21]. En el campo educativo suele denominarse software educativo a aquellos programas que permiten cumplir y apoyar funciones educativas. En esta categoría entran tanto los que dan soporte al proceso de enseñanza y aprendizaje (un sistema para enseñar matemáticas, ortografía, contenidos o ciertas habilidades cognitivas) como los que apoyan la administración de procesos educacionales o de investigación (por ejemplo: un sistema que permita manejar un banco de preguntas).

Cuando se hace referencia a calidad de software educativo, se requiere de un producto que satisfaga tanto las expectativas de los docentes como de los usuarios; a un menor costo, libre de defectos y cumpliendo con ciertas especificaciones institucionales y tecnológicas. La calidad del software está determinada no sólo por los aspectos técnicos del producto, sino por el diseño pedagógico y los materiales de soporte. Este último aspecto es uno de los más problemáticos ya que existen pocos programas que ofrezcan un soporte didáctico.

La evaluación de software educativo se ha centrado tradicionalmente en dos momentos:

- Durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen
- Durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa

2.2. Aplicaciones Telemáticas para innovar metodologías de enseñanza - aprendizaje

El nombre Telemática se genera de la palabra TELEcomunicaciones, y la palabra InforMÁTICA. La Telemática es una especialidad de la Ingeniería que se origina en la aplicación conjunta de los campos de la Informática y de las Telecomunicaciones. [ver referencia 22]

Las áreas involucradas en esta disciplina también son conocidas bajo el nombre de Tecnologías de la Información o TIC. En términos globales, la Telemática consiste en agregar inteligencia a los tradicionales sistemas de Telecomunicaciones a partir de aplicaciones en software, o bien, ofrecer servicios informáticos a distancia a través de sistemas de Telecomunicaciones, como, por ejemplo, Internet.

Mundialmente, la Ingeniería Telemática ha adquirido una creciente relevancia debido al desarrollo explosivo de la Internet y su aplicación con fines

educacionales, científicos, comerciales y de servicios.

Esta sección está dedicada a describir cómo las aplicaciones de Ingeniería Telemática podrían innovar las metodologías de enseñanza -aprendizaje, en ese contexto podríamos citar beneficios de la Telemática a los siguientes participantes del sistema:

Estudiantes: Los alumnos y alumnas de inicial, primaria y secundaria se interrelacionan con las Tecnologías de Información y Comunicación (TIC), según sus propias habilidades y necesidades. Ellos exploran, conocen y evalúan las TIC en el contexto de su aprendizaje. Su uso continuo es imprescindible para el proceso de aprendizaje en el mundo de hoy.

Docentes: El personal docente no sólo tiene la oportunidad de conocer y utilizar las TIC para mostrar sus posibilidades a los estudiantes. Puede, además, seleccionar progresivamente sus ventajas y desventajas pedagógicas durante su aplicación en clase. Asimismo, propiciar la investigación y exploración por parte de los estudiantes y fomentar la formación de un entorno TIC de carácter intercultural y competitivo.

Familia: A través del propio centro educativo, los padres y madres de familia tienen ahora la oportunidad de acceder a las TIC y programar encuentros de actualización y divertimento en las formas más variadas. De esta manera cada vez más peruanos y peruanas se benefician de una mayor capacidad de comunicación.

Podrán participar también de la vida económica y social de su comunidad y del país.

Investigadores: A través del uso de las TIC, los investigadores en materia educativa, pueden acceder al entorno virtual para potenciar sus proyectos. Del mismo modo, pueden formar parte de una red que está en constante

actualización en la incorporación de nuevos métodos y herramientas tecnológicas aplicadas a la investigación.

Debido a que el aprendizaje formal institucionalizado es el que da acceso a cualquier nivel de la actividad laboral, mediante graduaciones, certificados o diplomas; estos requisitos controlan monopólicamente el mercado de trabajo, el porvenir, cualidad y calidad de vida de todos los miembros de los que hoy se ha dado en llamar **Sociedad del Conocimiento**. Pero no sólo el leer y escribir y el manejo de la aritmética son suficientes para escapar a la clase de **analfabeto laboral**. Hoy se necesita, no un mínimo de habilidad en computación elemental, sino un conocimiento más comprensivo y entendimiento considerable de la tecnología, sus dimensiones, sus características, su ritmo de crecimiento y generación de efectos; algo casi totalmente ausente todavía en la mayoría de los países. En la actualidad, es indispensable y crucial conocer la complejidad del mundo en el cual no existen barreras posibles para la diseminación del conocimiento y los niveles de expectativas. Ya ni los pueblos, ciudades, naciones o regiones, pueden definir sus propios horizontes sin tener en cuenta la interconectividad planetaria.

Los medios de masa proveen más aprendizaje que el que provee la escuela. Ésta todavía es necesaria para organizar sistemáticamente y con sentido vocacional el aprendizaje para que éste se convierta en conocimiento como herramienta personal para la participación e interacción social. La sociedad de conocimiento requiere como prioridad que sus miembros aprendan cómo aprender, es decir, cómo saber conocer, nuevos saberes. Los conocimientos actuales cambian con velocidades insospechables pero no así las habilidades. Un ingeniero de hace diez años sin actualización permanente se vuelve obsoleto. Lo mismo sucede con un médico, abogado, economista, geólogo, gerente, líder político o sacerdote. Por otra parte la mejor universidad imaginable y con el número mayor de años de enseñanza no puede dar a un estudiante todas las posibilidades de elección.

2.2.1. Notación para Modelado de Procesos de Negocio: BPMN

En las metodologías de dirección ROA se desarrollaron las pautas para modelar los procesos en las etapas de cambio organizacional. De ahí la importancia de describir los procesos de negocios con una notación estándar de modo que la interpretación de un modelo dado quede en manos del modelador.

Los lenguajes de procesos han surgido por la necesidad de pasar del lenguaje natural a la ejecución, control, monitoreo y análisis de procesos.

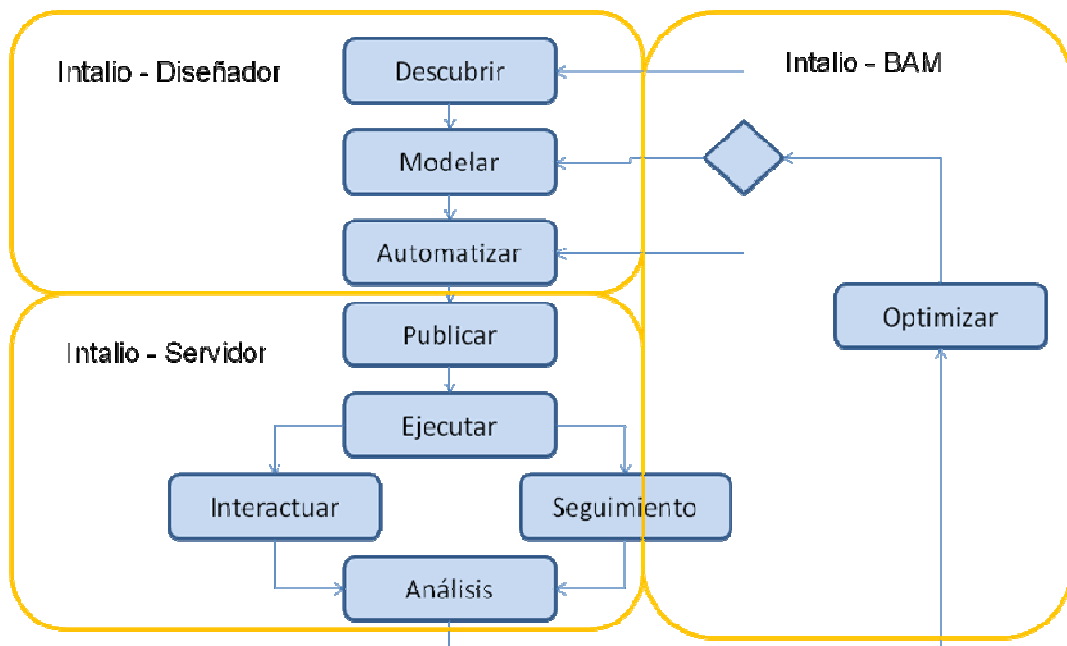


Figura 2.5: Etapas de un Modelado BPM Intalio

Lenguaje	Descripción	Impulsor
BPEL	Lenguaje de Ejecución de Procesos de Negocio	Creado por IBM y Microsoft el 2002
XPDL	Lenguaje de Definición de Procesos XML	Creado por la Coalición de Gestión de Flujos de Trabajo WfMC
jBPM	Lenguaje de Modelado de Procesos de negocios Java	Es el lenguaje creado por JBoss para expresar procesos
BPMN	Notación de Modelado de Procesos de Negocio	Desarrollado por Iniciativa de Gestión de Procesos de Negocios BPMI.org y continuado luego por el Grupo de Gestión de Objetos OMG

Cuadro 2.2: Lenguajes de Procesos

En la arquitectura ROA, el BPMN posee una semántica enmarcado dentro del contexto organizacional tal como se describe en el siguiente gráfico:

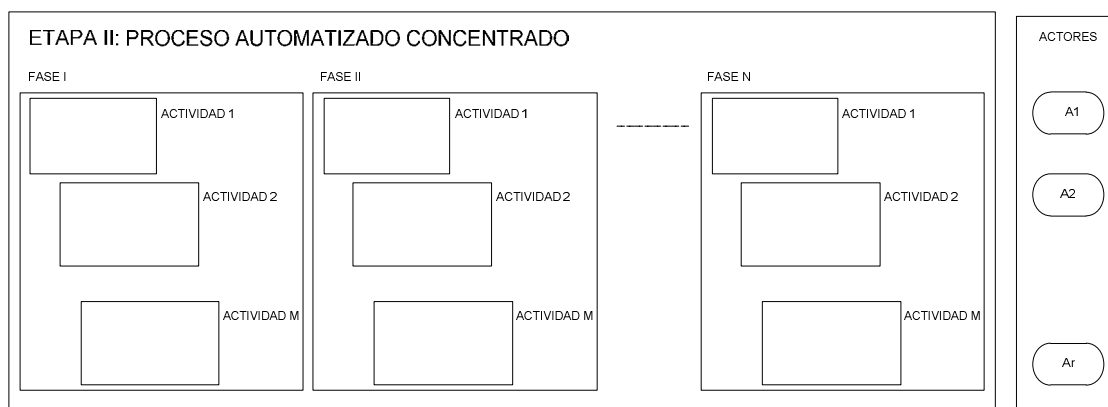


Figura 2.6: Proceso automatizado concentrado bajo el enfoque de Arriba hacia Abajo y de Abajo hacia Arriba

Categorías de Procesos

Desde su descripción, BPMN ha tratado de dar soporte a tres categorías

principales de Procesos:

- Orquestación
- Coreografía
- Colaboración

Se puede resumir que una orquestación existe mediante un proceso de alto nivel que controla a los otros sub procesos. En contrapartida, en una coreografía cada proceso juega un papel independiente y sólo existen interacciones entre los procesos participantes. La colaboración puede contener una coreografía y una o más orquestaciones.

Una orquestación está contenida por un Pool y normalmente en un contexto bien definido. No hay mecanismo central que guíe o mantenga trazo de una coreografía. Por lo cual no hay datos compartidos disponibles para todos los elementos de la coreografía. Para ubicar una coreografía dentro de diagramas BPMN, la forma es hacerlo entre los Pools.

A continuación, se muestra una tabla que resume la semántica ontológica del lenguaje BPMN:

Léxico natural	Léxico BPMN
En qué orden	Flujo de Secuencia Asociaciones
Cuándo	Eventos Compuertas
Qué recursos necesita	Objetos de Datos Asociaciones
Qué recursos produce	Flujo de Mensajes Actividades Objetos de Datos
Quién lo hace	Flujo de Mensajes Actividades
Qué reacción produce	Objetos de Datos Grupos Anotaciones
Hacer qué	Pools Lanes

Cuadro 2.3: Semántica de la Notación BPMN ROA

Aspectos de la Nomenclatura BPMN para el Modelado de Procesos ROA

Para visualizar el flujo de información a través de los procesos vamos a utilizar el concepto de un "token" de modo que nos permita explicar los comportamientos subyacentes de un modelo BPMN.

Un proceso es una actividad realizada por una empresa u organización en el cual, desde el punto de vista BPMN, se visualiza como una red de Flujos de Objetos; los cuales son un conjunto de actividades y control que dirigen el flujo. Por lo tanto, un token es un objeto "teórico" que utilizaremos para crear una "simulación" descriptiva del comportamiento.

Los flujos de objetos son principalmente los que expresan la semántica de un modelo de proceso BPMN; estos son:

- **Eventos:** Se representan con un círculo.
- **Compuertas:** Se representan con un rombo.
- **Actividades:** Se representan con un rectángulo con esquinas ovaladas.



Figura 2.7: Notaciones gráficas elementales del BPMN

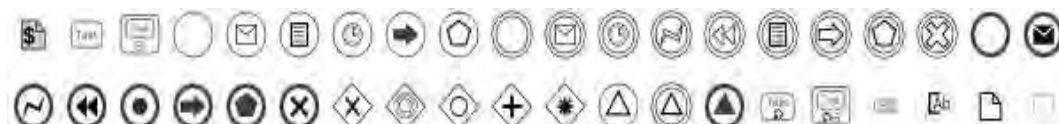


Figura 2.8: Listado de Elementos de notación gráfica BPMN

Suite de elementos BPMN para el Modelado de Procesos de Negocios con el Diseñador de Intalio.

Para conectar objetos se utiliza tres tipos de formas de conectores BPMN:

- **Flujo de Secuencia:** Se representa con una flecha con línea llena.
- **Flujo de Mensaje:** Se representa con una flecha con línea entre cortada.
- **Asociación:** Se representa con una flecha con línea punteada.

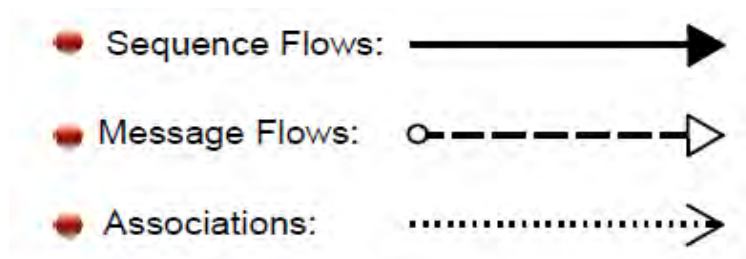


Figura 2.9: Notaciones gráficas de conectores BPMN

El BPMN utiliza los Swimlanes para dividir y organizar actividades en un diagrama, existen dos tipos: las Casetas ó Pools y los Carriles o Lanes que crean particiones dentro de un Pool.

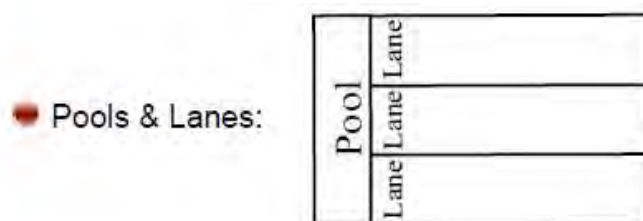


Figura 2.10: Gráfico de pools y lanes BPMN

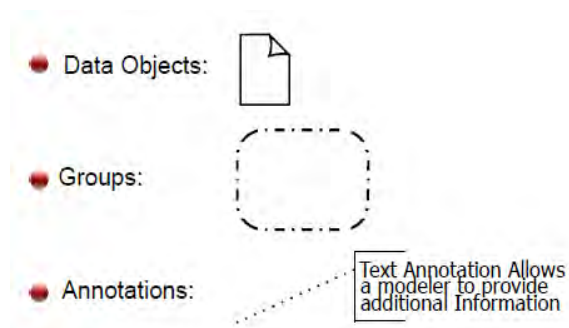


Figura 2.11: Notaciones gráficas de artefactos BPMN

El BPMN utiliza los artefactos para proporcionar información adicional acerca del proceso. Existen tres tipos:

- **Objetos de Datos:** Se representan por una hoja con la esquina superior derecha doblado.
- **Grupos:** Se representan con un rectángulo con esquinas ovaladas y líneas cortadas.
- **Anotaciones:** Se representan con una línea punteada y un cuadro para mensaje.

Evento: Un evento es algo que sucede durante el curso de un proceso de negocio y afecta su flujo de ejecución. Un evento tiene una causa y un impacto. BPMN define tres tipos de eventos:

- **Eventos de Inicio:** Se representan con un círculo simple.
- **Eventos Intermedios:** Se representan con dos círculos concéntricos simples.
- **Eventos de Fin:** Se representan con un círculo simple en negritas.



Figura 2.12: Notaciones gráficas de eventos BPMN

Evento de Inicio: Indica dónde inicia un proceso en particular.

Los eventos de inicio pueden ser utilizados en los siguientes escenarios:

- Cuando se desea indicar cómo la recepción de un mensaje puede activar la instancia de un proceso. Por ejemplo la recepción de una orden de servicio.
- Cuando se desea mostrar la activación vía la instancia de un proceso. Ejemplo: Envío de reportes los fines de mes.

Evento Intermedio: Indica dónde ha ocurrido algo durante la ejecución de un proceso. Un evento intermedio afecta el flujo de los procesos. Puede ser utilizado en los siguientes escenarios:

- Para indicar cuándo un mensaje debería ser recibido.
- Mostrar dónde se podrían producir retrasos.
- Cortar el flujo normal a través del manejo de excepciones.

Evento de Fin: Finaliza el flujo de un proceso. Los eventos de fin son opcionales. Pueden ser usados en los siguientes escenarios:

- Para finalizar el flujo de un proceso y enviar un mensaje.
- Para finalizar el flujo de un proceso y lanzar un error.
- Para finalizar el flujo de un proceso y solicitar una compensación.
- Para finalizar el flujo de un proceso.

Compuertas o Gateways: Son usados para controlar la forma cómo interactúan las secuencias de los flujos mientras éstos convergen o divergen en el proceso. Las decisiones tales como bifurcaciones (forks), uniones (merges) y junturas (join) en un flujo de un proceso son modelados con las compuertas. El comportamiento de cada tipo de compuerta determina cuántas de las puertas estarán disponibles para la continuidad del flujo. BPMN define cinco tipos de compuertas:

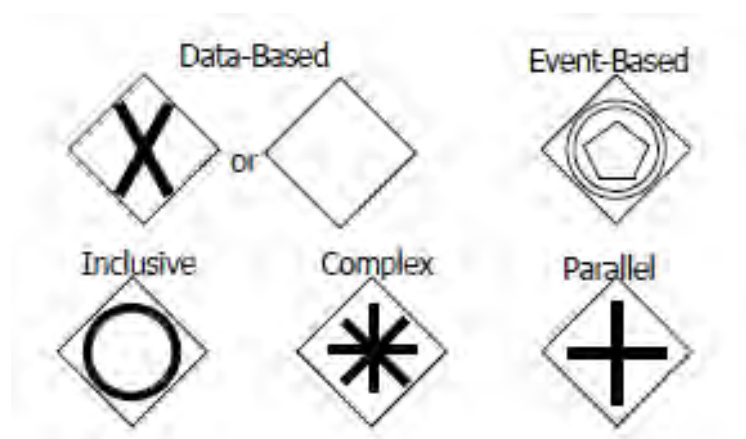


Figura 2.13: Notaciones gráficas de compuertas BPMN

Eventos

	"Coger"		"Lanzar"	
Sin Disparador				
Mensaje				
Temporizador				
Error				
Cancelación				
Compensación				
Condicional				
Enlace				
Señal				
Terminal				
Múltiple				

Figura 2.14: Notaciones gráficas a detalle de eventos BPMN

Compuerta Exclusiva Basada en Datos: Es utilizado en los siguientes escenarios:

- Como un punto de decisión; donde son posibles varios flujos de secuencia posteriores, aún cuando estos flujos están restringidos por una condición y solamente uno de estos será usado. Tal condición se puede evaluar basado en la data del proceso.
- Como una manera de juntar flujos de secuencia en uno solo. Los puntos de ejecución de entrada serán movidos directamente a través de la compuerta y seguirán su ruta.

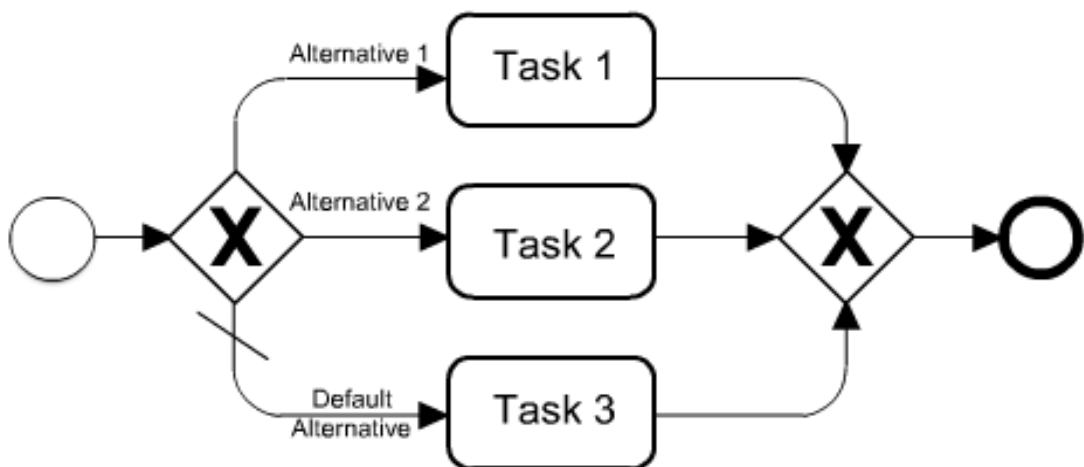


Figura 2.15: Notación de compuerta exclusiva BPMN

Una Compuerta Paralela: Proporciona un mecanismo para bifurcar y sincronizar los flujos.

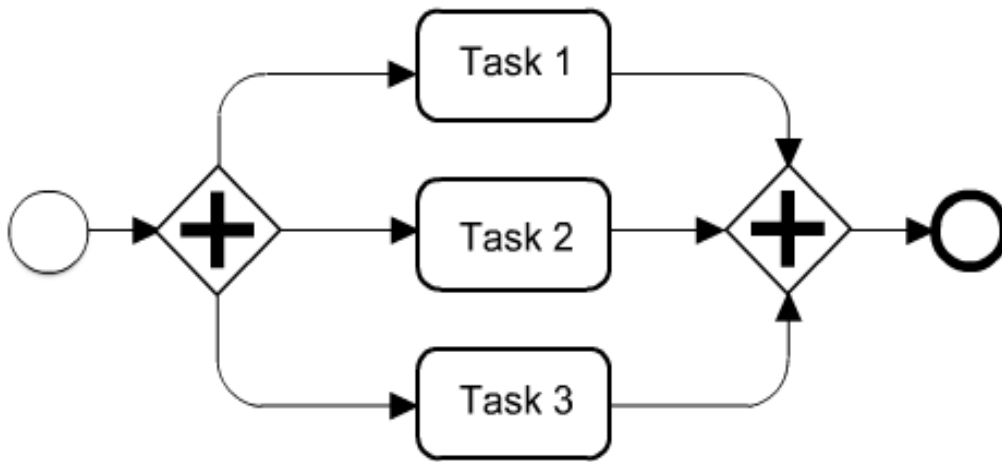


Figura 2.16: Notación de compuerta paralela BPMN

Compuerta Exclusiva Basada en Eventos: Es similar a la compuerta exclusiva basada en datos. La única diferencia es que en lugar de evaluar un conjunto de alternativas, para determinar un solo flujo de salida, el evento basado en la compuerta puede iniciar una carrera entre los diferentes eventos que el proceso permite recibir, el primero que se recibe gana la carrera y, por lo tanto, determina qué flujo de secuencia de salida deberá ser utilizado.

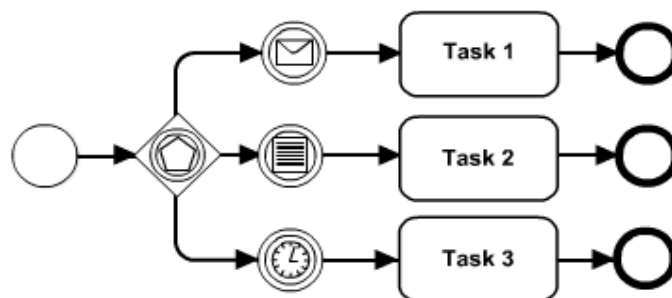


Figura 2.17: Notación de compuerta exclusiva BPMN con eventos

Una Compuerta Inclusiva: Puede ser utilizada en los siguientes escenarios:

- Como punto de decisión donde diversos flujos de secuencia de salida son posibles, estos flujos de salida son restringidos por condiciones; cada flujo de secuencia de salida que tenga una condición evaluada como verdadera será seguida.

- Como punto de unión, la compuerta inclusiva sincronizará a todos los puntos de ejecución producidos, pero como máximo uno por cada flujo de secuencia de entrada.

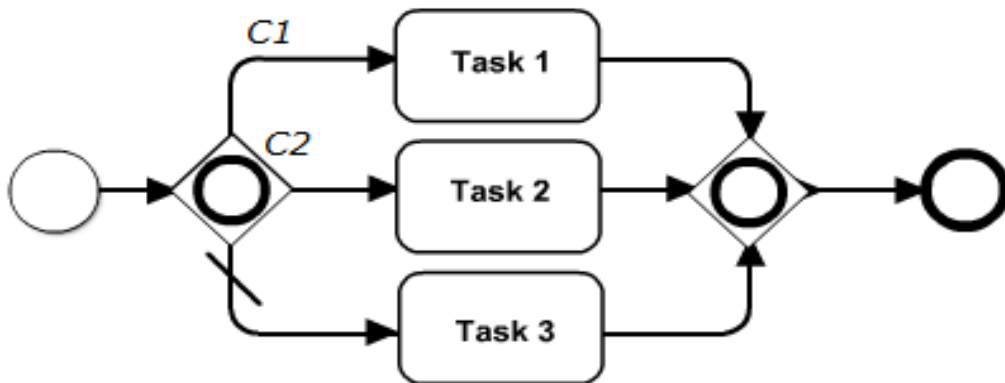


Figura 2.18: Notación de compuerta inclusiva BPMN

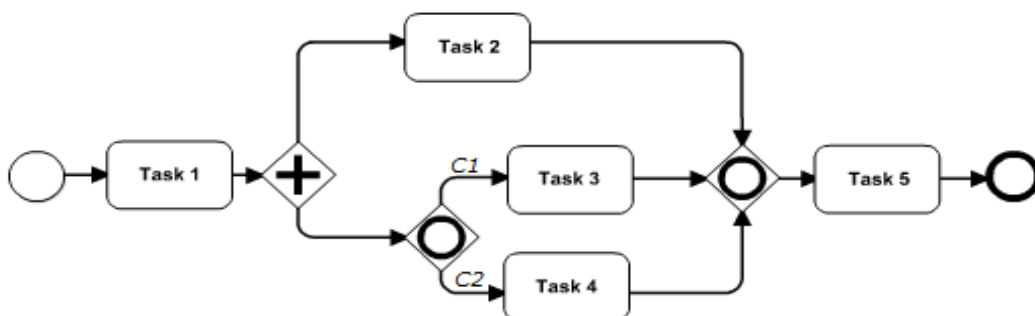


Figura 2.19: Notación de compuerta paralela compuesta BPMN

Una Compuerta Compleja: Se utiliza para manejar casos complejos el cual podría requerir la combinación de varias otras compuertas. Para evitar esto, el comportamiento de la compuerta compleja puede ser programado usando un lenguaje de expresión.

Una Actividad: Es una unidad de trabajo a ser realizado, ello podría ser una tarea, un proceso o un sub proceso. BPMN define dos principales formas de actividades:

- Una Tarea: Es una actividad atómica que es incluida con un proceso.
- Un Sub Proceso: Es un proceso que es incluido con otros procesos.

Los marcadores son definidos para especificar semánticas adicionales tales como los lazos.

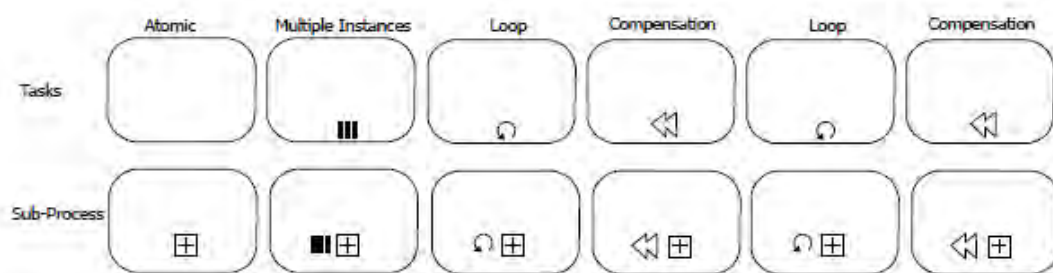


Figura 2.20: Notación de tareas y sub procesos BPMN

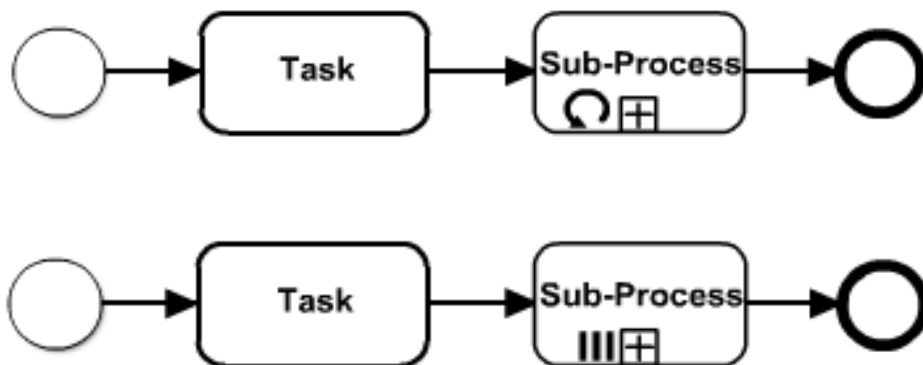


Figura 2.21: Notación de flujos en serie y paralelo BPMN

Los Swimlanes: Son usados para organizar actividades y mostrar la colaboración entre los partners. Un proceso de negocio es luego organizado dentro de una Caseta o Pool de un Swimlane. Un swimlane particular presenta las tareas relevantes hacia una unidad de negocio específica para una instancia.

2.3. Evolución y crecimiento de los Web Services

En los primeros años de la computación, las aplicaciones basadas en mainframes fueron consideradas como la mejor propuesta de solución para ejecutar aplicaciones de procesamiento de datos a gran escala. Con el advenimiento de las computadoras personales, el concepto de software de aplicación corriendo en máquinas aisladas llegó a ser popular en términos de costo de licencia y aplicación de fácil uso. Con el número creciente de programas de aplicación de PCs corriendo en diferentes máquinas, las comunicaciones entre tales aplicaciones de programas se tornaron extremadamente complicadas y por consiguiente surge un nuevo reto en el aspecto de la interacción aplicación - aplicación.

La computación distribuida es un tipo de computación en la cual diferentes componentes y objetos comprometidos en una aplicación pueden ser localizados en diferentes computadoras conectadas a una red. Hoy en día, Sun Java RMI (Remote Method Invocation), OMG CORBA (Common Object Request Broker Architecture) y Microsoft DCOM (Distributed Component Object Model); están emergiendo como las más comunes tecnologías de computación distribuida. Estas tecnologías, aunque diferentes en el diseño e implementación de su arquitectura básica, direccionan problemas específicos en sus objetivos de entorno. [ver referencia 4]

Hoy en día se utilizan modelos híbridos basados en contenedores J2EE que alberga un servidor CORBA, es decir, aplicaciones que contienen componentes en Java que utilizan CORBA para comunicarse con aplicaciones desarrolladas en lenguaje C. En el Perú un caso típico es la plataforma de Red Inteligente NGN, en donde la lógica del dominio de negocio es desplegada mediante los EJB; mientras que los módulos de cálculo para las tarifas son desarrolladas en C, debido a la necesidad de velocidad en los tiempos de procesamiento.

Para ubicar los web services en la historia CORBA es importante tener presente

algunos datos:

- Octubre 1991: CORBA 1.0 (Modelo de objeto)
- Agosto 1996: CORBA 2.0 (Interoperabilidad ORB y estandarización del lenguaje de mapeo en C++)
- Verano 1998: CORBA 2.3 (Estandarización del lenguaje de mapeo Java)
- Finales 1999: CORBA 3.0 (Calidad de Servicio, mensajería)

En promedio CORBA apareció alrededor de 1991, y desde 1996 ha sido un estándar robusto y útil que generó varios productos ORB interoperables de varios vendedores. Curiosamente SOAP apareció en el 2000, es decir 9 años después de la versión 1.0 de CORBA.

Entre las características de CORBA figuran:

- CORBA es un framework para construir sistemas distribuidos orientados a objetos.
- CORBA es independiente del lenguaje.
- CORBA es un estándar abierto para computación distribuida definida por OMG (organización fundada en 1989).
- CORBA usa formato binario compacto para la transmisión. Esto permite un eficiente uso del ancho de banda.
- La interacción entre el cliente y el servidor en CORBA es mediante los ORBs.
- Si una solicitud CORBA es enrutada a través de un firewall, este dispositivo necesita ser configurado añadiendo información explícita de enrutamiento al IOR (Interoperable Object Reference).
- En aplicaciones móviles si se usa CORBA, estos dispositivos necesariamente deben tener componentes ORB.
- CORBA requiere más hardware (en plataformas móviles) que los web services; por lo tanto aumenta el costo.
- El valor de los web services no está en el reemplazo de CORBA, sino en

el complemento.

Para aplicaciones pequeñas en donde se requiera una comunicación entre aplicaciones Java y dispositivos DSP en su mayoría programados en C, se utiliza JNI, que en esencia es una interface nativa Java que mapea o enlaza los tipos de datos entre el lenguaje de programación Java y C a nivel de código objeto. [23]

Antes de describir la arquitectura J2EE, es importante describir algunos aspectos de Java y los estilos de arquitectura de las aplicaciones distribuidas contemporáneas: las arquitecturas de nivel 2, nivel 3 y nivel n. Aunque estos estilos de arquitectura se encuentran con bastante frecuencia en las empresas de la actualidad, cabe señalar que estos estilos surgieron debido a la llegada de nuevas plataformas de hardware para clientes y servidores más baratas y de tecnología de redes. Anteriormente, los motores de las empresas eran sistemas principales con todos los servicios informáticos (desde la generación de usuario-interfaz hasta procesamiento de transacciones de gran volumen) centralizados. Consecuentemente, la tecnología cliente servidor fue la que provocó una automatización masiva de la empresa.

Dentro de la evolución de la tecnología Java, es importante mencionar algunas diferencias:

Applet

Es bajado desde un servidor web y es ejecutado en un browser.

Java Beans

Son componentes de software hechos en Java, su desarrollo se basa en un modelo de programación de componentes, para su proceso hay que seguir ciertos estándares y son reutilizables. Pueden ser utilizados en entornos visuales de programación y auto describen sus propiedades y pueden ser vistas y

alteradas en una herramienta de desarrollo visual.

Características:

- Sus variables no se acceden directamente, sino a través de métodos estándares (getters y setters).
- La nomenclatura de estos métodos siguen ciertos estándares.
- Una clase BeanInfo acompaña a la clase principal.
- Esta clase describe al JavaBean y permite que se pueda utilizar en un ambiente de desarrollo visual.
- Los JavaBeans pueden ser visuales (widgets) o no visuales (sólo lógica).

Servlets

Un servlet es una clase Java que corre en el Web Server (o un proceso lógicamente vinculado al Web Server). Un servlet acepta los requerimientos HTTP y responde con páginas HTML.

El HTML es normalmente dinámico y refleja información que el servlet obtiene de varios sistemas empresariales.

Características:

- Son independientes del servidor utilizado y de su sistema operativo.
- Los servlets pueden llamar a otros servlets, e incluso a métodos concretos de otros servlets.
- Los servlets pueden obtener fácilmente información acerca del cliente.
- Los servlets permiten la generación dinámica de código HTML. Los servlets son a los servidores, como los applets a los navegadores.

JSP

Los Java Server Pages son similares a un HTML sólo que el contenido se construye dinámicamente dentro de la página web. Los JSP soportan todos los

tags de HTML. El diseño puede variar sin modificar el contenido dinámico y es una extensión de la tecnología de los servlets y está basado en la reutilización de componentes.

Características:

- Separación entre generación de contenido y presentación.
- Con HTML o XML se da formato a la página.
- Mediante etiquetas JSP, script o Java Beans se añade el componente dinámico.
- Independencia de la plataforma, tanto cliente como servidor.
- Fácil de administrar y de usar.

Ventajas:

Frente a un ASP:

- El código está escrito en Java.
- Portable entre sistemas operativos.
- Se compilan y permanecen en memoria, no son interpretadas.

Frente a un Servlet:

- Permite separar el diseño de la página web del código.
- Facilita el mantenimiento.

Frente a Java Script:

- Java Script genera HTML dinámicamente en el cliente.
- No puede acceder a recursos del servidor.

Tecnologías J2EE

Las tecnologías J2EE están basadas en componentes, servicios y comunicaciones.

Componentes

- Componentes del lado del cliente:
 - Applets.
 - Application Clients.
- Componentes del lado del servidor:
 - EJBs.
 - Web Components (Servlets, JSP).

Servicios

- Funciones utilizadas por los componentes J2EE.
- APIs implementadas por el proveedor de la plataforma J2EE.

Comunicación

- Permite la comunicación entre componentes.
- Proveído por el contenedor.

Servicios y Comunicaciones J2EE

- Servicios de nombres:
 - Java Naming and Directory Interface: JNDI.
- Servicios de base de datos:
 - Java Data Base Connectivity: JDBC.
- Servicios de transacciones:
 - Java Transaction API: JTA.
 - Java Transaction Service: JTS.
- Servicio de comunicaciones:
 - Remote Method Invocation over Internet InterORB Protocol: RMI-IIOP.
 - Java Mail.
 - Java Beans Activation Framework: JAF.

- Java Message Service: JMS.

Contenedor J2EE

Cada componente corre dentro de un contenedor que es ofrecido por el proveedor de la plataforma; este contenedor provee servicios específicos a la implementación y ejecución del componente (ciclo de vida, seguridad, transacciones, etc.). Los servicios brindados dependen del tipo de contenedor.

Plataformas Soportadas

Sistemas Operativos:

- AIX 4.3.3 ML 10, 5.1 ML2, 5.2
- Linux 2.4 Kernel
- Red Hat Advanced Server 2.1
- SuSE 7.3, SuSE 8, SLES 7
- Solaris 8
- HP/UX 11
- Microsoft (Windows NT 4.0 SP6a, W2K Server SP3, Advanced Server SP3, Win2003)
- OS/400 OS/390, zOS,zLinux

Servidores Web Soportados:

- Apache
- Microsoft Internet Information Server
- Sun One (iPlanet) Web Server
- Domino Web Server

Especificaciones implementadas:

- J2EE 1.3
- Web Services:
 - SOAP 2.3, WSIF
 - UDDI clients
 - Web Services Gateway
 - UDDI Registry – for private UDDI registry
- Seguridad:
 - Java 2, JAAS, CSiv2
 - Soporte para crypto card Eracom CSA 8000

Arquitectura de dos niveles

Los sistemas cliente servidor típicos están basados en la arquitectura de dos niveles, donde existe una clara separación entre los datos y la lógica de presentación de la empresa. Estos sistemas están generalmente controlados por datos; siendo el servidor, en gran cantidad de ocasiones, un servidor de base de datos. Aunque este enfoque nos permite compartir datos en la empresa, tiene muchos inconvenientes.

En una aplicación tradicional de 2 niveles, la carga de procesamiento es facilitada al PC cliente mientras que el servidor actúa simplemente como controlador del tráfico entre la aplicación y los datos. Como resultado, el rendimiento de la aplicación no sólo sufre debido a los recursos limitados de la PC, sino que el tráfico de la red también tiende a aumentar. Cuando la aplicación completa es procesada en una PC, la aplicación es forzada a realizar múltiples peticiones de datos antes incluso de presentar algo al usuario. Estas múltiples peticiones de bases de datos pueden sobrecargar la red.

Otro problema típico relacionado con el enfoque de 2 niveles tiene que ver con el mantenimiento. Incluso el menor cambio realizado a una aplicación puede

conllevar una completa alteración en la base del usuario. Aunque sea posible automatizar el proceso, todavía debe enfrentarse a la actualización de cada instalación de cliente.

Es más, algunos usuarios puede que no estén preparados para una alteración total y posiblemente ignoren los cambios, mientras que otro grupo puede que insista en realizar los cambios de inmediato. Esto puede provocar que diferentes instalaciones de cliente utilicen diferentes versiones de la aplicación.

Arquitectura de tres niveles

Con el afán de solucionar estos inconvenientes, la comunidad de software desarrolló la noción de una arquitectura de 3 niveles. Una aplicación se divide en tres capas lógicas distintas, cada una de ellas con un grupo de interfaces perfectamente definidas. La primera capa se denomina capa de presentación y normalmente consiste en una interfaz gráfica de usuario de algún tipo. La capa intermedia, o capa de empresa, consiste en la aplicación o lógica de empresa, y la tercera capa, la capa de datos, contiene los datos necesarios para la aplicación.

La capa intermedia (lógica de aplicación) es básicamente el código al que recurre el usuario (a través de la capa de presentación) para recuperar los datos deseados. La capa de presentación recibe entonces los datos y los formatea para su presentación. Esta separación entre la lógica de aplicación de la interfaz de usuario añade una enorme flexibilidad al diseño de la aplicación. Pueden construirse y desplegarse múltiples interfaces de usuario sin cambiar en absoluto la lógica de aplicación, siempre que la lógica de aplicación presente una interfaz claramente definida a la capa de presentación.

J2EE ofrece diversas abstracciones para satisfacer las necesidades de cada uno de estos niveles. Por ejemplo, los EJB ofrecen mecanismos para abstraer tanto

el acceso a datos como la lógica de empresa. De un modo similar, los servlets y las páginas Java Server permiten abstraer la capa de presentación y su interacción con la capa de empresa.

La tercera capa contiene los datos necesarios para la aplicación. Estos datos consisten en cualquier fuente de información, incluido una base de datos de empresa como Oracle o Sybase, un conjunto de documentos XML o incluso un servicio de directorio como el servidor LDAP.

Arquitectura de n niveles

Este tipo de sistema puede dar cabida a varias configuraciones diferentes. En una arquitectura de n niveles, la lógica de aplicación está naturalmente dividida por funciones y no físicamente. [ver referencia 11]

Una arquitectura de n niveles se descompone en las siguientes partes:

Una interfaz de usuario, que maneja la interacción del usuario con la aplicación. Esta puede ser un navegador web que se transmite mediante la salida WAN de un firewall, una aplicación de escritorio o, incluso, un dispositivo inalámbrico.

Una lógica de presentación, que define lo que muestra la interfaz de usuario y cómo son gestionadas las demandas del usuario. Dependiendo de las interfaces de usuario que se mantengan, puede que sea necesario contar con versiones de la lógica de presentación ligeramente diferentes para satisfacer al cliente adecuadamente.

Una lógica de empresa, que modele las reglas de empresa de la aplicación. A menudo a través de la interacción con los datos de la aplicación.

Servicios de infraestructura, que proporcionen la funcionalidad adicional requerida por los componentes de la aplicación; tales como mensajería y apoyo

transaccional.

La capa de datos, donde residen los datos de la empresa.

Las aplicaciones basadas en esta arquitectura emplean esencialmente el patrón Modelo-Vista-Controlador (MVC). Lo que esto significa fundamentalmente es que los datos (el modelo) son independientes respecto de la presentación de la información (la vista). En una posición intermedia se encuentra la lógica de aplicación/empresa (el controlador) que controla el flujo de información. De este modo, una aplicación es diseñada a partir de estos tres componentes funcionales (modelo, vista y controlador) interactuando entre sí.

Arquitectura de empresa

En una empresa, intentamos crear una arquitectura cohesiva que incorpore diferentes aplicaciones. Antes que un cambio en la arquitectura (la arquitectura de empresa es básicamente n niveles) necesitamos un cambio en la percepción. Para convertir un sistema de n niveles en un sistema de empresa, simplemente necesitamos ampliar el nivel intermedio permitiendo la existencia de múltiples objetos de aplicación en lugar de una única aplicación. Estos objetos de aplicación deben tener una interfaz individual que les permita funcionar entre sí.

Una interfaz puede concebirse como un contrato. Cada objeto afirma a través de su interfaz, que aceptará ciertos parámetros, realizará ciertas operaciones y devolverá un conjunto específico de resultados. Los objetos de la aplicación se comunican entre sí utilizando sus interfaces.

Con la arquitectura de empresa, podemos tener múltiples aplicaciones utilizando un conjunto común de componentes a través de una organización. Esto promueve la normalización de la práctica empresarial gracias a la creación de un único conjunto de funciones de empresa al que puede acceder toda la organización. Si la norma de una empresa cambia, dichos cambios deben

trasladarse también al objeto de empresa y, si es necesario, a la interfaz asociada y consecuentemente a cualquier objeto que acceda a la interfaz. Sin embargo, ciertas funcionalidades de los sistemas legados existentes no pueden eliminarse por lo que debe desarrollarse un envoltorio (o interfaz) específico a esta funcionalidad en el nivel intermedio.

Con el paso de los años, Java ha desarrollado tres ediciones de plataformas diferentes; cada una de ellas destinada a cubrir un conjunto diferente de necesidades de programación:

La plataforma Java 2 Standard Edition (J2SE). Esta es la plataforma más utilizada de Java, consistente en un entorno de ejecución y un conjunto de varias API para crear una amplia variedad de aplicaciones: que abarca desde applets, pasando por aplicaciones independientes ejecutables en distintas plataformas, hasta aplicaciones de cliente para diversas aplicaciones de empresa.

La plataforma Java 2 Enterprise Edition (J2EE). J2EE es una plataforma para crear aplicaciones de servidor, los cuales tienen requisitos adicionales durante la fase de desarrollo. J2EE proporciona la infraestructura necesaria para satisfacer estas necesidades.

La plataforma Java 2 Micro Edition (J2ME). Esta edición, la última en agregarse a la familia Java, permite la creación de aplicaciones Java para micro dispositivos (dispositivos con un apoyo de pantalla y memoria limitado, como teléfonos móviles, PDAs, etc.).

2.3.1. Conceptos fundamentales

Middleware es un software de conectividad que consiste en un conjunto de servicios que permiten que varios procesos que se ejecutan en una o más máquinas puedan interactuar en una red.

Middleware es esencial para emigrar aplicaciones mainframe hacia las aplicaciones cliente/servidor y también para mantener comunicación entre las plataformas heterogéneas.

Esta tecnología ha evolucionado, durante los años noventa, para mantener interoperabilidad en apoyo del movimiento a las arquitecturas cliente/servidor.

Entre las iniciativas middleware más divulgadas tenemos: El Entorno de Computación Distribuida (DCE) de la Fundación de Software Libre , la Arquitectura de Agentes de Solicitud de Objetos Comunes (CORBA) del Grupo de Administración de Objetos, y el Modelo de Objeto de Componentes /Distribuidos COM/DCOM de Microsoft.

Los servicios middleware son conjuntos de software distribuidos que existen entre la aplicación, el sistema operativo y los servicios de la red en un nodo del sistema en la red.

Los servicios Middleware proporcionan un conjunto más funcional de Interfaces de Programación de Aplicaciones (API) que el mismo sistema operativo y los servicios de la red juntas podrían permitir para una aplicación. Como por ejemplo:

- Localizar transparentemente por la red y proporciona interacción con otra aplicación o servicio.
- Proporcionar independencia de los servicios de la red.
- Brindar fiabilidad y disponibilidad.
- Permitir escalar en capacidad sin perder su funcionalidad.

En términos generales un Middleware puede asumir las formas siguientes:

1. Monitor de Procesamiento de Transacción (TP), que provee herramientas y un ambiente de desarrollo y despliegue de aplicaciones distribuidas.

2. Llamada de Procedimiento remoto (RPCs), que permite distribuir la lógica de una aplicación por la red. La lógica del programa en los sistemas remotos puede ejecutarse simplemente llamando a una rutina local.

3. Middleware orientado a Mensajería (MOM), que provee el intercambio de datos de un programa a otro, habilitando la creación de aplicaciones distribuidas. MOM es equivalente a un correo en el sentido que es asíncrono y exige a los destinatarios de los mensajes interpretar su significado y tomar una acción apropiada.

4. Agentes de Solicitud de Objetos (ORBs), que permiten a los objetos comprender y compartir una aplicación distribuida a través de redes heterogéneas.

Entre las tecnologías emergentes en esta línea tenemos a:

- a) El Grupo de Administración de Objetos (OMG) a través de la Arquitectura de Agentes de Solicitud de Objetos Comunes (CORBA).
- b) La especificación del Modelo de Objeto de Componentes de Microsoft (COM /DCOM).
- c) La especificación de la máquina virtual Java edición empresarial J2EE a través de la Invocación de Métodos Remotos (RMI); RMI permite que Java pueda ser ejecutada remotamente en forma nativa.

A lo largo de los últimos años, los fondos y el tiempo invertidos en el desarrollo de aplicaciones se han visto reducidos, mientras que la demanda para crear complejos procesos económicos han aumentado. En ese ámbito nos encontramos con los siguientes requisitos que cumplir en el desarrollo de

aplicaciones:

Productividad de programación: La adopción directa de nuevas tecnologías es insuficiente, si éstas no son utilizadas adecuadamente aprovechando su potencial al máximo, y si no son integradas apropiadamente con otras tecnologías relevantes. De este modo, la habilidad para desarrollar y después desplegar la aplicación tan rápida y eficazmente como sea posible es muy importante. Conseguir este objetivo puede resultar complicado, debido a la gran variedad de tecnologías y normas que han sido desarrolladas a lo largo de los años, que requieren capacidades altamente desarrolladas.

Fiabilidad y disponibilidad: En la economía actual de Internet, el tiempo de inactividad puede resultar fatal para una empresa. La habilidad para organizar y poner en marcha sus operaciones de base web y mantenerlas activas, es crucial para lograr el objetivo con éxito. También debe ser capaz de garantizar la fiabilidad de sus transacciones económicas de modo que sean procesadas por completo y con exactitud.

Seguridad: Internet no sólo ha aumentado exponencialmente el número de usuarios potenciales, sino también el valor de la información de una compañía; de modo que la seguridad de esta información se ha convertido en una preocupación de primer orden. Es más, a medida que las tecnologías son más avanzadas, las aplicaciones más sofisticadas y las empresas más complejas; poner en marcha un modelo de seguridad efectivo es cada vez más difícil.

Reajustabilidad: La capacidad de una aplicación para crecer y ajustarse a una nueva demanda, tanto en su operación como en su base de usuario, es esencial. Esta condición se cumple especialmente cuando consideramos que la base potencial de usuarios de una aplicación puede ser de millones de usuarios individuales, vía Internet. La capacidad efectiva de reajuste depende no sólo de la habilidad para manejar un amplio aumento en el número de clientes, sino

también del uso efectivo de los recursos de un sistema.

Integración: Aunque la información ha crecido hasta convertirse en un dispositivo económico clave, mucha de esta información existe como datos en sistemas de información viejos y obsoletos. Para maximizar la utilidad de esta información, las aplicaciones deben poder ser integradas en los sistemas de información ya existentes; lo cual no es necesariamente una tarea fácil ya que las tecnologías actuales con frecuencia han avanzado hasta superar algunos de estos sistemas legados. La habilidad para combinar las nuevas y las viejas tecnologías: es fundamental para el éxito del desarrollo de las empresas de hoy en día.

Justamente, el objetivo de J2EE es simplificar algunas de las complejidades técnicas y especificar los siguientes puntos para crear aplicaciones:

- Un modelo de programación, consistente en un conjunto de interfaces de programación de aplicaciones (API) y en los enfoques para la creación de dichas aplicaciones.
- Una infraestructura de aplicación, para respaldar las aplicaciones de empresa creadas utilizando las APIs.

Existen esencialmente dos caminos principales: uno emprendido por Microsoft, con su nueva iniciativa: .NET, y el segundo, emprendido por Sun y otras empresas tales como BEA, IBM, Oracle, etc, con Java.

Teniendo en cuenta las opciones, se analiza algunas de las ventajas más trascendentes:

Independiente de plataforma: Con la información de una empresa desplegada en formatos dispares en distintas plataformas y aplicaciones, es importante

adoptar un lenguaje de programación que pueda funcionar perfectamente en la empresa sin tener que recurrir a mecanismos extraños e ineficientes de traducción. Un modelo de programación unificador también reduce las dificultades surgidas de la integración de muchas de las diversas tecnologías que se convierten en específicas para ciertas plataformas y aplicaciones.

Objetos gestionados: J2EE proporciona un entorno gestionado para componentes y las aplicaciones de J2EE son céntricas respecto del contenedor. Ambas nociones son críticas para construir aplicaciones de cliente. Al ser gestionados, los componentes de J2EE utilizan la infraestructura proporcionada por los servidores de J2EE sin que el programador sea consciente de ello. Las aplicaciones de J2EE son también declarativas, un mecanismo con el que puede modificar y controlar el funcionamiento de las aplicaciones sin cambiar de código. A primera vista, estas características pueden resultar incómodas de ejecutar. Sin embargo, cuando considera las aplicaciones a gran escala con varios cientos de componentes interactuando para ejecutar complejos procesos empresariales, estas características hacen que crear y mantener tales aplicaciones resulte menos complejo. Junto con su independencia respecto de la plataforma, éste es uno de los aspectos más importantes de J2EE.

Reusabilidad: Separar los requisitos económicos de una aplicación en sus partes integrantes es un modo de conseguir la reutilización; utilizar la orientación al objeto para encapsular la funcionalidad compartida es otro. Sin embargo, a diferencia de los objetos, los componentes distribuidos requieren una infraestructura más compleja para su construcción y gestión. Los conceptos básicos orientados a objetos no proporcionan este tipo de marco, pero la Enterprise Edition de Java ofrece una arquitectura notablemente rigurosa para la reutilización de componentes. Los componentes de empresa (llamados Enterprise JavaBeans) desarrollados en J2EE son de básica granulometría y reproducibles. Manteniendo estos componentes básicos, es posible crear funcionalidades de empresa complejas de un modo libremente coordinado.

Además, ya que los componentes son reproducibles, lo cual quiere decir que es posible identificar ciertos metadatos sobre los componentes, las aplicaciones pueden ser creadas componiendo tales componentes. Ambas características fomentan la reutilización del código a alta granulometría.

Modularidad: A la hora de desarrollar una aplicación de servidor completa, los programas pueden ampliarse y complicarse rápidamente. Siempre es mejor romper una aplicación en módulos discretos, cada uno de ellos responsable de una tarea específica. Esto hace que nuestras aplicaciones sean mucho más fáciles de mantener y comprender. Por ejemplo, los servlets de Java, las Páginas de JavaServer y Enterprise JavaBeans: proporcionan un modo de modularizar nuestra aplicación, rompiendo nuestras aplicaciones en diferentes niveles y tareas individuales.

Históricamente Java surgió en 1991 cuando un grupo de ingenieros de Sun Microsystems trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido. Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizada.

Desarrollaron un código neutro que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una máquina hipotética o virtual denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: Write Once, Run Everywhere. A pesar de los esfuerzos realizados por sus creadores, ninguna empresa de electrodomésticos se interesó por el nuevo lenguaje. Como lenguaje de programación para computadores, Java se introdujo a finales de 1995. La clave fue la incorporación de un intérprete Java en la versión 2.0 del programa

Netscape Navigator, produciendo una verdadera revolución en Internet. Java 1.1 apareció a principios de 1997, mejorando sustancialmente la primera versión del lenguaje. Java 1.2; más tarde rebautizado como Java 2, nació a finales de 1998. Al programar en Java no se parte de cero.

Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.).

Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la informática moderna porque incorpora todos estos conceptos de un modo estándar, mucho más sencillo y claro que con las citadas extensiones de otros lenguajes. Esto es consecuencia de haber sido diseñado más recientemente y por un único equipo. El principal objetivo del lenguaje Java, que es llegar a ser el nexo universal que conecte a los usuarios con la información, está situada en el ordenador local, en un servidor de web, en una base de datos o en cualquier otro lugar. Java es un lenguaje muy completo (de hecho se está convirtiendo en un macro-lenguaje: Java 1.0 tenía 12 packages; Java 1.1 tenía 23 y Java 1.2 tiene 59). En cierta forma casi todo depende de casi todo. Por ello, conviene aprenderlo de modo cíclico: primero una visión muy general, que se va refinando en sucesivos pasos. Una forma de hacerlo es empezar con un ejemplo completo en el que ya aparecen algunas de las características más importantes. La compañía Sun describe el lenguaje Java como simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico.

A pesar de las características a su favor mencionadas, Java no presentó un

modelo de programación unificado a escala empresarial hasta principios del año 2000. No es que no tuviera la capacidad, sino que en el pasado estaba algo desorganizada. Sun reconoció este fallo y lanzó la Plataforma Java 2, Enterprise Edition (J2EE).

2.3.2. Introducción a los Web Services

Los servicios web son módulos de software que realizan tareas o conjuntos de tareas discretas, a los que se accede y se llama a través de una red, sobre todo la World Wide Web. El desarrollador puede crear una aplicación cliente que llama a una serie de servicios web mediante llamadas a procedimientos remotos (RPC) o un servicio de mensajes en los que se proporciona un fragmento o la mayor parte de la lógica de la aplicación. Los servicios web publicados se describen de forma que los desarrolladores pueden localizarlos y determinar si se ajustan a sus necesidades.

Por ejemplo, una empresa puede proporcionar a sus clientes un servicio web por el cual se comprueba el inventario de productos antes de realizar un pedido. Otro ejemplo es el servicio de seguimiento de paquetes de la empresa de mensajería Aero Continente, mediante el cual los clientes pueden examinar el estado de sus envíos.

Los servicios web utilizan SOAP (Simple Object Access Protocol) para la carga XML y utiliza un transporte del tipo HTTP para llevar los mensajes SOAP de un lado a otro. En realidad, los mensajes SOAP son los documentos XML que se envían entre el servicio web y la aplicación que efectúa la llamada.

Los servicios web se pueden escribir en cualquier lenguaje, y se ejecutan en todas las plataformas. Los clientes de servicios web también se pueden escribir en cualquier lenguaje y se ejecutan en todas las plataformas. Así, por ejemplo, un cliente escrito en Delphi que se ejecuta en Windows puede llamar a un servicio web escrito en Java que se ejecuta en Linux.

En el cuadro 2.4 se muestra la historia de la tecnología de los web services, donde se puede apreciar un cambio rápido entre los años 2000 y 2002. Esta tendencia implica la existencia de dos variantes en los estándares de los Web Services (WSDL-UDDI-SOAP y ebXML) evolucionando ambas en el espacio de los web services. Con el paquete JAX, los desarrolladores pueden usar transparentemente ambas variantes tecnológicas.

2.3.3. Tecnologías y Arquitecturas de Desarrollo de los Servicios Web

La arquitectura de los servicios web permite desarrollar servicios que encapsulan todos los niveles de funcionalidad empresarial. En otras palabras, un servicio web puede ser muy simple, como por ejemplo, uno que informe sobre la temperatura actual, o una aplicación compleja. Esta arquitectura también permite la combinación de varios servicios web con el fin de crear nuevas funciones. La arquitectura de servicios web tiene tres cometidos: proporciona datos, los solicita y hace de intermediario.

Como proveedor, crea el servicio web y lo pone a disposición de los clientes que desean utilizarlo. Realizan la solicitud las aplicaciones clientes que consumen el servicio web. Este servicio solicitado también puede ser un cliente de otros servicios web. El intermediario, como un registro de servicio, permite interactuar a la aplicación cliente y al proveedor.

Estos tres cometidos interactúan por medio de las operaciones de publicación, búsqueda y enlace. El proveedor utiliza la interfaz de publicación del intermediario para comunicarle la existencia del servicio web, con el fin de ponerlo a disposición de los clientes. La información publicada describe el servicio e indica dónde se encuentra. La aplicación que hace la solicitud consulta al intermediario para que busque los servicios web publicados. Con la información obtenida del intermediario, la aplicación cliente puede enlazarse al servicio web (llamarlo).

Estandar de Servicio Web	Versión	Fecha de Lanzamiento	Estado
WSDL (Lenguaje de Descripción de Servicio Web)	1.1	Marzo 15, 2001	W3C Nota emitida por IBM, Microsoft y Ariba
	1.2	Enero 24, 2003	Emitida como un Trabajo Borrador para el W3C. Existe una nueva WSDL Binding (usando WSDL con SOAP 1.2 HTTP)
UDDI (Interface de Descripción Descubrimiento e Integración)	1.0	1999	Propuesto por Microsoft y la organización de UDDI.org
	2.0	2001	Gestionado por UDDI.org
	3.0	Julio 19, 2002	Actualmente gestionado por OASIS
SOAP (Protocolo de Acceso a Objetos Simple)	1.1	Mayo 8, 2000	Propuesto por Microsoft, IBM UserLand, Develop Mentor
	1.1 con Adenda	Diciembre 11, 2000	Propuesto por HP y Microsoft
	1.2	Diciembre 19, 2002	Actualmente es candidato recomendado al W3C
Especificación de Mensaje ebXML	1.0	Mayo 11, 2001	Gestionado pro OASIS. Usa SOAP 1.1 con Adenda de transporte y ruteo
	2.0	Abril 1, 2002	Gestionado por OASIS. El estándar es aprobado en Septiembre 5, 2002
Servicio de Registro ebXML	1.0	Mayo 8, 2001	Gestionado por OASIS
	2.0	Diciembre 18, 2001	Gestionado por OASIS. El estándar es aprobado en Diciembre 2001
	2.1	Junio 2002	Tipo de Direccionamiento y errores mínimos en la versión 2.0

Cuadro 2.4: Historia de la Tecnología de los web services

En la Figura 2.22 se resume la forma en que interaccionan los tres cometidos.

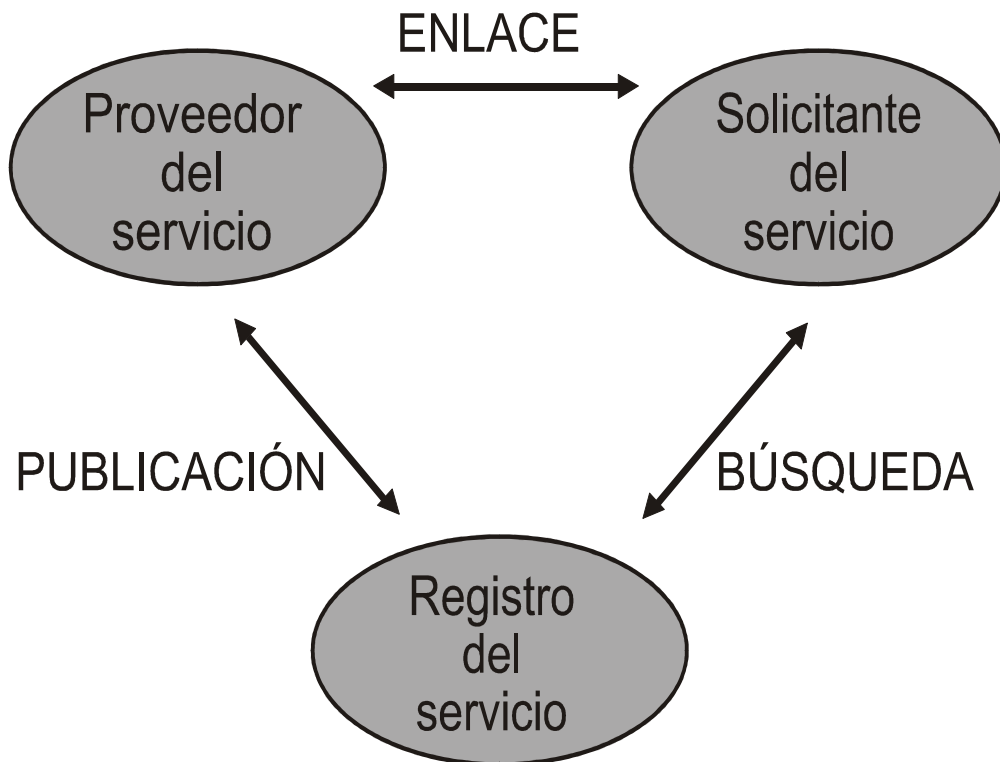


Figura 2.22: Operaciones y cometidos de los web services

En el cuadro 2.4 se muestran los componentes lógicos de los web services tales como: los servicios de solicitud, agente, proveedor y registro. En un típico escenario usando web services, el servicio de solicitud (que puede ser un proveedor o comprador) son consumidores de los servicios del negocio. El proveedor es un cliente del servicio de proveedor. El comprador es un cliente del servicio de agente. El servicio de agente actúa como un intermediario para diferentes servicios de negocios proporcionados por el servicio de proveedor. El servicio de agente publica sus servicios de negocios en un servicio de registro privado (usando ambos registros ebXML y UDDI). El servicio de proveedor también publica sus servicios de negocios (usando por ejemplo WSDL) en un servicio de registro privado ebXML y hacia el servicio de agente.

El comprador está interesado en encontrar un producto de negocio específico desde el servicio de agente. El servicio de agente almacena los servicios de negocio proporcionados por el servicio de proveedor. Durante el descubrimiento del servicio, el comprador encuentra un apropiado servicio de negocio desde el proveedor vía el servicio de registro ebXML del servicio de agente. Luego el comprador enlaza e invoca el servicio de negocio. Las acciones en un escenario de web services pueden incluir:

- ◆ **Descubrimiento/Búsqueda:** Investigando para un servicio de negocio particular, usualmente mediante la referencia a un estándar (por ejemplo, número UN/SPSC).
- ◆ **Consultar:** Indaga acerca del servicio, usando un conjunto pre definido de parámetros (tal como un URI).
- ◆ **Enlazar:** Enlace en tiempo real del nombre del servicio, el punto final y el URL actual; esto es como conectar la línea telefónica después que el número telefónico actual es discado.
- ◆ **Publicar:** Publica el servicio de negocio en el Servicio de Registro usando una especificación de interface estándar (tal como WSDL).
- ◆ **Despublicar:** Despublica el servicio de negocio en el servicio de registro usando una especificación de interface estándar (tal como WSDL).

En las figuras 2.24 y 2.25 se muestra la notación UML, que describe el caso de uso y el diagrama de secuencia de la asociación que existe entre los actores del sistema desde la perspectiva de los componentes lógicos mostrados en la figura 2.22. La figura 2.25 describe los cinco escenarios de negocios o casos de uso relacionados al uso de los web services. Un servicio de solicitud desea navegar a través de un servicio de registro e indagar o consultar acerca de diversos tipos de servicios de negocios en los que está interesado. Una vez descubierto el apropiado servicio de negocio que le agradaría usar o invocar, el servicio de registro enlaza los servicios con el servicio de proveedor remoto.

Componentes Lógicos	Descripción
Servicio de Solicitud	<p>El consumidor (tales como el comprador, agentes de compra, o proveedores) que solicita un servicio de negocio particular, investigando (descubriendo) un registro de servicio.</p> <p>Por ejemplo: en un comprador que está navegando, un catálogo es un servicio de solicitud para compras. Típicamente, un servicio de solicitud usa un protocolo estándar para descubrir los servicios desde un registro de servicios (por ejemplo ebXML, o UDDI). Una vez que el servicio es encontrado, el servicio de solicitud enlaza el servicio vía unProxy SOAP</p>
Servicio de Agente	<p>Intermediario, agente o broker que negocia, agrega y provee servicios de negocios de una solicitud particular en representación del servicio de solicitud (por ejemplo un agente de compras es un agente intermediario de servicio para los compradores). Típicamente son como los portales para información y servicios, o intercambio de comercio. Los servicios de negocios típicamente residen en repositorios estándar basados en ebXML o UDDI. Un proveedor de servicio puede ser el mismo agente de servicio</p>
Servicio de Proveedor	<p>El proveedor del servicio crea y provee los servicios de negocio o la funcionalidad del sistema (como rol de productor, por ejemplo: un proveedor es un servicio proveedor para distribuir servicios a los compradores). Los servicios de negocios son publicados en lenguajes de descripción estándar tales como WSDL. El servicio de solicitud produce los servicios con interfaces y descripciones que están disponibles mediante estándares en los protocolos (con un apropiado mecanismo de seguridad)</p>
Servicio de Registro	<p>Las páginas amarillas o blancas que albergan todos los servicios de negocios y los proveedores de servicios asociados. Pueden ser un servicio de registro ebXML o UDDI</p>

Cuadro 2.5: Evolución de la tecnología de los web services

El servicio de agente y el servicio de proveedor necesitan pre-registrarse con el servicio de registro, luego del cual, éstos pueden publicar o despublicar sus servicios de negocios, los cuales son usualmente publicados en el servicio de registro.

2.3.4. Estándares y Seguridad en Web Services

Las normas en las que se basa el desarrollo de servicios web son tecnologías en proceso de evolución. Las principales son SOAP (Simple Object Access Protocol, Protocolo de acceso a objetos sencillo), WSDL (Web Services Description Language, Lenguaje de descripción de servicios web), UDDI (Universal Description, Discovery, and Integration, Descripción, detección e integración universales) y WSIL (Web Services Inspection Language, Lenguaje de inspección de servicios web).

Estos estándares se ubican en las capas de la estructura del framework en la tecnología de los Web Services:

Internet (TCP/IP). La red subyacente pública para los servicios es Internet sobre TCP/IP.

Transporte (SOAP sobre HTTP). La capa de transporte subyacente puede ser HTTP, SMTP, SOAP sobre HTTP.

Lenguaje de Descripción de Servicio (WSDL, TP). El servicio de negocio es descrito en un lenguaje común que describe el tipo de servicio y su funcionalidad (por ejemplo puertos, URI, puntos terminales).

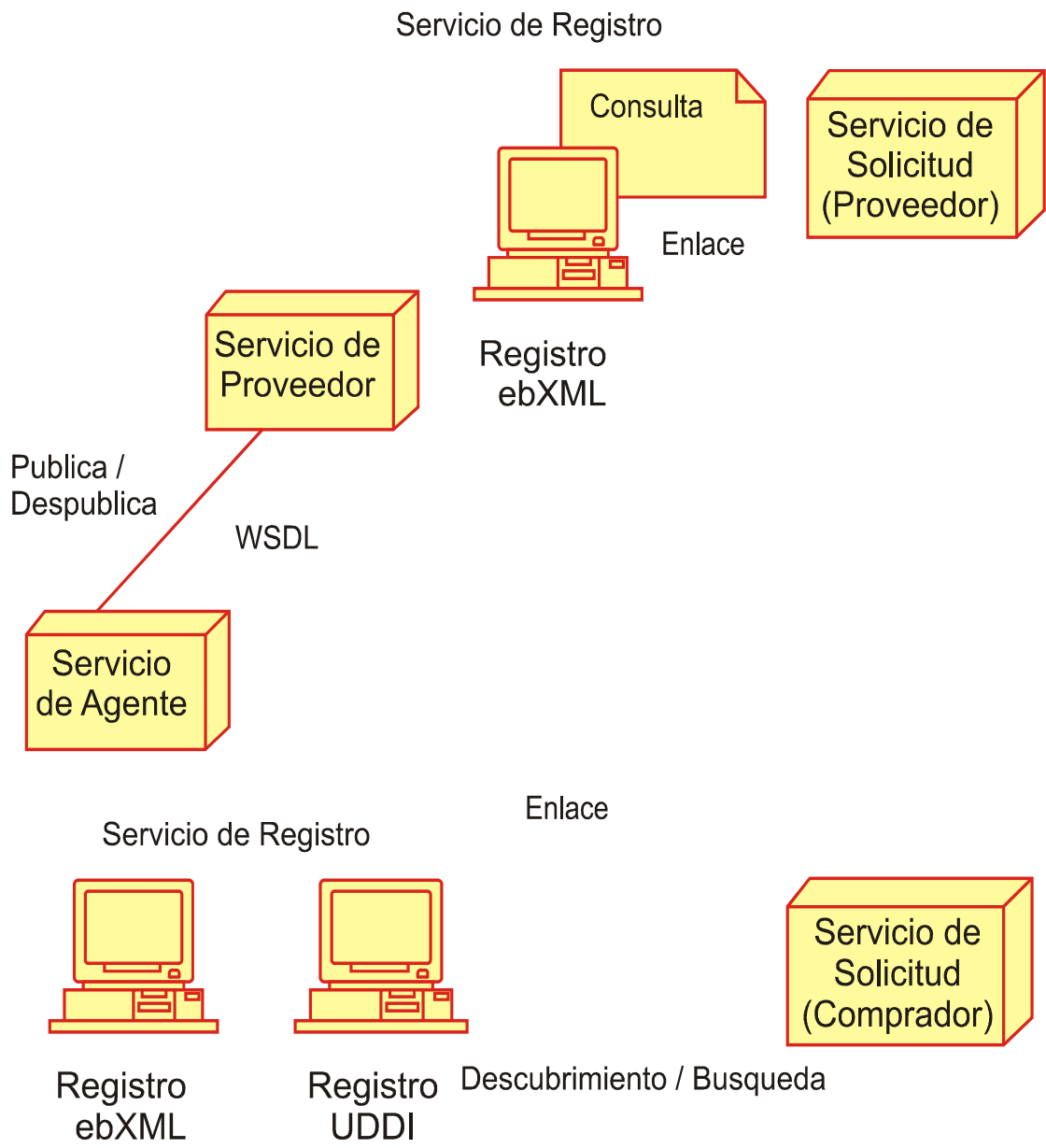


Figura 2.23: Componentes lógicos de los web services

Detalle	Microsoft	Sun	IBM
Soporte al estándar Web Service (SOAP 1.2, WSDL, UDDI)	Sí	Sí	Sí
Arquitectura / Framework de Web Service	.NET	Sun ONE	IBM Web Services
Herramientas y Entorno de Desarrollo Comprensivo	Servidor .NET y Estudio .NET	Servidor de Aplicación Sun ONE, Servidor de Integración Sun ONE, y Sun ONE Studio	Servidor de Aplicación Web Sphere, y Web Sphere Application Developer Studio
Expone EJB como Web Services	No (Pero puede inter operar con otro Web Services exponiendo objetos COM/COM+ a través de inter operabilidad .NET)	Sí	Sí
Soporte a mensajes Síncronos y Asíncronos	DCOM Síncrono	Síncronos: JAXM, SAAJ Asíncronos: JAX-RPC	Síncrono: SAAJ Asíncrono: JAX-RPC
Metodología para Diseñar arquitectura de Web Services	No	Si, Patrones de Diseño J2EE de Sun Tone	Patrones Parciales eBusiness
Cuando usarlo	Solamente Plataforma Wintel PC, PASSPORT / Directorio Activo que ya lo usa	Plataforma Abierta e interoperable	Plataforma WebSphere con extensión propietaria

Cuadro 2.6: Comparación de diversas arquitecturas de web services

Enrutamiento de Transacción (JMS). El enrutamiento de la transacción del contenido de la data y la transacción al siguiente nodo de servicio de negocio; se realiza usando la capa de transporte de menor nivel con o sin garantía de envío de mensaje.

Descubrimiento de Servicio (UDDI, TP). Busca y ubica un servicio de negocio desde los nodos de Registro de Servicio.

Negociación de Servicio (BP, BPEL4WS). Se acuerda en qué cosa va a ser intercambiado y es interoperable entre el servicio que solicita y el servicio que provee.

La mayor parte de la seguridad en los web services está orientada a los mensajes SOAP, ya que el contenido de la data está desacoplada de la capa de transporte. La seguridad punto a punto de los Web Services debe soportar:

- Autenticación
- Autoría y Control de Acceso
- Integridad en la data y las transacciones
- Confidencialidad
- Auditoría

Una visión más amplia de la seguridad en los Web Services cubre [8]:

- Seguridad en la mensajería SOAP (por ejemplo DSIG, WS-security).
- Seguridad en el transporte de los datos/red (por ejemplo el uso de VPN).
- Seguridad transaccional (tal como encriptación de datos, autenticación y control de acceso).
- Seguridad en el servicio de registro (tales como el registro UDDI o WSDL en texto plano).
- Seguridad en el lado del cliente (por ejemplo un navegador UDDI).

- Servicios de seguridad en el kernel y su integración con XML web services (por ejemplo la Especificación de Administración de llaves XML - XKMS).

Por ejemplo, para la seguridad en la mensajería, WS-Security brinda protección a los mensajes SOAP; para la seguridad de la data en la capa de transporte se emplea HTTPS con encriptación incluida, y seguridad en la plataforma. Por ejemplo: en Solaris con políticas de implementación de sistemas de detección de intrusos.

Estos son mecanismos válidos de seguridad para soportar la integridad, privacidad, así como el seguimiento de los datos. Para la seguridad punto a punto de los web services, se requiere añadir la verificación de la identidad tanto del servicio de solicitud como del servicio del proveedor, mediante ciertos requerimientos de autenticación y autorización. Esto involucra el uso de una infraestructura de Llaves Públicas como XKMS, Liberty y SAML

Simple Object Access Protocol (Protocolo de acceso a objetos sencillos, SOAP)

SOAP es un protocolo de mensajes independiente del transporte. Los mensajes SOAP son documentos XML. SOAP utiliza mensajes unidireccionales, aunque es posible combinar mensajes en secuencias de solicitud y respuesta. La especificación SOAP define el formato del mensaje XML, pero no su contenido ni la forma en que se envía. No obstante, SOAP especifica la forma en que los mensajes SOAP se encaminan por medio de HTTP.

Todos los documentos SOAP tienen un elemento raíz <Envelope>. El elemento raíz es el primero de un documento XML, y contiene los demás elementos del documento. Este envelope consta de dos partes: la cabecera (header) y el cuerpo (body).

La cabecera contiene los datos de encaminado o contexto, y puede estar vacía. El cuerpo contiene el mensaje propiamente dicho. También puede estar vacío.

WSDL (Web Services Description Language, Lenguaje de descripción de servicios web)

Los servicios web sólo resultan útiles si otras aplicaciones pueden reconocer qué hacen y la forma de llamarlos. Los desarrolladores deben estar suficientemente informados sobre un servicio web para poder escribir un programa cliente que lo llame. WSDL es un lenguaje basado en XML que se utiliza para definir servicios web y describe la forma de acceder a ellos. Concretamente, describe los contratos de datos y mensajes que ofrece un servicio web. Los desarrolladores deben examinar el documento WSDL de los servicios web para averiguar qué métodos hay disponibles y cómo se realizan las llamadas con los parámetros adecuados.

UDDI (Universal Description, Discovery, and Integration, Descripción, detección e integración universales)

UDDI es una norma en proceso de evolución, que trata sobre la descripción, la publicación y la localización de los servicios web que ofrecen las empresas. Se trata de una especificación para el registro distribuido de información sobre servicios web.

Cuando se ha desarrollado un servicio web y se ha creado un documento WSDL que lo describe, es necesario que exista una forma de proporcionar la información WSDL a los usuarios del servicio web. Cuando un servicio web se publica en un registro UDDI, los posibles usuarios pueden buscar e informarse de la existencia de los servicios web. El contenido de un registro UDDI se puede comparar con las guías telefónicas. En las páginas blancas del registro figuran datos como el nombre, la dirección y el número de teléfono de la empresa que ofrece los servicios web.

Las páginas amarillas identifican el tipo de negocio y lo clasifican por sectores de

actividad. Las páginas verdes proporcionan datos sobre los servicios web que ofrece la empresa.

WSIL (Web Services Inspection Language, Lenguaje de inspección de servicios web)

WSIL, al igual que UDDI, proporciona un método de descubrimiento de servicios para servicios web. A diferencia de UDDI, WSIL utiliza un modelo descentralizado y distribuido, en lugar de un modelo centralizado. Los documentos WSIL son, básicamente, punteros de listas de servicios que permiten a los clientes de servicios web examinar los servicios disponibles. La norma WSIL establece la forma de utilizar documentos con formato XML estándar para inspeccionar sitios web en busca de servicios y series de reglas que determinan la forma en que se proporciona la información. Los documentos WSIL contienen varias referencias a documentos de descripción de servicios anteriores. El proveedor del servicio aloja el documento WSIL de forma que los consumidores pueden encontrar los servicios disponibles.

APIs de Java para llamadas a procedimientos remotos basadas en XML (JAX-RPC)

JAX-RPC define bibliotecas API de Java que los desarrolladores pueden utilizar en sus aplicaciones para desarrollar y consumir servicios web. Mediante JAX-RPC, un cliente Java puede consumir un servicio web que reside en un servidor remoto a través de Internet; incluso aunque el servicio esté escrito en otro idioma y se ejecute en una plataforma diferente. Los clientes que no sean de tipo-Java también pueden consumir servicios JAX-RPC. JAX-RPC utiliza un protocolo de mensajería XML, tal como SOAP, para transmitir una llamada a procedimiento remoto a través de una red. Por ejemplo, un servicio web que devuelve la cotización de unas acciones recibiría una solicitud HTTP SOAP con una llamada a un método realizada desde el cliente. Mediante JAX-RPC, el servicio extrae del mensaje SOAP la llamada al método, la traduce

convenientemente y realiza esa llamada.

A continuación, el servicio utiliza JAX-RPC para convertir la respuesta al método otra vez en un mensaje SOAP y enviar el mensaje de vuelta al cliente. El cliente recibe el mensaje SOAP y utiliza JAX-RPC para traducirlo en una respuesta. El entorno de ejecución JAX-RPC genera stubs y ties, que son clases que permiten la comunicación entre el cliente y el servicio. Un stub, que reside en el cliente, es un objeto local que representa un servicio remoto y actúa como proxy para el servicio. Un objeto tie, que reside en el servidor, actúa como un proxy en el servidor.

2.3.5. Estrategias y Soluciones basadas en Web Services

Existen varios principios generales en lo que a estrategias (a nivel desarrollador) de seguridad en Web Services se refiere.

A continuación, un resumen de estos principios:

- Adoptar el principio de privilegios mínimos. Los procesos que ejecutan código o secuencias de comandos deberán ejecutarse con una cuenta con privilegios mínimos para limitar los posibles daños que podrían producirse si se ve comprometido el proceso. Si un usuario malintencionado consigue insertar código en un proceso de servidor, los privilegios concedidos al proceso determinan en gran medida los tipos de operaciones que puede realizar el usuario. El código que requiere confianza adicional (y privilegios elevados) deberá aislarse en procesos independientes.
- No confiar en los datos introducidos por el usuario. Las aplicaciones deberán validar completamente todos los datos introducidos por el usuario antes de ejecutar operaciones con esos datos. La validación podrá incluir el filtrado de caracteres especiales. Esta medida preventiva

protege a la aplicación de usos incorrectos accidentales o ataques deliberados por parte de usuarios que intentan insertar comandos dañinos en el sistema. Entre los ejemplos más habituales, figuran los ataques de inyección SQL, la inserción de secuencias de comandos y el desbordamiento del búfer.

- Utilizar opciones predeterminadas seguras. Una práctica habitual entre los desarrolladores: es la de utilizar opciones de configuración de seguridad reducida, simplemente para que funcione la aplicación.
- No depender de la seguridad por medio de la oscuridad. Si se intenta ocultar secretos mediante el uso de nombres de variables engañosos o guardarlos en ubicaciones de archivos no habituales, no se estará mejorando la seguridad. En el juego del escondite, es mejor utilizar características de la plataforma o técnicas de probada eficacia para proteger los datos.
- Asumir que los sistemas externos no son seguros.
- Reducir el área expuesta. Evitar exponer información no necesaria para evitar abrir puertas que pueden traer consigo vulnerabilidades adicionales. También se debe tener cuidado con los errores; no exponer más información de la necesaria al devolver un mensaje de error al usuario final.
- Cometer errores de forma segura. Si se produce un error en la aplicación, se debe asegurar de no dejar al descubierto ninguna información confidencial. Tampoco se deberá proporcionar demasiados detalles en los mensajes de error; es decir, no se deberá incluir información que pueda ayudar a un intruso a explotar una vulnerabilidad de la aplicación.

- No olvidar que el alcance de la seguridad lo define su punto más débil. La seguridad es un motivo de preocupación en todos los niveles de la aplicación.

2.4. Aplicación de los Web Services en las Telecomunicaciones

La Plataforma de Red Inteligente UNICA (Universal Network Intelligent Core Architecture) para el servicio Cabiphone, se basa en la tecnología NGN, una red abierta de nueva generación, que puede proveer varios servicios como voz datos y multimedia. NGN no sólo implementa la integración de redes, también implementa la integración de servicios, esto permite a la red de paquetes heredar los servicios e la red TDM y proveer rápidamente nuevos tipos de servicios. La aparición de NGN indica el advenimiento de nuevos tiempos en la red de Telecomunicaciones. UNICA está basada en el lenguaje Java e implementa módulos empleando la técnica de los web services sobre el concepto de computación distribuida. [ver referencia 29]

2.4.1. Arquitectura NGN

La arquitectura NGN está basada en cuatro niveles o capas:

CAPA DE ACCESO. Es la capa donde diferentes usuarios son conectados a la red y el formato de la información original es convertida para que pueda ser transportada por la capa de transporte.

CAPA DE TRANSPORTE. Esta capa provee una uniforme e integrada transmisión con alta confiabilidad, alta QoS (calidad del Servicio) y gran capacidad.

CAPA DE CONTROL DE RED. Implementa el control de la llamada y soporta

los diferentes protocolos de acceso.

CAPA DE APLICACIÓN. Esta capa provee los servicios adicionales.

En la figura 2.31 se aprecia a detalle la implementación de una plataforma de telefonía IP empleando UNICA.

Dentro de los elementos de la capa de acceso podemos citar:

UMG (Universal Media Gateway). Comprende el acceso de Troncales y abonados.

TMG (Trunk Media Gateway). Comprende el acceso de Troncales.

AMG (Access Media Gateway). Comprende el acceso de Abonados POTS y ADSL.

SG (Signaling Gateway). Convierte la señalización TDM SS7 A señalización IP.

El protocolo SIP

SIP es un protocolo de señalización para manejar sesiones de voz datos y multimedia. La característica de este protocolo es que trata con terminales Inteligentes, en contraste con la red telefónica tradicional. Esto permite rápidamente crear nuevos servicios, debido a que la inversión en estos servicios es barata en comparación con los de la telefonía tradicional que es lento y costoso.

Características de SIP

Mobilidad. SIP permite a un cliente registrarse dinámicamente por lo que las llamadas podrían ser ruteadas a él usando una dirección conocida similar al correo electrónico.

Flexible Estructura de Mensaje. La estructura de la mensajería de SIP es de

texto similar a la estructura de http, esto lo hace más flexible y simple.

Separación de Señalización y de Medios. En SIP las rutas de señalización y de medios son totalmente independientes.

Dispositivos SIP

User Agents (UA).- Son dispositivos finales que terminan la señalización SIP. Ellos pueden ser clientes (UAC) que inician la sesión, y los servidores (UAS) que responden a los clientes.

Proxies.- Son dispositivos en la ruta de señalización de los UA que encaminan la llamada hacia su destino.

Register Server.- Son especializados User Agents Server que registran dinámicamente a los dispositivos finales y a su actual localización.

Redirect Server.- Son especializados User Agents Server que responden a los requerimientos de sesión re-direccionando estos a otros dispositivos.

Mensajes básicos de SIP

INVITE.- Invita a participar a una sesión o conferencia, esto puede ser unicast o multicast.

BYE.- Finaliza la participación de un cliente en una llamada.

CANCEL.- Termina una búsqueda.

OPTIONS.- Pregunta a un participante acerca de sus capacidades de media.

ACK.- Es un mensaje de aceptación de la llamada.

REGISTER.- Informa a un SIP server acerca de la localización de un usuario.

Respuestas a mensajes SIP

- 1XX Mensajes de Información(100 trying, 180 ringing, 183 progress)
- 2XX Completamiento exitoso de llamada(200 OK)
- 3XX reenvío de llamada , (302 temporalmente movido, 305 usa proxy)
- 4XX error (403 prohibido)
- 5XX servidor error (500 error interno, 501 no implementado)
- 6XX global failure (606 Not Acceptable)
- Familia de SIP
- SIP: Session Initiation Protocol
- SDP: Session Description Protocol
- H.248 Media GW Control Protocol
- RTP/RTCP Real Time Protocol / Control Protocol
- RSVP Resource ReSerVation protocol

Servicio Cabiphone

Es un servicio dirigido a las cabinas públicas que permite hacer llamadas de larga distancia usando Vo-IP. La habilitación del servicio requiere se instale en la cabina pública un dispositivo IAD (convertor de voz a IP) y un teléfono convencional.

El servicio podrá ser instalado sobre una línea Speedy, convencional o Business, con velocidad mínima de 600 Kbps. A efectos de consumo del servicio se habilita una cuenta pre-pagada asociada a cada teléfono habilitado en la cabina y deberá ser recargada con tarjetas prepago. Este servicio funciona sobre la base del servicio Speedy, y es considerado como un servicio de valor agregado (SVA).

Características

La instalación del servicio es Plug and Play, es decir, no requiere de ningún tipo

de configuración adicional, ya que los equipos vienen pre-configurados de fábrica. No se desconfigura por falta de suministro eléctrico. A efectos de consumo del servicio, se habilita una cuenta pre pagada asociada a cada teléfono instalado en la cabina. La tarificación del servicio es al minuto (redondeo al minuto). Los 10 primeros segundos de la llamada son libres de costo. La recarga del crédito, con el uso de la tarjeta, podrá ser realizada mediante dos formas:

- A través de la web (www.telefonica.com.pe/cabiphone).
- A través del teléfono 13126 (instrucciones de recarga mediante grabación de voz -IVR).

Las recargas de tarjetas son acumulables por cada cuenta prepagada. Ejemplo: El cabinero podrá ingresar una tarjeta de S/.50.00 y posteriormente una de S/.80.00 para una misma cuenta pre-pagada, sin necesidad de haber consumido en su totalidad la tarjeta de S/.50.00. En la página web del cabinero se mostrará la información siguiente:

- Destino, tiempo y costo de la última llamada
- Fecha y hora de establecimiento de llamada
- Tipo de llamada (destino fijo o móvil). Saldo de la cuenta
- Histórico de llamadas
- Histórico de recargas
- Reporte de llamadas efectuadas

En la web se deberá acumular el tráfico histórico de cada cuenta pre pagada (teléfono convencional instalado), se dispondrá de reportes y consultas históricas para el cabinero. El terminal IAD deberá contar con un sistema de seguridad de configuración que sea accesible mediante password o deberá estar encriptado de manera que sólo puedan ser re-programados por personal autorizado.

Procedimiento de Llamada Cabiphone

El cabinero adquiere una tarjeta prepago de la agencia asignada a su zona. El cabinero carga el crédito de su tarjeta a la cuenta asociada al teléfono, ya sea mediante las locuciones del IVR o vía web. Cuando un usuario llega a solicitar el servicio CabiPhone, debe dirigirse al módulo en que se encuentra uno de los dos aparatos telefónicos. El usuario digita el código del país y el número telefónico al que desee llamar. El equipo IAD recibe el dato ingresado y lo envía a la plataforma de Vo-IP, para la autenticación de la cuenta pre-pagada. La plataforma de Vo-IP enruta la llamada hacia el país destino y establece la comunicación con el número solicitado.

El teléfono destino recibe la llamada.

Al finalizar la comunicación: El usuario se dirige al cabinero y solicita el costo de la llamada. El cabinero accesa vía web a la plataforma pre-pago y verifica en su cuenta el destino y tiempo empleado en la llamada y efectúa el cobro al usuario de acuerdo a su tarifa. El usuario efectúa el pago correspondiente y se retira.

Infraestructura Complementaria

IAD

Es un dispositivo que se utiliza para conectar POTS, ISDN y PABX hacia una red IP. El más simple de estos dispositivos es un adaptador de teléfono convencional (2 puertos FXS) a ethernet (1 puerto RJ45). Este dispositivo se comunica via el protocolo SIP con servidor SIP para su registro y autenticación. En Cabiphone, las funciones de los IADs serán el de generar llamadas de LDI

por los dos teléfonos convencionales conectados al IAD.

Session Border Gateway

Este equipo fue concebido principalmente para resolver el problema que surgió a partir de la implementación de servicios IP en redes que utilizan NAT (Network Address Translation). El border es cualquier frontera de una red IP, frontera entre proveedores de servicios y clientes/usuarios, o entre dos proveedores de servicios.

La función control satisface los nuevos requerimientos de seguridad, servicios confiables y cumplimientos regulatorios en las fronteras de la red.

Funciones:

Las funciones que realiza el SBG son las siguientes:

- La función NAT/SIP.
- Realiza el interworking de SIP a H.323.
- Brinda funciones de SLA (Services Level Assurance).
- Otorga niveles de calidad de servicio QoS.

Media Gateway (MG)

En Cabiphone, las funciones del MG serán: Servir de interfaz de voz entre la red IP y la red PSTN, quiere decir que convierte los paquetes de voz RTP de la red IP a una canal de audio TDM, para así llegar a la red PSTN Internacional. Este Media Gateway estará controlado por el MGC a través del protocolo MeGaCo.

Softswitch: Realiza el registro y administración de los recursos del Media Gateway (MG), control de llamadas, control de la media (voz, datos, video), así como el interfuncionamiento entre los protocolos de señalización.

Función:

Las funciones que realiza el Softswitch son las siguientes:

Controlar en su totalidad las llamadas a cursar, quiere decir que puede realizar funciones de tránsito de señalización; así como realizar adaptaciones para completar las llamadas de acuerdo al tipo de servicio a brindar.

De este elemento se pueden rescatar los registros de tasación y los índices de tráfico.

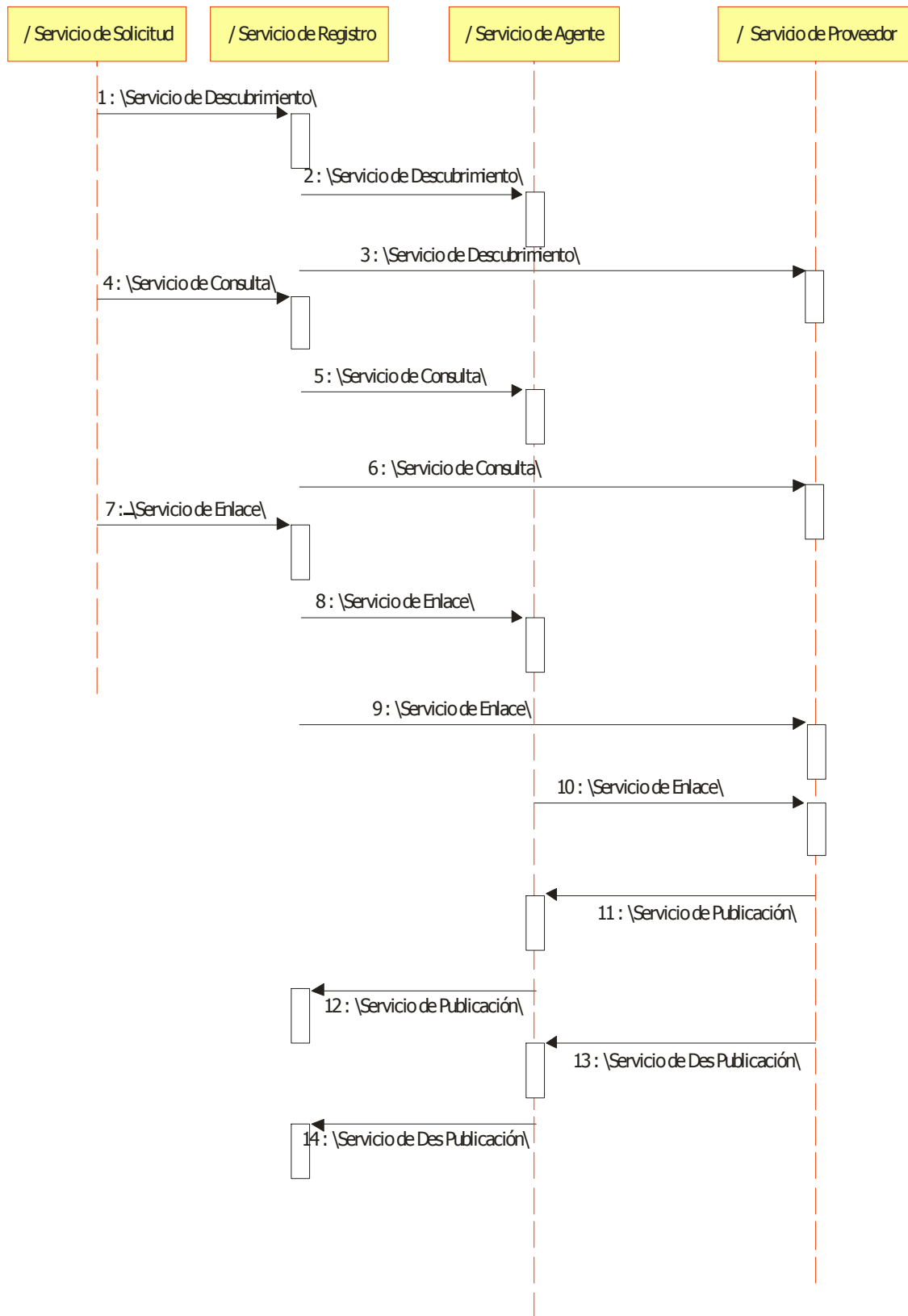


Figura 2.24: Diagrama de secuencia de los Web Services

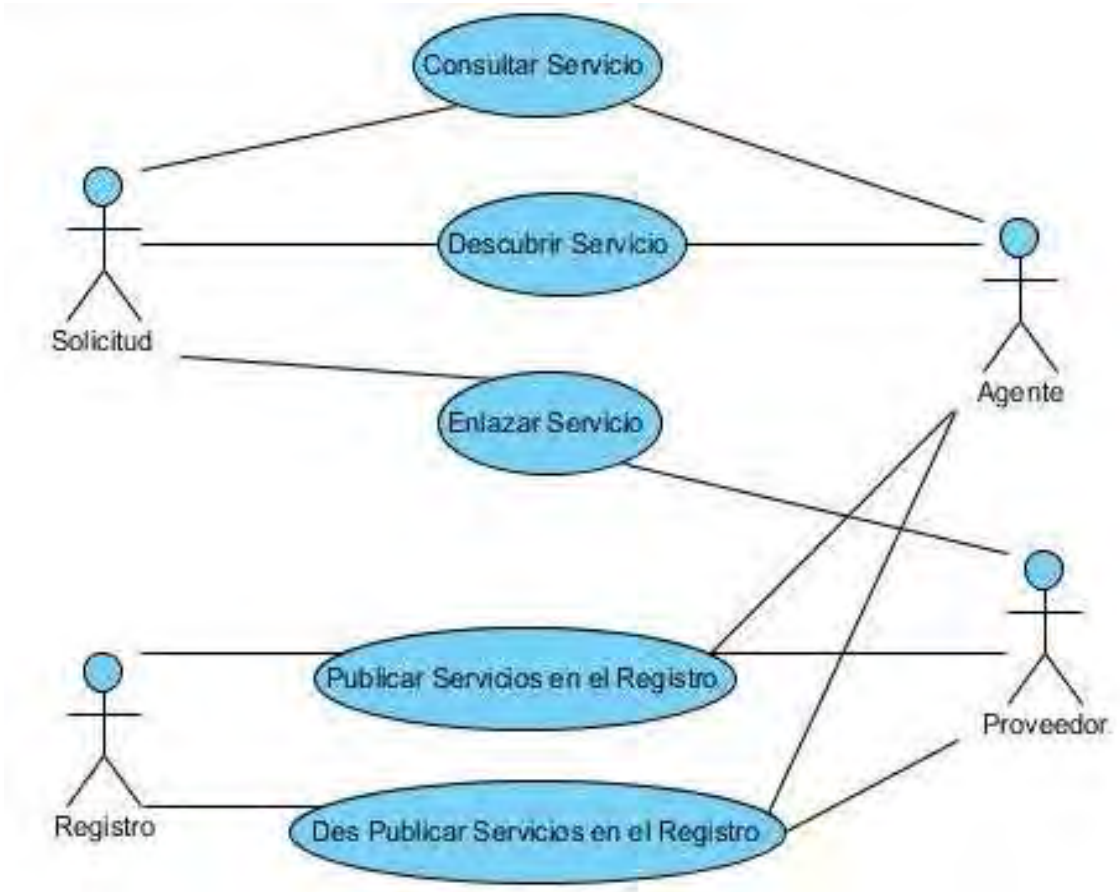


Figura 2.25: Caso de uso de los Web Services

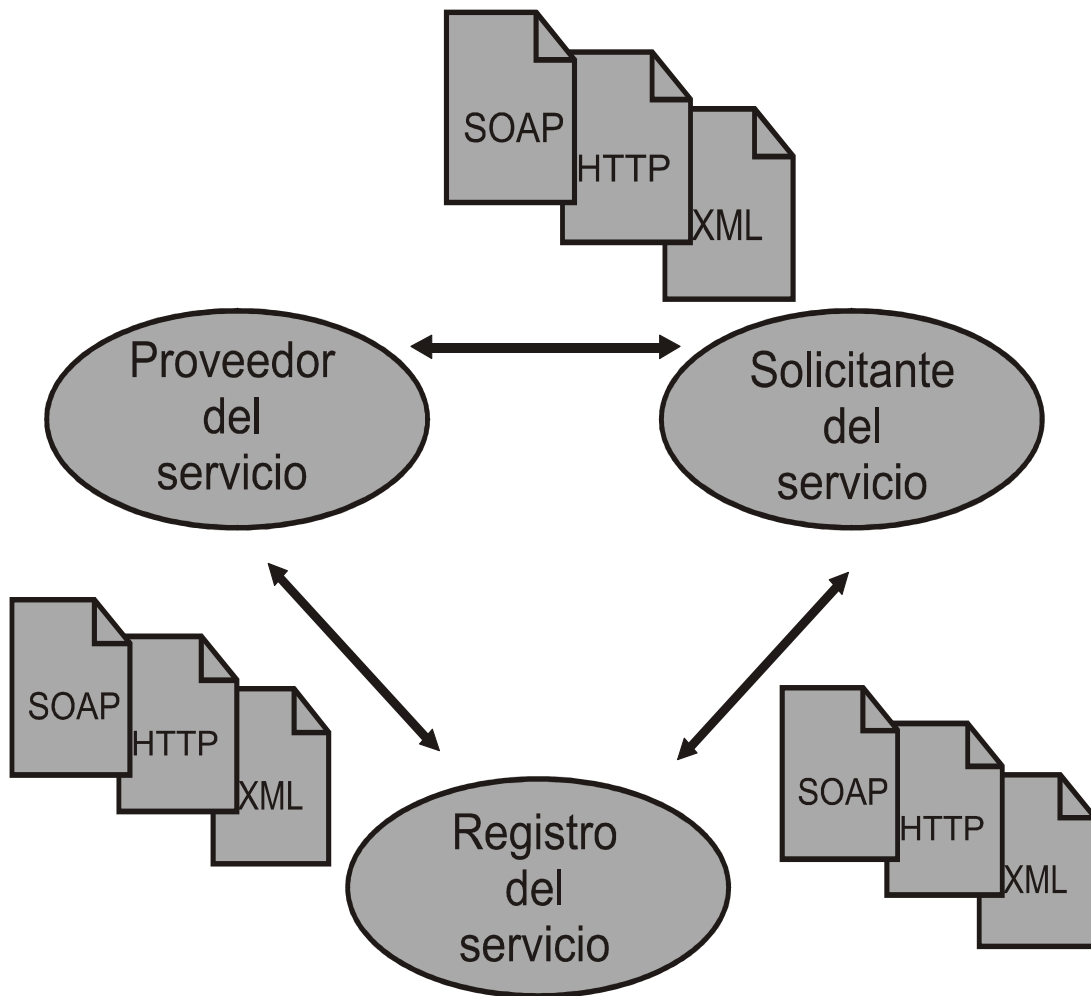


Figura 2.26: Protocolo de acceso a objetos sencillo

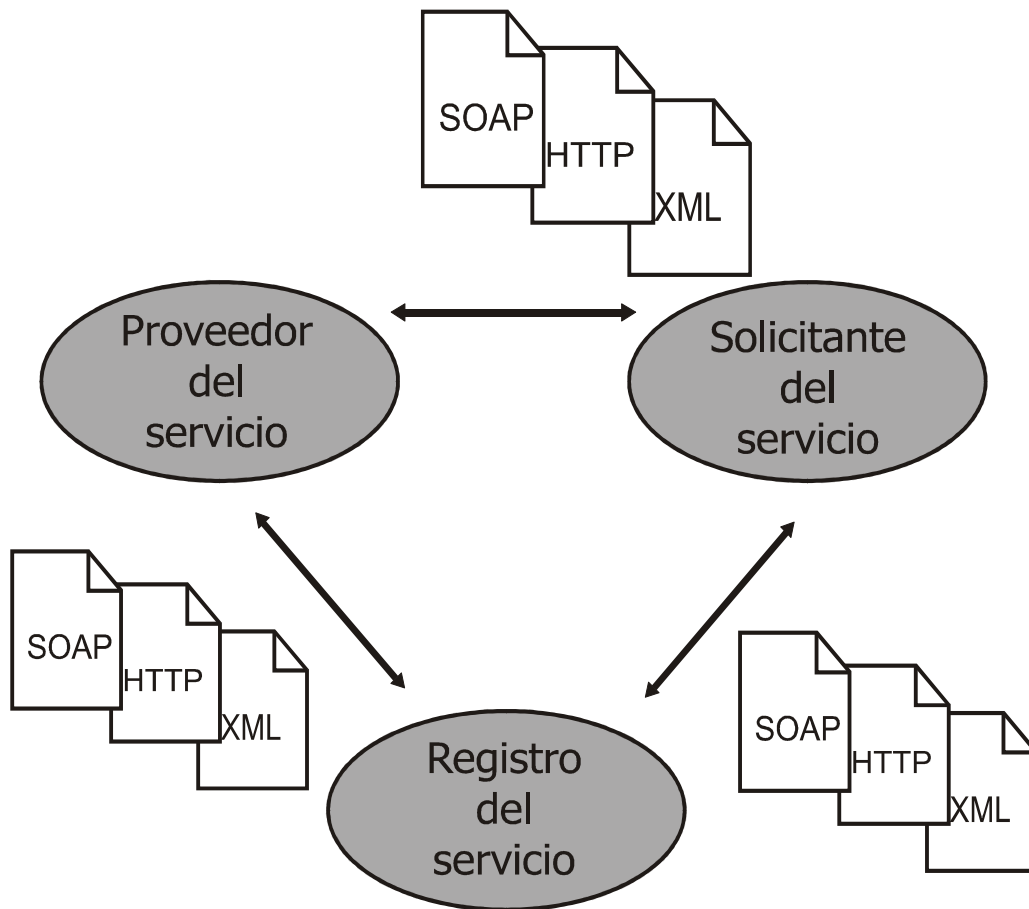


Figura 2.27: Lenguaje de descripción de los Web Services

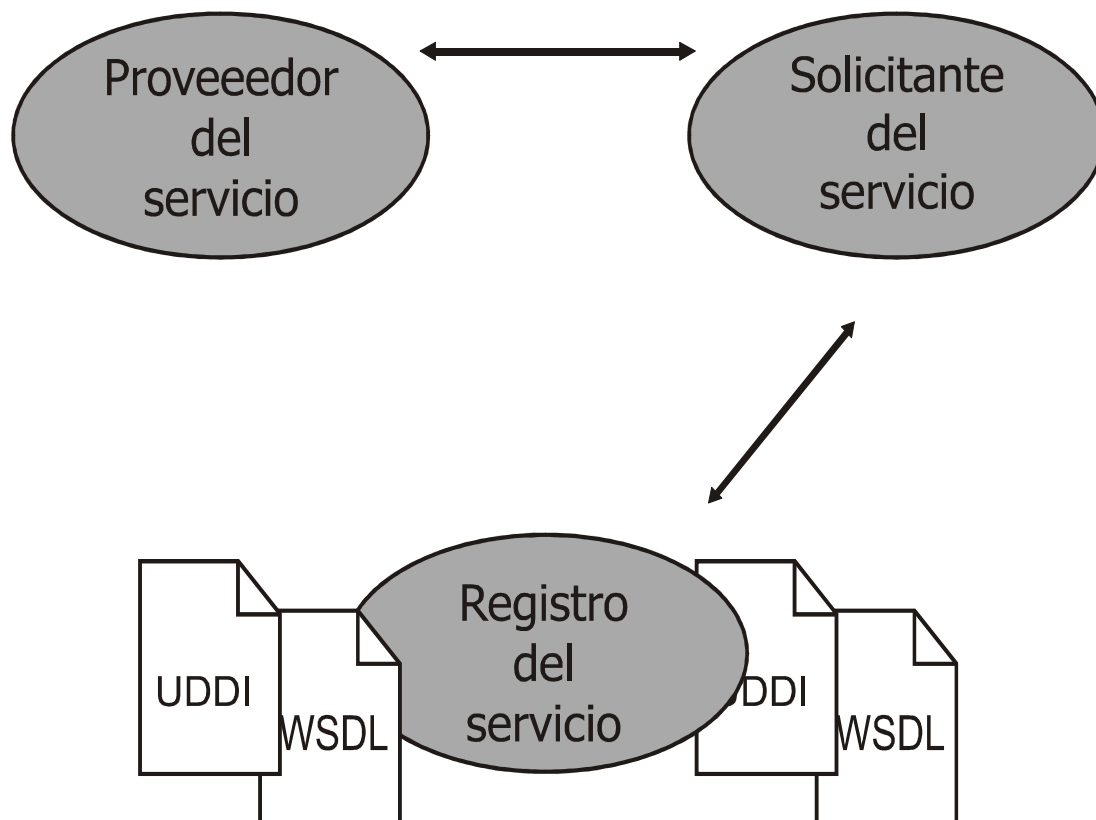


Figura 2.28: Descripción, detección e integración universales

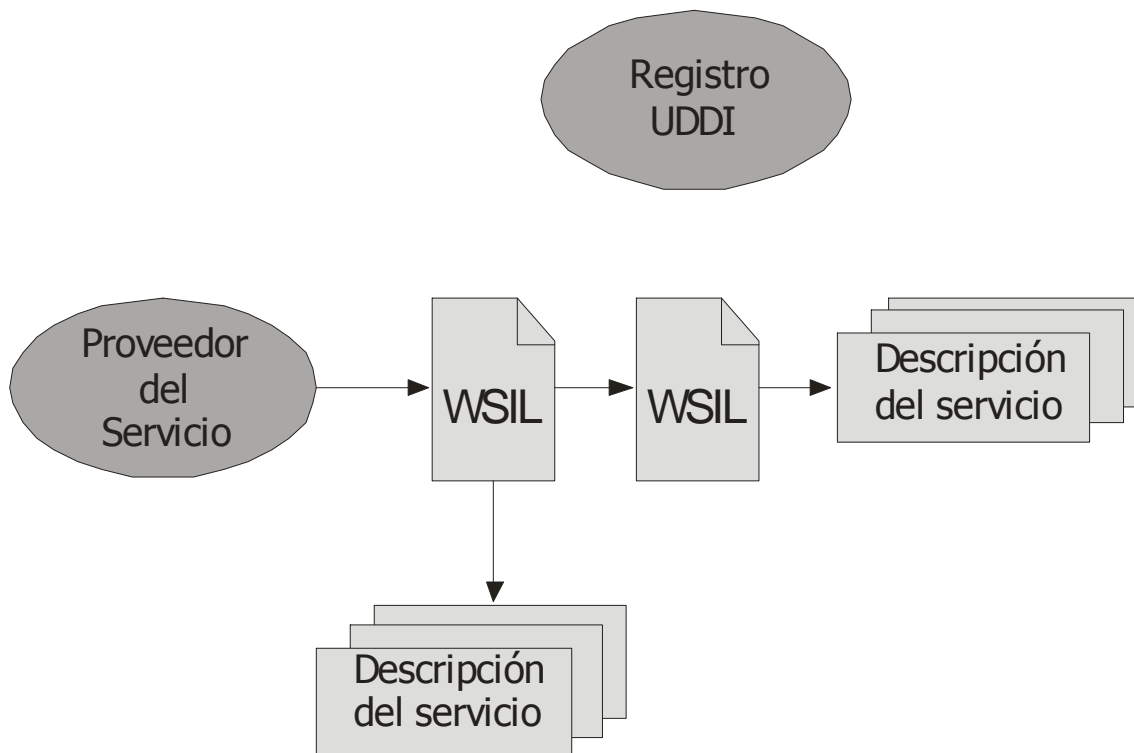


Figura 2.29: Lenguaje de inspección de los Web Services

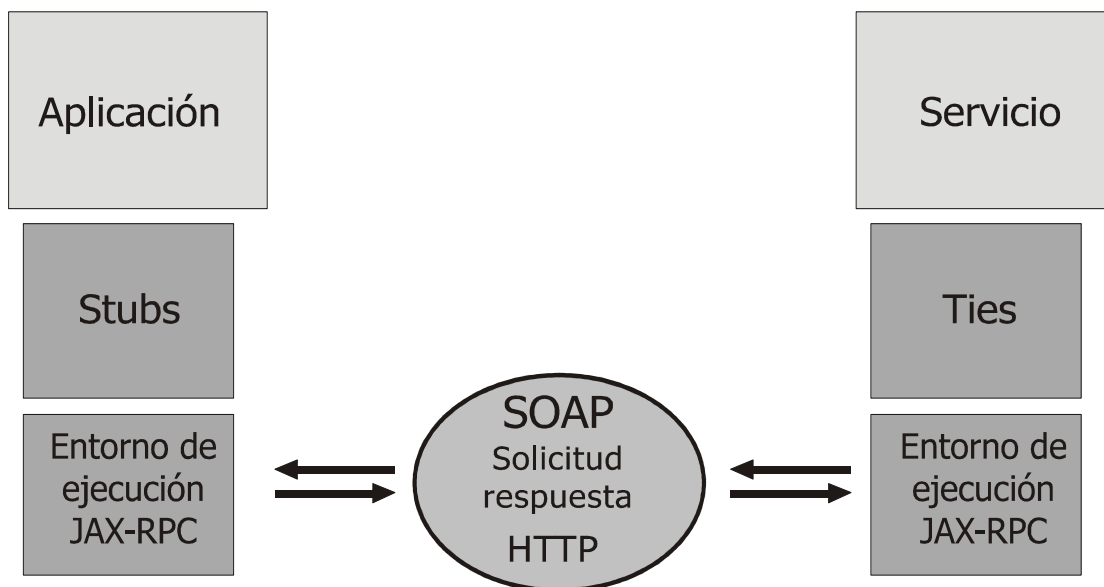
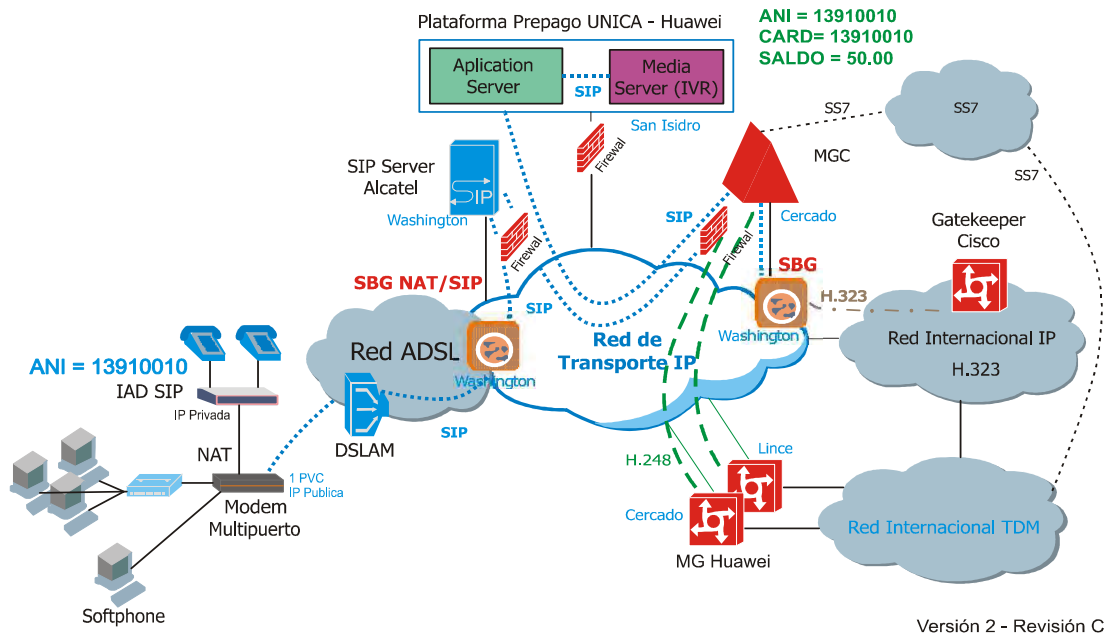


Figura 2.30: API de Java para llamadas a procedimientos remotos basadas en XML

DIAGRAMA DE RED PROYECTO CABIPHONE



Telefónica del Perú S.A.A.

Figura 2.31: Esquema de despliegue de la plataforma UNICA que utiliza una arquitectura basada en Web Services

Capítulo 3

Diseño e Implementación del Sistema Automatizado de la Biblioteca Digital

En esta sección se limitará el alcance del desarrollo del Sistema de Información de Bibliotecas Digitales Distribuidas, que es una evolución sobre los sistemas de bibliotecas tradicionales ya que es marcadamente distribuido e interoperable con otras aplicaciones a través del uso de estándares de Internet con tecnología Web Services a través de los flujos del proceso modelado sobre la notación BPMN.

Estas características fundamentales permitirán crear un entorno que estimule la comunicación entre lectores, autores, empresas editoras y bibliotecas de todo el mundo ya que fácilmente pueden integrarse entre sí a través de los descriptores WSDL que pertenecen a la capa de servicio, pues éstos se integran con la capa de dominio y la capa de persistencia en forma transparente.

La Arquitectura Orientada a Regulaciones ROA es un entorno sistémico de desarrollo creado con el propósito de implementar procesos de negocios orientado a la automatización digital utilizando tecnologías emergentes tales como BPMN y Web Services sobre SOA. Para implementar esta arquitectura se requiere de cuatro pilares orgánicos:

- 1) Metodologías de Dirección de la Arquitectura ROA
- 2) Tecnologías de Soporte de la Arquitectura ROA
- 3) Infraestructuras de Regulación de la Arquitectura ROA
- 4) Líneas de Desarrollo de la Arquitectura ROA

En lo que respecta a las líneas de desarrollo, existen diversas corrientes emergentes tales como la inteligencia de negocios, sistemas de gestión de proceso de negocios, redes sociales, entre otras. En el proyecto de tesis se utiliza la plataforma de Modelado de Proceso de Negocio de Intalio por ser una solución de código abierto difundido ampliamente en la comunidad de TI.

Estos cuatro pilares son la base del entorno sistémico, para lo cual se ha adaptado gran parte de las mejores prácticas recomendadas a nivel internacional. Cuando desarrollamos un proyecto de tecnología de información, se requieren los cuatro componentes, esto es:

Metodología de Dirección: La arquitectura ROA, nos permite fusionar nuevos paradigmas de gestión de proyectos. Como prueba de concepto se ha fusionado las metodologías PMBOK del PMI y RUP, la primera orientado a la gestión de proyectos en un contexto amplio, la segunda orientada específicamente a proyectos de desarrollo de software.

Con este enfoque a continuación se muestra un cronograma referencial de la arquitectura ROA, en el cual se contemplan los alcances de un proyecto de Modelado de Procesos de Negocios sobre la arquitectura planteada. Asimismo se muestra un cuadro resumen de actividades a realizar.

Cronograma Referencial para el Despliegue de un Proyecto basado en la Arquitectura ROA:

Etapas de Iniciación del Proyecto

- Reunión de Asignación del Grupo de Trabajo
- Definición de la Arquitectura de Procesos
- Definición de la Arquitectura de Aplicaciones
- Definición de la Arquitectura de Bases de Datos

Etapas de Realización del Proyecto

- Configuración del Entorno de Desarrollo
- Patrones de Procesos de Capa Lógica de Negocio
- Patrones de Procesos de Capa de Vistas
- Patrones de Procesos de Capa de Persistencia
- Integración de Capas BPM

Etapas de Finalización del Proyecto

- Prueba Interna de Integración
- Prueba Piloto
- Cierre del Proyecto

Listado de Tareas para el Despliegue de un Proyecto basado en la Arquitectura ROA:

- Definición de las Fases del Proceso
- Definición del Macro Proceso
- Definición del Proceso
- Definición de Formularios GUI
- Definición de Esquemas XSD
- Definición de Acceso a Datos SQL
- Definición de Servicios Web WSDL

Dentro del contexto metodológico de dirección ROA, existe un flujo de trabajo al momento de implementar la metodología, el cual se focaliza en el modelado del proceso. Dicho flujo de trabajo se clasifica en actividades orientadas a implementar el proceso conceptual y regulado.

Requerimientos no funcionales:

- Disponibilidad
- Ventana de Tiempo Operativa
- Tiempo de Respuesta

- Pico de Carga

Decisiones Arquitectónicas:

- Modo de publicación de descriptores de los servicios
- Forma de exposición de los servicios
- Modo de formato de mensajes
- Definición del cómo y dónde se efectuará la transformación de los mensajes
- División de responsabilidad entre el ESB y los componentes empresariales

Un proceso de negocio es un flujo coordinado de actividades cuyas acciones se basan en decisiones, y conducidos por participantes, que interactúan sobre la data, información y conocimiento para lograr un resultado. Todo proceso de negocio debe estar soportado por una Gestión de Procesos de Negocios BPM, que permita enlazar decisiones entre las personas y el sistema en un contexto empresarial. Es decir, un proceso soporta un modelo de negocio que se adapta a los cambios.

Tipo de Modelado	Nivel	Roles
Proceso Conceptual	Nivel Funcional	Analista de Procesos Funcionales
	Nivel Operativo	Analista de Procesos Operativos Implementador de Documentación
Proceso Regulado	Pre Realizable	Arquitecto de Base de Datos Arquitecto de Web Services Implementador de Pruebas Auditor de Calidad de Pruebas
	Realizable	Integrador de Sistemas Implementador de Sistemas

Cuadro 3.1: Niveles de modelado ROA - BPM

Para dar soporte a la implementación ROA, se definen perfiles de acuerdo al alcance del proyecto. En un caso general los perfiles podrían definirse del siguiente modo:

- 1) Analista de Procesos Funcionales
- 2) Analista de Procesos Operativos
- 3) Implementador de Documentación
- 4) Arquitecto de Bases de Datos
- 5) Arquitecto de Servicios Web
- 6) Implementador de Pruebas
- 7) Integrador de Sistemas
- 8) Auditor de Calidad de Software
- 9) Implementador de Sistemas

Es importante destacar que los entregables del nivel funcional corresponden al Analista de Procesos Funcionales, en cuyas reuniones más están orientadas a perfiles de gerencia, cuyo objetivo es formular y definir el proceso de negocio, las reglas de negocio y las decisiones del negocio. (ver Cuadro 3.1)

Una Regla de Negocio es un mediador de información en sistemas computacionales para la toma de decisión en los participantes del proceso tales como gerentes, empleados y personal de ventas. Es decir, una regla de negocio: es la mínima unidad lógica que usa la data y el conocimiento para evaluar parte de una proposición acerca de la decisión de un proceso. El punto de encuentro entre un proceso y una decisión es la regla de negocio. En sistemas cuyos procesos son más complejos, se requiere un gestor especializado de decisiones conocido como Gestor de Decisiones Empresariales (EDM - Enterprise Decision Management).

Cada uno de estos roles está enlazado con alguno de los niveles del proceso de modelado BPM. Desde el punto de vista ROA, el Analista de Procesos es el encargado de modelar las Fases y el Macro Proceso base del proyecto.

Es importante tener en cuenta los archivos de configuración establecidos en el Servidor de Aplicación para Automatización de Procesos de Negocios Intalio:

- Consola de Administración:
 - [http://\[server\]/bpms-console](http://[server]/bpms-console)
- Interfaz de Usuarios:
 - [http://\[server\]/ui-fw](http://[server]/ui-fw)
- Depósito de Reportes:
 - [http://\[server\]/bam](http://[server]/bam)
- Motor de Ejecución:
 - [http://\[server\]/ode](http://[server]/ode)

3.1. Análisis de la Biblioteca Digital Distribuida

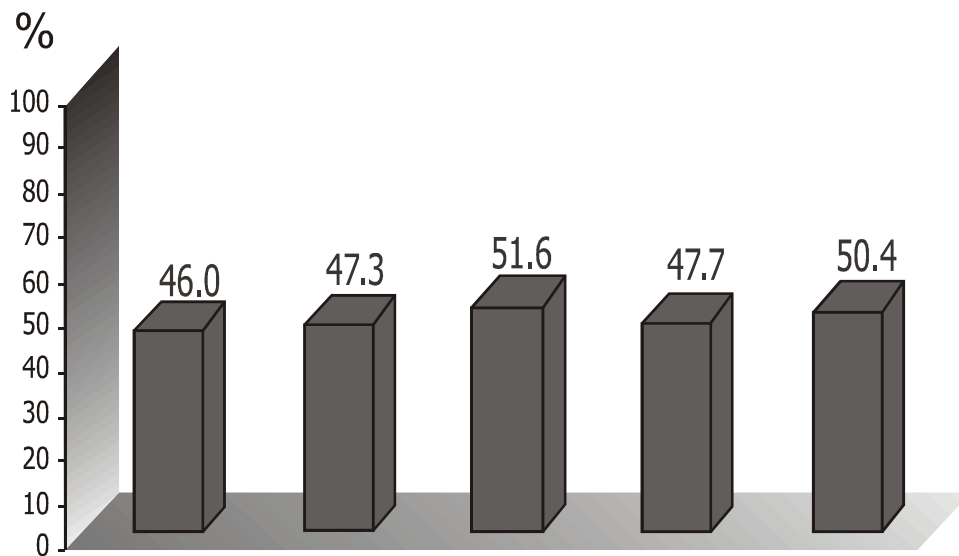
El prototipo del sistema de biblioteca digital distribuida sólo cubre el caso de uso de Registro de Usuario de Biblioteca, ello debido a que en el diseño BPM se modelan e implementan procesos de apoyo académico. Se ha descartado implementar la búsqueda de contenidos educativos referida a libros digitales debido a que el prototipo está abocado a la prueba de concepto y no a la realización total del proyecto.

Para la mayoría de los casos se suele implementar una plataforma de Aprendizaje Electrónico sobre plataformas de E-Learning ó Portales y éste a su vez acoplado a una plataforma BPMS para complementar con la implementación de procesos de apoyo académico. Las instituciones que adopten este sistema (por ejemplo: colegios, universidades e institutos) se harán cargo sólo de la administración local de su nodo, responsabilizándose de mantenerlo en funcionamiento y registrando en ella los contenidos educativos que posea.

3.1.1. Consulta a fuentes de información apropiada e identificación de problemas

Los Censos Nacionales de Población y Vivienda de 1993, realizados por

el Instituto Nacional de Estadística e Informática (INEI) constituyen una valiosa fuente de información sobre diversos aspectos de la realidad nacional, y permiten la realización de investigaciones de apoyo a la toma de decisiones de entidades públicas, privadas e internacionales que trabajan en procura del desarrollo del país. En el presente trabajo, se analiza algunos factores predominantes [ver referencia 24] con respecto al Proyecto Huascarán con ROA.



FUENTE: INEI - ENCUESTA NACIONAL DE HOGARES IV TRIM. 1997-2001

Figura 3.1: Tasa de Asistencia de Educación Inicial de la población de 3 a 5 años de edad, 1997-2001

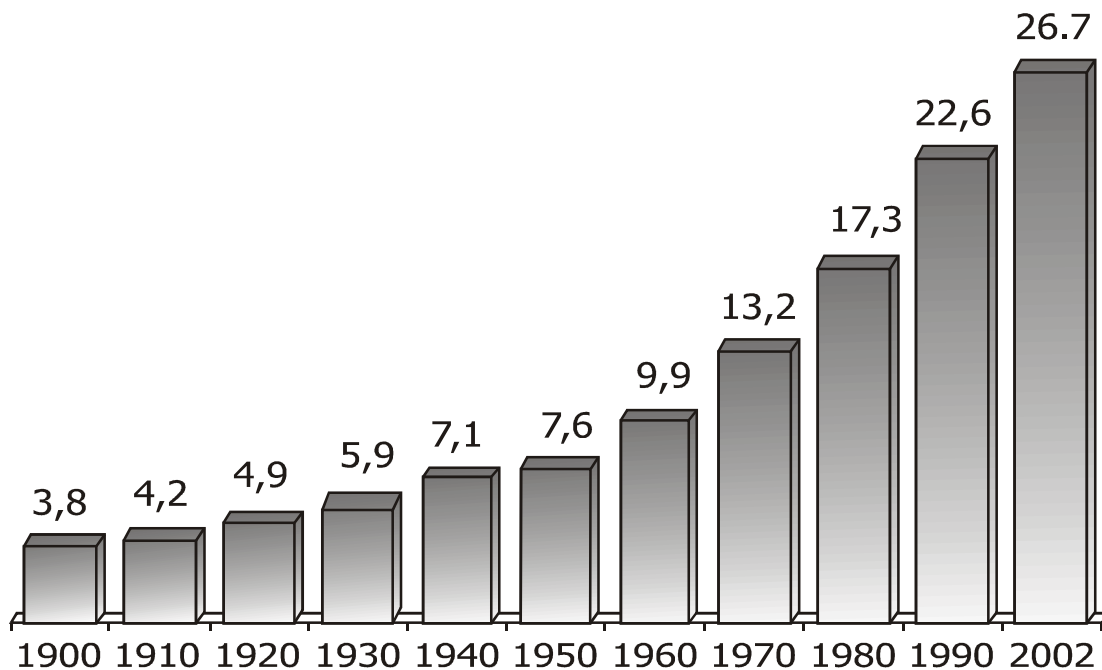
Al hablar de análisis del espacio peruano, nos estamos refiriendo al espacio como soporte físico de la actividad humana y de su intervención, convirtiéndose en uno de los productos de la actividad social cuyo estudio es tan indispensable para el conocimiento de la sociedad. El espacio, producto social es así, el reflejo de la sociedad que lo moldea.

Evaluar el espacio nacional significa decir lo que es la sociedad peruana y mostrar en qué elementos del espacio debería el ordenamiento territorial concentrar sus esfuerzos y subrayar los frenos de naturaleza espacial que se opondrían a los cambios deseados, para alcanzar el éxito del proyecto

Huascarán en un posible escenario ROA.

El presente estudio revela así las estructuras espaciales del territorio peruano y su transformación mediante las dinámicas demográficas, sociales y económicas como se puede apreciar en el censo de población de 1940 con sus estimaciones de población. (ver Figura 3.2)

PERU: EVOLUCION DE LA POBLACION EN EL TIEMPO (En millones de personas)



FUENTE: INEI - Censo de Población 1940 - Estimaciones de Población 1950 - 2050

Figura 3.2: Evolución de la población en el tiempo

3.1.2. Análisis de posibles causas de los problemas detectados en el sistema convencional

La crisis del sistema educativo se vive en diversas esferas: económica, técnica y personal. Un alto porcentaje de estudiantes egresa del sistema sin dominar conocimientos fundamentales para una vida productiva en la sociedad.

Muchos docentes experimentan una frustración creciente, con una sensación de desgaste improductivo. Es probable que ni los estudiantes ni los docentes sean responsables por esta situación, sino que ésta es una consecuencia de una definición totalmente inadecuada de la estructura educativa. Para solucionar esta situación existen diversas propuestas:

- 1) La presente educación síncrona, que supone que todos los estudiantes deben aprender lo mismo al mismo tiempo, debe ser sustituida por una educación asíncrona, que permita el aprendizaje de acuerdo al ritmo y posibilidades de cada estudiante.
- 2) La educación masiva, herencia de la Revolución Industrial, debe pasar a ser personalizada.
- 3) La enseñanza actual, centrada en la exposición del docente, debe pasar el énfasis a lo que el estudiante realiza.
- 4) El aprendizaje debe ser planeado no sólo para los estudiantes, sino también para los docentes. Estos cambios son posibles hoy en día gracias a los avances realizados en la tecnología y en la investigación educativa, los cuales nos permiten proponer, quizás por primera vez en la historia, una solución sistémica para los problemas planteados.

La consecución de logros generados por un cambio real en el sistema educativo, comienza por el desarrollo de cuadros locales de expertos en informática educativa, que puedan ser los catalizadores del cambio. El paso siguiente será la implementación de proyectos piloto, coincidentes con la capacitación del personal docente que participará en los mismos. Recomendamos que estos proyectos piloto se ubiquen en el entorno de los institutos de formación docente de modo que, a mediano y largo plazo, todo egresado de tales institutos esté en condiciones de integrar la informática educativa con sus actividades docentes.

El habernos desarrollado dentro de una estructura determinada nos hace pensar que así deben ser las cosas, y que así lo hayan sido a lo largo de los tiempos. Pero además, en el caso del sistema educativo, la estructura que vivimos no ha sido la norma sino la excepción. A lo largo de la historia, la educación ha sido personalizada y que, sólo a partir de la Revolución Industrial, se adoptó la educación de producción en serie que hoy sufrimos. La investigación cognitiva de las últimas décadas nos urge a cambiar el estilo de enseñanza, y esbozar un proyecto de cambio que atienda a las necesidades de los educandos, de los docentes, y de la sociedad en general. [ver referencia 25]

La tecnología no es para hacer de mejor forma lo que veníamos haciendo antes, sino para hacer nuevas cosas y mejorar el aprendizaje. El cambio educativo se debe realizar en torno a un sistema que estimule al maestro a convertirse en un experto y entusiasta de un área que el estudiante admira.

Actualmente el papel que cumple la tecnología como apoyo a la labor educativa excede en buena medida la formación del maestro.

En términos pedagógicos, la tecnología replantea tanto los escenarios como los roles, los métodos de enseñanza, la producción misma de conocimiento y saber pedagógico.

A continuación se detalla las transformaciones (Cambio de Paradigma) que se van a ir produciendo progresivamente en la escuela como consecuencia de las nuevas exigencias que va a plantear la sociedad de la información. [ver referencia 25]

Escuelas Convencionales Usualmente tienen las siguientes características:

Aprendizaje: Los estudiantes aprenden absorbiendo información y

destrezas escuchando a los profesores y leyendo los libros de textos.

Enseñanza: Los profesores introducen la información, suministran ejercicios para su destreza práctica y para memorizar información, y examinan la habilidad de los estudiantes para recordar la lección.

Currículo: El currículo enfatiza el dominio de destreza y conceptos a través del currículum dividido en áreas de contenido. Los estudiantes son calificados bien o mal de acuerdo con test específicos.

Clases: Las clases son por lo general escenarios aislados donde los profesores distribuyen información y los estudiantes practican su destreza y responden preguntas.

Evaluación: La evaluación se centra sobre preguntas cortas y test fáciles que enfatizan la habilidad de recuerdo de información antes que la comprensión o la aplicación en un camino significativo.

Tecnología: Las tecnologías educativas tradicionalmente han incluido lápices y papeles, pizarra, libros de texto, y otros recursos que ayudan a los estudiantes a desarrollar destrezas básicas, conceptos y generalizaciones.

Escuelas Reestructuradas. A diferencia de las escuelas convencionales, sus características varían en:

Aprendizaje: Los estudiantes aprenden construyendo su propio conocimiento a través de la investigación, experiencia-profesores, libros de texto y otros recursos.

Enseñanza: Los profesores instruyen a los estudiantes en las actividades

que requieren pensar críticamente, resolver problemas y buscar respuestas a sus propias cuestiones. Los profesores sirven como modelo a los estudiantes.

Currículo: El currículo fomenta la investigación del estudiante, y es diseñado para invitar a los estudiantes a resolver problemas reales que están incluidos dentro de todas las áreas de contenido. El conocimiento profundo de los conceptos importantes es enfatizado.

Clases: Las clases son lugares multipropósito donde los estudiantes se ocupan de la investigación y la solución de problemas en actividades relacionadas a tópicos específicos de enseñanza. El foco es sobre la cooperación y equipos de trabajo.

Evaluación: La evaluación se centra sobre la demostración del estudiante sobre su habilidad para expresar, aplicar y defender conocimientos y destrezas. Los estudiantes también tienen la habilidad de autoevaluarse y sobre todo las actitudes en una mejora continua y en la profundidad de procesamiento.

Tecnología: Una variedad de tecnologías están ahora disponibles para asistir a los estudiantes en la creación de conocimientos y destrezas. Muchas de estas tecnologías pueden ahora soportar investigaciones, análisis, resolución de problemas y procesos de comunicación más efectivos que los recursos tradicionales.

3.1.3. Análisis de alternativas de solución

Las aplicaciones distribuidas requieren acceso a una serie de servicios de empresa. Los servicios típicos incluyen procesamiento de transacciones, acceso a bases de datos, mensajería, multihilos (multithreading), etc. Una alternativa tradicional constaría en utilizar la arquitectura de J2EE que unifica el acceso a

estos servicios en una API de servicios de empresa. Sin embargo, en lugar de ello se ha decidido utilizar una arquitectura orientada a regulaciones sobre un Sistema de Gestión de Procesos de Negocios, que reemplaza J2EE con BPEL, y el modelado UML con BPMN.

En este contexto, se optó por escoger la tecnología de computación distribuida Web Services y descartar Peer-to-Peer debido al contexto nacional del sector educación [ver referencias 25, 26]. El prototipo propuesto permite la comunicación entre aplicaciones heterogéneas (escritas en distintos lenguajes y ejecutándose en múltiples plataformas) mediante el intercambio de mensajes basados en XML, a través de servidores centralizados en cada nodo o centro educativo.

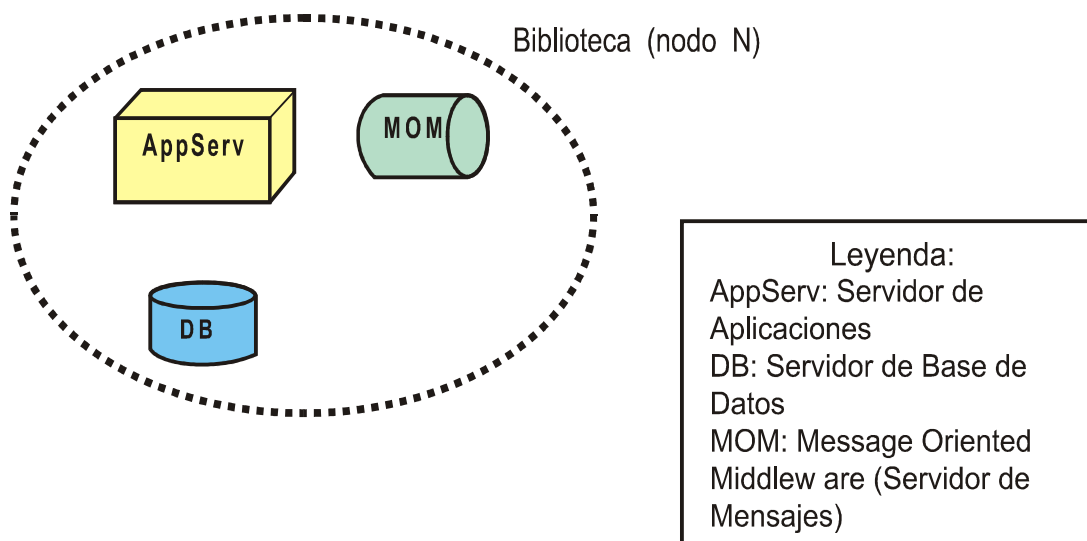


Figura 3.3: Arquitectura básica de un nodo aislado de la Biblioteca Digital Distribuida

Para la implementación de Dexter's Lib se decidió usar una plataforma de gestión de procesos de negocios empresariales BPM, que permita modelar los procesos de atención de la biblioteca digital. Una arquitectura tradicional aislada de un nodo se muestra en la Figura 3.3; siendo el Servidor de Aplicaciones el entorno de ejecución de la aplicación, el Servidor de Base de Datos el

repositorio de la información a suministrar, y el Servidor de Mensajes el encargado de transmitir y recibir mensajes asíncronos entre varios nodos. En la Figura 3.4 se muestra un Nodo típico de la Biblioteca Digital, en el cual se publican los servicios mediante procesos de negocios vía web services a través de las interfaces de trabajo AJAX de la plataforma Intalio.

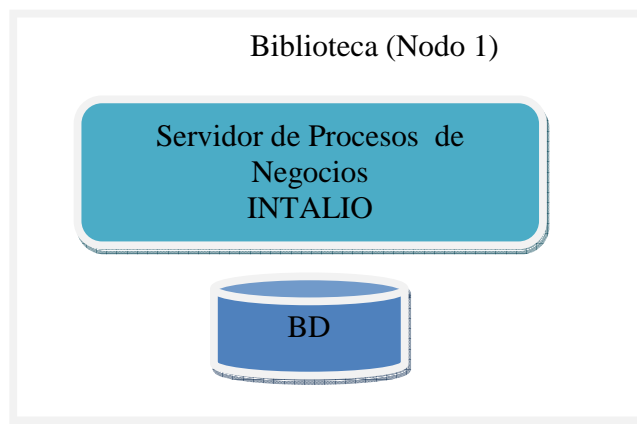


Figura 3.4: Arquitectura de un nodo BPM de la Biblioteca Digital Distribuida

3.1.4. Establecimiento del papel del computador

Partiendo del principio filosófico que estima que el desarrollo humano integral es la base esencial del programa formativo en el cual jóvenes y adultos, mujeres y hombres independientemente del nivel educativo alcanzado y de otros factores como género, razas, edad, ideología o condiciones personales, son seres en permanente evolución y perfeccionamiento, dueños de unas capacidades y potencialidades que los habilitan como sujetos activos y participantes en su proceso de formación y con grandes expectativas con frente al mejoramiento de su calidad de vida y que abre las puertas a la creatividad donde podrán participar activamente en los procesos y transformaciones económicas, sociales, políticas científicas y culturales; situación que permitirá la contribución con aplicación de los conocimientos tecnológicos recientes, científicos y humanísticos que estimulan en fomento del liderazgo positivo, la

participación ciudadana y la construcción de un nuevo modelo de sociedad verdaderamente democrática.

En este contexto el computador como tal va a sufrir una evolución conceptual en el cual los dispositivos de comunicación fijos o móviles van a tener mayores recursos de CPU, el cual va a permitir desarrollar sistemas embebidos que se integren con los procesos de información de instituciones académicas.

Por otra parte las TIC que facilitan el diseño, desarrollo, seguimiento, evaluación de proyectos por ejemplo educativos, productivos, comunitarios, etc. será el comienzo de una cadena de procesos que llevará a mejorar el nivel de vida en las comunidades generando un mejor nivel de desarrollo integral, entendido como el disfrute óptimo de las condiciones sociales, económicas, políticas, organizativas y de bienestar social que beneficiarán a las poblaciones campesinas e indígenas de nuestras instituciones y grupos regionales del país.

3.1.5. Planificación del desarrollo de la Biblioteca Digital Distribuida

Hoy en día existe una gran cantidad de conocimiento e información disponibles. El gran reto es poder brindar ese conocimiento de una manera rápida y eficiente para obtener la información relevante que se pueda aplicar en la solución de problemas. Las bibliotecas han sido tradicionalmente los repositorios de la sabiduría acumulada de la humanidad, desde los tiempos de la famosa biblioteca de Alejandría (año 48 aC), que contaba con más de 500,000 volúmenes (24 rollos de papiro equivalían a un volumen) hasta nuestros días en que contamos con bibliotecas. Actualmente, la mayoría de las bibliotecas cuentan con sistemas de cómputo para facilitar la búsqueda de información en un servicio conocido como catálogo electrónico. Algunas bibliotecas han tratado de ir más allá, y cuentan con una cantidad considerable de material en formato digital.

Una biblioteca digital es una colección organizada de documentos almacenados

en formato digital que a su vez ofrece los servicios de búsqueda y recuperación de información. Los documentos que se encuentran en una biblioteca digital pueden ser texto, imágenes, audio, video o combinaciones de los anteriores. Idealmente se deben de almacenar y poder recuperar documentos completos, y las búsquedas se realizan sobre el contenido completo de los documentos. Es decir, si el documento es texto, la búsqueda se realiza sobre el texto completo del documento y una vez localizado el documento deseado es posible obtenerlo de manera inmediata. Esto contrasta con los sistemas tradicionales que se concentran solamente en búsquedas basadas en el título, descripción o palabras clave.

Existen muchos retos tecnológicos para poder llegar a contar con una implantación completa de una herramienta que cumpla todas las características deseables de una biblioteca digital ya que la herramienta debe contemplar los siguientes aspectos:

- Captura de información
- Almacenamiento
- Clasificación e indexamiento
- Facilidades para buscar, filtrar y resumir grandes volúmenes de datos, imágenes e información
- Interfaces para la presentación de resultados
- Distribución del contenido de la biblioteca al usuario final
- Administración y control de acceso

La biblioteca digital no debe ser una entidad individual, se requieren medios tecnológicos para enlazar recursos, los usuarios deben poder acceder a los enlaces entre bibliotecas digitales y servicios de información de manera transparente. El acceso universal a las bibliotecas digitales y a los servicios de información debe ser un objetivo principal.

Sin embargo, no todo es fácil ni simple a la hora de pensar en la biblioteca digital.

Existen una serie de problemáticas que ponen freno a su rápida expansión, por lo que se debe tomar en cuenta al momento de realizar una planificación técnica, algunos factores como son:

Disponibilidad: Todo lo que existe registrado (impreso, fotografiado, filmado, pintado, dibujado, etc.) tendría que convertirse a formato digital para estar disponible a todos los usuarios con un terminal de trabajo.

Recuperación y adecuación: Cada usuario de este hipotético terminal de trabajo (que permitiría el acceso a la biblioteca digital) tendría que poder acceder a todos los documentos electrónicos relevantes de este universo digital, de una manera rápida y fácil.

Autenticidad: Cada usuario debería tener la seguridad de que el documento que encuentra en la red es el documento auténtico y original.

Utilización: Cada uno de los documentos recuperados mediante el terminal de trabajo tendría que ser recuperado de forma que todo usuario pudiera usarlo.

Protección de la propiedad intelectual: La protección de los derechos de autor debería estar garantizada en todo documento recuperado.

Asequibilidad: Los costes de acceso y recuperación de los diversos documentos tendrían que ser razonables y no superar los costes de sus equivalentes tradicionales.

Funcionalidad Mínima del Sistema

A continuación se lista la funcionalidad principal que tendrá el prototipo del sistema de biblioteca usando ROA:

- Registro de Usuarios de Biblioteca - ROA: Mediante esta funcionalidad,

los consumidores de información podrán registrarse como usuarios de la biblioteca digital distribuida. Para ello se modela en BPMN los flujos de información a ser validados por los Actores. El Registro de Usuario puede involucrar tanto procesos humanos como procesos automatizados.

Seguidamente se lista la funcionalidad usando SOA:

- Registro de Recursos Educativos - SOA: Los proveedores de información deberán ser capaces de registrar los contenidos educativos que posean. Tal registro puede involucrar sólo los datos básicos para su ubicación (por ejemplo: título, autor, código).
- Búsqueda de Recursos Educativos - SOA: Los consumidores de información deberán ser capaces de realizar búsquedas básicas y avanzadas sobre los recursos de su interés. La búsqueda puede involucrar consultas distribuidas hacia otros nodos del prototipo del sistema de biblioteca.
- Agregar Retroalimentación - SOA: Mediante esta funcionalidad, los consumidores de información serán capaces de dejar comentarios y evaluar los contenidos que hayan revisado. Por cada recurso, el sistema deberá proporcionar un criterio de evaluación promedio que represente en forma ponderada el peso de todos los factores de evaluación.

Otros Requerimientos del Sistema

- **Uso de Software Libre:** Las limitaciones económicas del sector educativo en nuestro país motivaron a desarrollar el prototipo del sistema de biblioteca usando herramientas de software libre; siendo el sistema a desarrollar también software libre. Con ello será posible poner a disposición de la comunidad el código fuente del sistema para su mejora continua y el acceso irrestricto al mismo por parte de múltiples instituciones educativas sin la barrera inicial del licenciamiento comercial.

- **Entorno de Ejecución:** Se necesita un entorno de ejecución escalable, multiplataforma y que dé soporte a Web Services. La escalabilidad es necesaria porque en el largo plazo se prevé que habrá varios nodos del prototipo de biblioteca que compartirán información; por lo que será necesario atender a demandas crecientes con facilidad. Se prevé además que sistemas operativos como Linux dominarán el mercado de PC's en el futuro, por lo que el sistema deberá estar preparado para ser fácilmente adaptado a nuevos sistema operativos. El entorno además deberá proveer capacidades de integración vía Web Services, pues ésta es la tecnología usada para interconectar los nodos del prototipo de la biblioteca digital distribuida.

Para atender los requerimientos de escalabilidad y multiplataforma, se decidió utilizar un Servidor de Procesos de código abierto basado en BPEL en lugar de un Servidor de Aplicación basados en J2EE, esto permite desarrollar la aplicación sin preocuparnos de los detalles de códigos en Java, lo cual permite escalabilidad (el que es transparente durante el desarrollo), y dado que está basado en BPMN, la aplicación puede fácilmente ser ejecutada en variados sistemas operativos, a través de los servidores de procesos que ejecutan el BPEL generado a partir del modelo BPMN.

En este proyecto en particular se tuvo que crear una arquitectura propia denominada ROA con la cual se aseguró el éxito del proyecto prototipo denominado Dexter's Lib; cuyo nombre se debe a su relación con las caricaturas animadas en el ámbito educativo [ver referencia 27]. La nueva Arquitectura Orientada a Regulaciones ROA permite adaptar metodologías existentes basadas en el contexto de automatización de procesos en un proyecto dado. En la práctica para Dexter's Lib se construyó una versión resumida de la metodología PMBOOK propuesto por el Instituto de Gestión de Proyectos PMI para gestionar el avance del proyecto como un todo; así como la metodología

GTZ como pilar de reducción administrativa de procesos para descubrir y optimizar el flujo de información entre los actores de una organización determinada. Asimismo el uso de la metodología de Proceso Unificado Racional RUP con una variante para adaptar la Notación de Modelado de Procesos de Negocios BPMN en lugar del Lenguaje de Modelado Unificado UML, que nos permitió definir los roles en el equipo de desarrollo de software.

Gestión de Riesgos para un Proyecto BPM						
N°	Riesgo Identificado	Respuesta Potencial	Causa Principal	Sub Categoría	Categoría	Prioridad
1	Procedimientos inconclusos de instalación de la plataforma BPM	Esbozar un cronograma de capacitación.	Capacitación inapropiada	Calidad	Riesgo Técnico	Alta
2	Procedimientos inconclusos de escenarios de integración BPM	Desarrollar prototipos de integración.	Poca información de integración Intalio-Alfresco-Liferay-LDAP-Email.	Calidad	Riesgo Técnico	Alta
3	Tecnologías nuevas en la Ingeniería de Procesos	Reforzar capacitación en XSLT, BPEL y Patrones de Procesos	Se requiere conocer BPMN, BPEL, XML, XSLT, SOAP, y SOA.	Herramientas	Riesgo Técnico	Media
4	Tecnologías nuevas en la Ingeniería de Software	Reforzar capacitación en Ajax, Javascript, Java, y SQL	Se requiere conocer J2EE, Spring, Hibernate, Ajax, Struts, SQL Avanzado.	Herramientas	Riesgo Técnico	Media
5	Uso de diversas tecnologías y diversos fabricantes en la solución	Desarrollar DEMOS-Pilotos para validar la arquitectura a seguir	Debido a la adopción de utilizar Open Source	Compatibilidad	Riesgo Técnico	Media
6	Temperatura gélida en el Centro de Datos	Configurar XServer para acceso remoto	Instalación Hardware en el Centro de Datos	Problemas Ambientales	Riesgo Externo	Baja
7	Poca presencia de Especialistas BPM en el mercado nacional	Capacitación con Especialistas Nacionales y/o Extranjeros	SOA y ROA son arquitecturas emergentes.	Temas del Proveedor	Riesgo Externo	Media
8	Orientación a soluciones de código abierto Open Source	Orientar el ahorro de licencia hacia capacitaciones especializadas	Reducir costos de licencias, operativos y de mantenimiento.	Nuevas Regulaciones del Gobierno	Riesgo Externo	Media
9	Carencia de un proyecto similar como referente	Tomar contacto con instituciones con soluciones equivalentes	En el mercado existen pocos proyectos similares		Riesgo Externo	Media
10	Procesos no acorde con la realidad	Particionar a futuro los proyectos en 2 etapas: 1-Modelado del Proceso y 2-Automatización del Proceso Modelado	El compromiso de cumplir con plazos no validados con un Especialista en el tema	Alcance del Proyecto	Riesgo Organizacional	Alta
11	Postergación de fechas de entrega	Contar con un Consultor guía	Desconocimiento de detalles tecnológicos	Retraso del Personal	Riesgo Organizacional	Alta
12	La prioridad está en función de la gestión de turno	Integrar a todas las áreas involucradas	Carencia de políticas institucionales a largo plazo	Cambios en Prioridad del Proyecto	Riesgo Organizacional	Baja
13	Poco presupuesto para la Arquitectura Orientada a Servicios	Organizar charlas informativas	No existe una concepción de arquitectura SOA institucional	Reducción de Presupuesto	Riesgo Organizacional	Media
14	Procedimientos inconclusos de mantenimiento de la plataforma BPM	Involucrar al área de operaciones en la validación de entregables	Desconocimiento de requisitos del área de Operaciones	Estimación del Proyecto	Riesgo Organizacional	Alta
15	Existen requerimientos de arquitectura no especificados	Asistencia a conferencias de los fabricantes involucrados en el tema	Poca experiencia en BPM sobre ROA	Requerimientos	Riesgo Organizacional	Media

Cuadro 3.2: Matriz de riesgos ROA - BPM

En el cuadro 3.2 se presenta el detalle de gestión de riesgos de un proyecto ROA basado en PMBOOK; en el cual se muestran los riesgos detectados, así como una matriz de probabilidades e impacto esperado.

A nivel de marco regulatorio, la metodología GTZ cumple con los estándares metodológicos de reducción administrativa basado en las leyes peruanas: Ley

Nº27444 (Ley de Procedimiento Administrativo General); 29158 (Ley del Poder Ejecutivo); 29060 (Ley del Silencio Administrativo); 29091 (publicación en portal), ISO/TC 176/SC 2/N 544R2, etc.

Cuando se emplea ROA-BPM por primera vez surgen actividades de alto riesgo debido a temas relacionados con el dominio de las técnicas metodológicas y tecnológicas para el proyecto.

Matriz de Probabilidades versus Impacto						
Probabilidad	0.9	0.09	0.27	0.45	0.63	0.81
	0.7	0.07	0.21	0.35	0.49	0.63
	0.5	0.05	0.15	0.25	0.35	0.45
	0.3	0.03	0.09	0.15	0.21	0.27
	0.1	0.01	0.03	0.05	0.07	0.09
		0.1	0.3	0.5	0.7	0.9
		Impacto				

Criterio de Prioridades:

Bajo	
Medio	
Alto	

Cuadro 3.3: Matriz de riesgos versus impacto ROA - BPM

GTZ es una metodología que permite descubrir y documentar los procedimientos que existen en los procesos, de ahí la importancia de esta técnica. GTZ utiliza como herramientas la tabla ASME, análisis de costos, diagrama de bloque, análisis de estructura orgánica, entre otras. Sin embargo, el hecho de utilizar un enfoque basado en procedimientos, de algún modo dificulta la lectura del proceso en forma global cuando existen ramificaciones y bifurcaciones. Más aún cuando se presentan eventos múltiples concurrentes, en esos escenarios, es preferible utilizar la notación dinámica BPMN para mapear estos flujos que por sí serían difíciles de mostrar en un diagrama de bloque convencional.

En la Figura 3.5 se muestra la aplicación de GTZ en un contexto de arquitectura ROA:

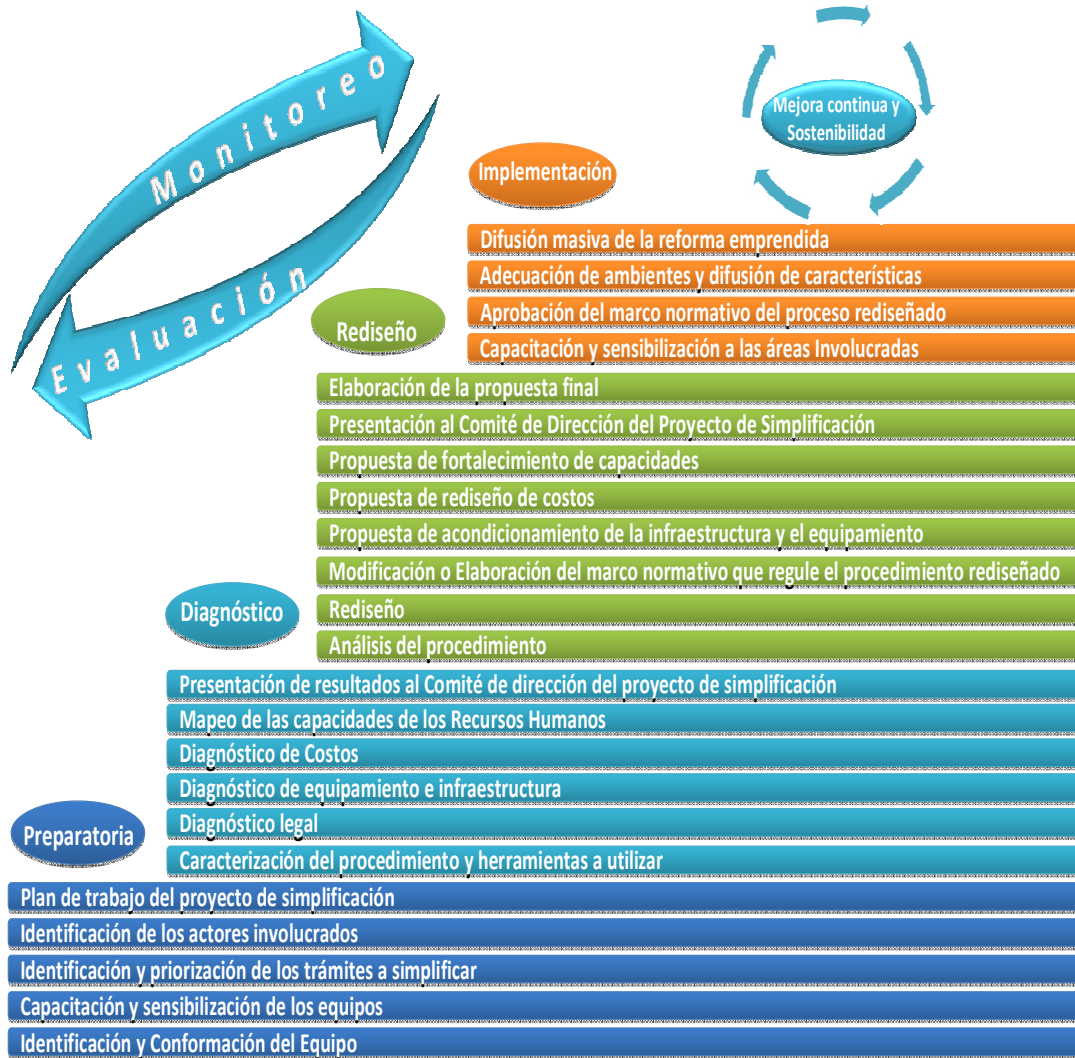


Figura 3.5: Etapas de la simplificación administrativa basado en la Metodología GTZ

Para solucionar esta deficiencia GTZ, surge el enfoque ROA-GTZ, que nos permite integrar la técnica de BPMN como parte de esta metodología. El valor agregado de esta técnica es que rápidamente se detectan los procesos, los puntos de decisión, actores, roles y áreas involucradas, es decir, los tiempos de análisis se reducen debido a la ventaja de visualización gráfica de los macro procesos y procesos a detalle.

Para adaptar GTZ a ROA es importante establecer las fronteras de los aportes metodológicos de cada metodología, al final el límite de la frontera es el macro modelo BPMN para luego pasar a código ejecutable BPEL.

Antes de realizar el macro modelado BPMN de los procesos de una organización, es necesario tener en claro los conceptos que rigen un proceso en particular. A nivel de entidades públicas: GTZ recomienda la siguiente clasificación de procesos (1):

- *Procesos estratégicos o para la gestión de una organización. Incluyen procesos relativos a la planificación estratégica, establecimiento de políticas, fijación de objetivos, provisión de comunicación, aseguramiento de la disponibilidad de recursos necesarios y revisiones por la dirección.*
- *Procesos operativos o de realización. “Son los procesos relacionados con las funciones sustantivas de la entidad, es decir aquellas ejercidas directamente sobre su ámbito de competencia e indispensables para el cumplimiento de su finalidad resultando en la prestación de servicios a la población o la regulación o promoción de determinada actividad social o económica”, también conocidos como procesos esenciales (1).*
- *Procesos de apoyo o para la gestión de recursos. Incluyen todos aquellos procesos para la provisión de los recursos que son necesarios en los procesos estratégicos, operativos y de medición de la entidad.*
- *Procesos de medición, análisis y mejora. Incluyen aquellos procesos necesarios para medir y recopilar datos para realizar el análisis del desempeño y la mejora de la eficacia y la eficiencia. Incluyen procesos de medición, seguimiento y auditoría, acciones correctivas y preventivas, y son una parte integral de los procesos estratégicos, operativos y de apoyo.*

En ese sentido, para definir los procesos esenciales es importante tener claro los siguientes elementos de la caracterización de un proceso:

¹ Fuente: Guía Metodológica complementaria para el análisis de las matrices de delimitación de competencias y distribución de funciones sectoriales

- Entrada/Insumos/Requisitos: Qué insumos se han usado para iniciar el proceso.
- Salida/Resultados:Cuál es el producto o servicio creado por el proceso.
- Cliente: Para quién hacemos el proceso.
- Objetivo: Para qué hacemos el proceso.
- Propietario: Quién es el responsable del proceso.

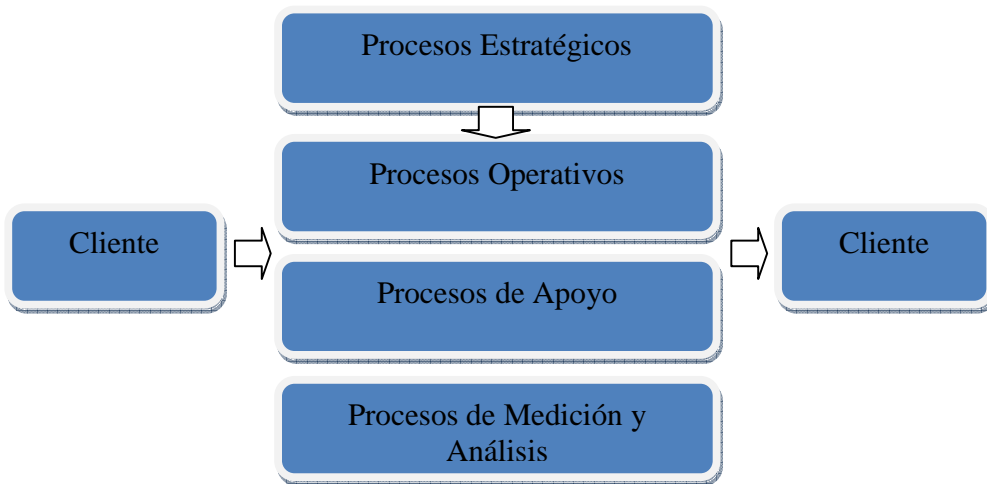


Figura 3.6: Tipos de procesos y su interacción con el cliente

Antes de iniciar el modelado de un proceso, en concreto, se debe identificar los elementos de un proceso dado; en el siguiente diagrama se muestra el detalle:

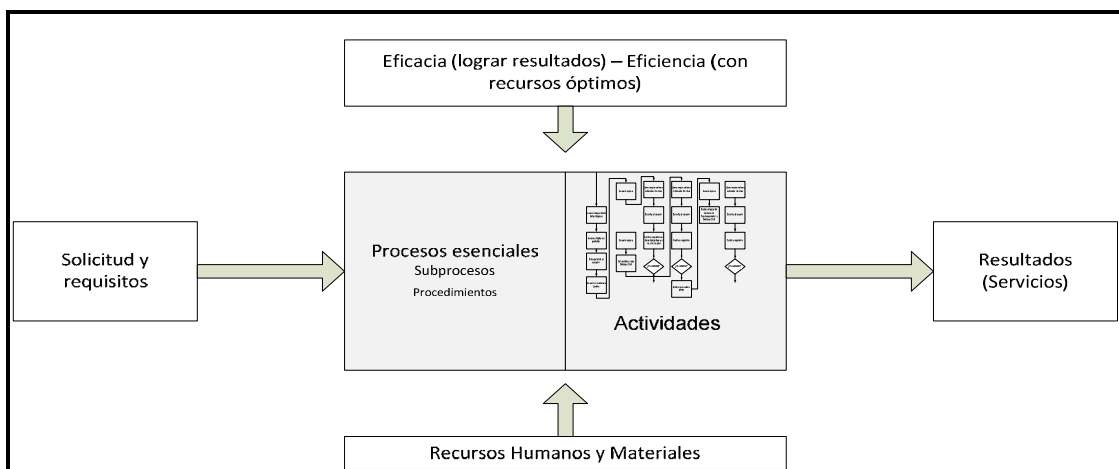


Figura 3.7: Interacción interna entre los elementos de un proceso

El siguiente paso es alinear los procedimientos de los procesos con los macro procesos institucionales; en aquellos procesos que no están considerados como prioritarios existen dos posibilidades: eliminar el proceso o modificarlo para que se encaje con el plan estratégico.

De los Objetivos Estratégicos a los Resultados

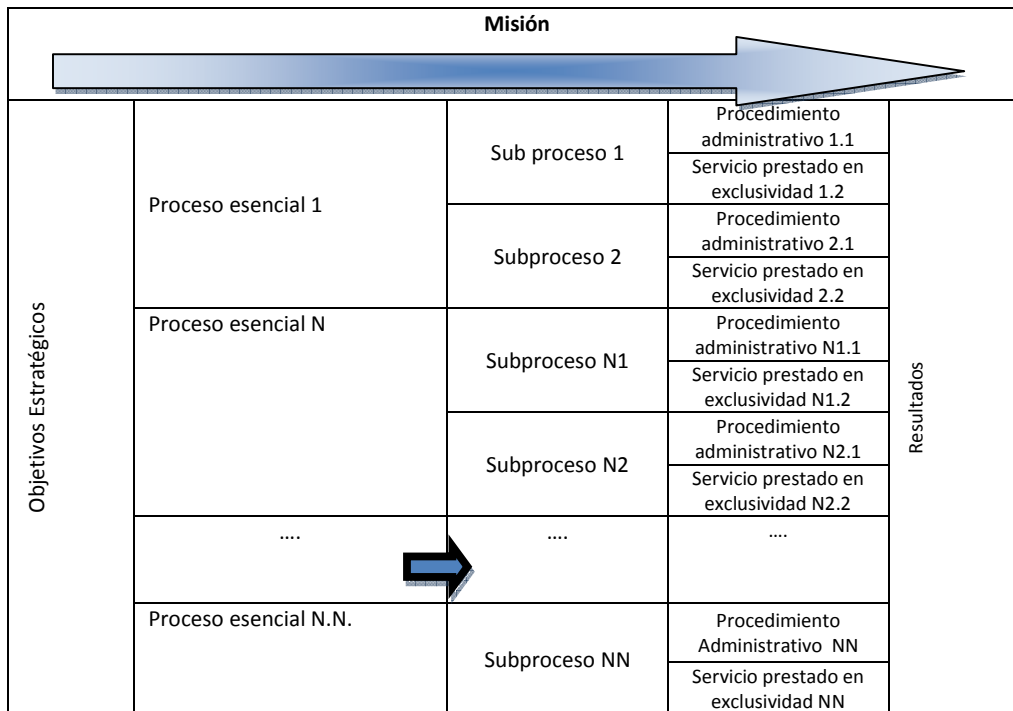


Figura 3.8: Nivel de resolución de los objetivos estratégicos a los resultados

Para identificar aquellos procesos que están alineados con las prioridades de la institución, se suele utilizar una matriz de priorización como la que se describe a continuación:


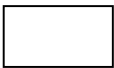
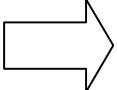


Criterio de priorización	Procedimiento 1	Procedimiento 2	Procedimiento N.
Alineamiento a misión y objetivos de la entidad, es un proceso esencial	5	3	1
Mayor demanda del trámite	3	5	1
Criterio N	5	3	5
Total	13	11	7

Cuadro 3.4: Matriz de Priorización ROA - GTZ

GRUPO DE ACTORES	ACTOR	ROL EN EL PROYECTO	RELACION PREDOMINANTE	JERARQUIZACIÓN DE SU PODER
Clasificación de los diferentes actores	Conjunto de personas con intereses homogéneos que participan en un proyecto NOMBRE	Funciones que desempeña cada actor y el objetivo que persigue con su accionar	Se define como las relaciones de afinidad (confianza) frente a los opuestos (conflicto) 1. A FAVOR 2. INDIFERENTE 3. EN CONTRA	Capacidad del actor de limitar o facilitar las acciones 1. ALTO 2. MEDIO 3. BAJO

Cuadro 3.5: Matriz de Actores ROA - GTZ

Como siguiente paso luego de identificar los actores y procesos basados en las líneas prioritarias de la organización, se procede a construir la tabla ASME que será modificado para adaptarlo a la arquitectura ROA, para lo cual se utiliza la siguiente nomenclatura:

Uso del símbolo	Símbolo
Operación: identifica las actividades más relevantes del procedimiento. Ejemplo: orientar a un administrado, consultar una base de datos o firmar un documento.	
Revisión: identifica actividades que son dedicadas a verificar la "calidad" de algo. Ejemplo: verificar que los requisitos sean completos y correctos.	
Traslado: identifica actividades en la que documentos o personas deben trasladarse. Es útil diferenciarlas, pues permite analizar, si este traslado es realmente inevitable.	
Espera: identifica actividades que no revisten acción y que detienen temporalmente el flujo del procedimiento. Ayuda a detectar cuellos de botella, pues las actividades de "espera" debieran ser idealmente pocas o de corta duración.	
Archivo: identifica acciones orientada a archivar documentos o expedientes; el detectar que se archivan demasiados documentos o expedientes, sugiere analizar si realmente es necesario solicitarlos.	

Cuadro 3.6: Símbolos ASME usados en la adecuación de la metodología ROA-GTZ para identificar los tipos de actividad

Como se puede apreciar ASME tiene limitaciones intrínsecas, tales como su complejidad al momento de manejar flujos alternativos, también posee el limitante de nivel de resolución. Es decir, se podría tener un ASME para Macro

Procesos y otro ASME para el proceso con detalles administrativos con su respectiva contraparte de procesos de línea de negocio. Justamente para solucionar este inconveniente ROA-GTZ propone utilizar BPMN para mapear los diversos niveles de resolución.

Valor	Descripción
VA = valor añadido	Cuando su realización contribuye al resultado final del procedimiento. Por ejemplo, el firmar un expediente, confirmar que los documentos requeridos están completos, orientar a un administrado respecto de los pasos a seguir para su trámite, genera valor añadido, pues está dando información necesaria al interesado.
Control = si es una actividad de control	Actividad que permite garantizar la calidad del procedimiento. Por ejemplo, revisar un expediente para garantizar que está completo antes que se produzca la siguiente actividad. Se debe estar atento, pues en general, en muchas entidades, se detectan actividades de control repetidas.
SVA = sin valor añadido	Actividad que se realiza y no genera valor al procedimiento. Por ejemplo el poner el VºBº sobre un expediente, que ya tiene un VºBº de otro funcionario autorizado, es una actividad SVA (sin valor añadido).

Cuadro 3.7: Calificación del valor de actividad ROA - GTZ

Es decir, ROA-GTZ es una variante de la metodología GTZ que utiliza la técnica de las tablas ASME para descubrir y mapear los flujos de actividades de los procesos entre los actores de un flujo de negocio dado. A diferencia de GTZ, ROA-GTZ utiliza BPMN como notación en un segundo nivel de abstracción como complemento, pero no en reemplazo al diagrama de bloques.

En la Figura 3.9 se muestra la aplicación de esta técnica para un Proceso de Apertura de Trámite de Solicitud de Latencia para un proceso:

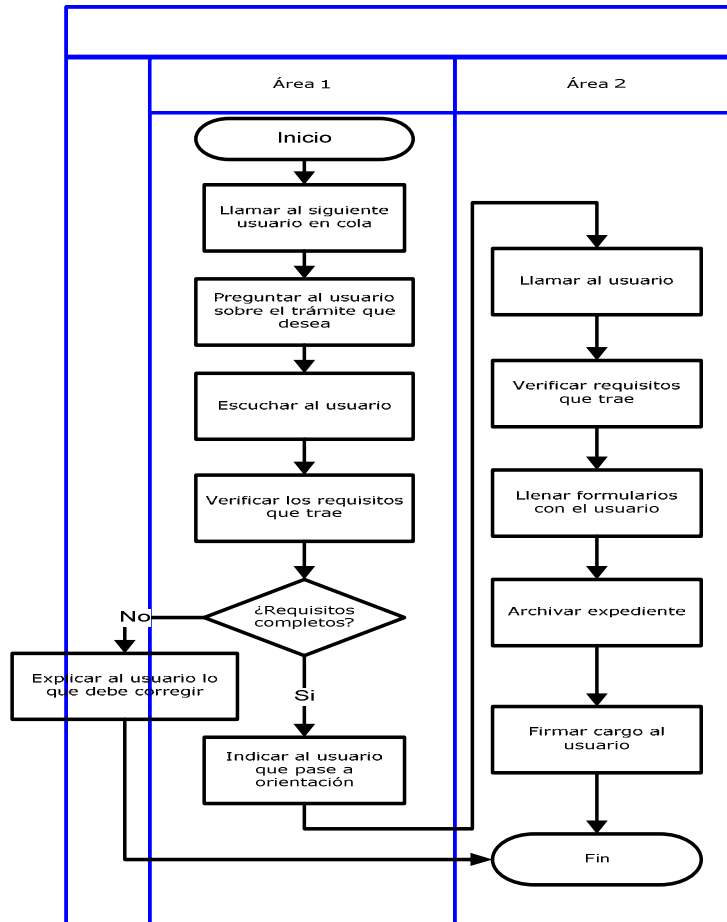


Figura 3.9: Diagrama de bloques utilizado en la Metodología GTZ

Finalmente se procede a mapear el diagrama de bloques con la herramientas informática BPM (*Business Process Management: Gestión de procesos de Negocio*), que permiten dibujar de manera dinámica los diagramas de bloques, y el mismo software, identifica cuellos de botella, pasos reiterados e incluso llega a generar el código del programa que lleva el procedimiento a nivel de automatización.

Es importante recalcar que para una buena visibilidad en el avance de un proyecto, se recomienda dividir el proceso en sub procesos tal como se muestra en el Cuadro 3.8. La ventaja de ROA es que permite utilizar lo mejor de GTZ con

PMBOOK para que sirva como entrada al momento de implementarlo en una plataforma de un Sistema de Gestión de Procesos de Negocios.



Proyecto Prototipo - Defensoría del Asegurado - Versión 1.2 (Escenario de Flujo Normal)		Defensoría del Asegurado - Etapa I										
		Fase I: Procesar Dato del Que se Queja				Fase II: Crear Registro de Queja		Fase III: Atiende Queja del Usuario			Fase IV: Procesar Cierre de Queja	
		F1.1.FN Buscar Datos del Asegurado	F1.2.FN Mostrar Ficha de Queja con los Datos del Asegurado	F1.1.FA1 Registrar Datos de Usuario	F1.2.FA1 Mostrar Ficha de Queja con los Datos del Usuario	F2.1 Crea y Registra Queja	F2.2 Notifica Registro de Queja	F3.1 Revisión del Coordinador	F3.2 Revisión del Director	F3.3 Revisión del Jefe de Servicio	F4 Enviar Respuesta de Queja	F1.2.FA1 Mostrar Ficha de Queja con los Datos del Usuario
Presentación	Esquemas	100%	100%	NA	NA	100%	0%	100%	0%	0%	0%	0%
	Transformadores	0%	0%	NA	NA	0%	0%	0%	0%	0%	0%	0%
	Javascript	NA	NA	NA	NA	NA	NA	NA	NA	NA	0%	0%
	Formularios	100%	100%	NA	NA	100%	0%	100%	0%	0%	0%	0%
Dominio	Esquemas	100%	100%	NA	NA	100%	0%	100%	0%	0%	0%	0%
	Transformadores	0%	0%	NA	NA	0%	0%	0%	0%	0%	0%	0%
	BPM - Vertical	0%	0%	NA	NA	0%	0%	0%	0%	0%	0%	0%
	Integración	0%	0%	NA	NA	0%	0%	0%	0%	0%	0%	0%
Servicio	Esquemas	100%	100%	NA	NA	100%	0%	0%	0%	0%	0%	0%
	Transformadores	NA	NA	NA	NA	NA	NA	0%	0%	0%	0%	0%
	Plantillas	100%	100%	NA	NA	100%	0%	0%	0%	0%	0%	0%
	WS-Servicios	0%	0%	NA	NA	0%	0%	0%	0%	0%	0%	0%
Persistencia	SQL-(Toad)	90%	90%	NA	NA	90%	0%	0%	0%	0%	0%	0%
	Pro. Almacenado	90%	90%	NA	NA	90%	0%	0%	0%	0%	0%	0%
	WS-Persistencia	NA	NA	NA	NA	NA	NA	0%	0%	0%	0%	0%

Cuadro 3.8: Matriz de avance de un proyecto ROA – BPM utilizando GTZ y PMBOOK

Un aspecto clave en todo proyecto ROA es el manejo de tiempos basado en los avances; para ello se propone una matriz de avance que se detalla en el Cuadro 3.8, en el cual se parte un proceso en fases y éstos en sub procesos o actividades basado en cuatro perspectivas de capas de abstracción, esto es: Capa de Presentación, Capa de Dominio, Capa de Servicio y Capa de Persistencia. La ventaja de esta separación es que nos permite sincerar los tiempos que se van a utilizar en un proyecto determinado que utiliza ROA-BPM.

Con respecto al análisis de costos utilizado en GTZ, se incorpora una mejora al fusionar la técnica de Análisis de Monte Carlo para analizar posibles soluciones o mejoras en el flujo del proceso actual.

#	Funcionario/ empleado(a)	Componente Intrapersonal				Componente Interpersonal			Capacidad de adaptarse al cambio			Manejo del estrés		Comen- tarios
		Com- prensi ón de sí mismo	Aserti- vi- dad	Autorr- eali- zación	Indepe- n- dencia	Empat- ia	Relacione- s interper- sonales	Respon- sabi- lidad social	Solució- n de problem- as	Prueb- a de realid- ad	Flexibi- li- dad	Toleran- cia al estrés	Contro- l de impuls- os	
1	Funcionario 1													
2	Funcionario 2													
3	Funcionario 3													
5	Empleado 1													
6	Empleado 2													
...	Empleado N													

 Áreas de mejora²
 Fortalezas

Cuadro 3.9: Matriz de habilidades ROA - GTZ

Existen algunas metodologías tales como 6 Sigma, Calidad total, MPT que se aplican a la mejora continua de los procesos, en todas estas metodologías se puede adaptar ROA en dichos escenarios como un paso adicional para realizar la reingeniería del proceso.

En el cuadro 3.10 se muestra la nueva tabla ASME-VM, luego del rediseño. Una mejora ROA-GTZ es que en forma directa se mapea al diagrama BPM, lo cual nos permite tener una visión gráfica general y unificada del proceso que se analiza. En escenarios complejos donde se requiere justificar técnicamente la implementación de una mejora propuesta, se suele incorporar un análisis de Monte Carlo para la toma de decisión más consistente frente a los escenarios de flujos alternativos propuestos.

² Cabe aclarar que los casilleros marcados en gris no descalifican la labor que los funcionarios vienen desempeñando, sino que únicamente indican áreas de mejora que pueden ser abordadas en futuros programas de capacitación.

Paso	Actividad	Área	Tiempo	Contador de Recursos							Tipo de actividad					Tipo de valor			
				Recursos Humanos					Otros recursos		Operación	Revisión	Traspaso	Espera	Archivo	VA	Control	SVA	
				R1	R2	R3	R4	R5	R6	R7	○	□	⇨	D	▽				
1	Recepcionar solicitud	Mesa de Partes	2	0	0	1	0	0	1	0	●						1		
2	Verificar conformidad	Mesa de Partes	3	0	0	1	0	0	1	0	●							1	
3	Preparar Hoja de Trámite	Mesa de Partes	5	0	0	1	0	0	1	0	●						1		
4	Ingresar los datos de Hoja de Trámite	Mesa de Partes	2	0	0	1	0	0	0	1	●						1		
5	Llamar al conserje	Mesa de Partes	1	0	1	0	0	1	0	0	●						1		
6	Trasladar al técnico destinatario	Conserje	10	0	0	0	0	1	0	0	●						1		
7	Recepcionar el expediente	Oficina destinataria	1	0	0	0	1	0	1	0	●						1		
8	Registrar en cuaderno de cargo	Oficina destinataria	1	0	0	0	1	0	1	0	●							1	
9	Trasladar al despacho del técnico destinatario	Oficina destinataria	1	0	0	0	1	0	0	0	●						1		
10	Tomar conocimiento	Oficina destinataria	60	0	0	1	0	0	0	0	●						1		
11	Preparar informe sobre costo de reproducción de información	Oficina destinataria	15	0	0	1	0	0	1	0	●						1		
12	Ordenar a secretaria que lo lleve a Mesa de Partes	Secretaría General	1	0	0	1	0	0	0	0	●						1		
13	Trasladar informe a Mesa de Partes	Secretaría Técnico	5	0	0	0	1	0	0	0	●						1		
14	Firma cuaderno de cargos	Mesa de Partes	0,5	0	0	1	0	0	1	0	●							1	
15	Entregar informe al interesado	Mesa de Partes	3	0	0	1	0	0	1	0	●						1		
16	Firmar cuaderno de cargos	Interesado	0,5	0	0	0	0	0	1	0	●							1	
			111	0	1	9	4	2	9	1							12	4	0

Cuadro 3.10: Matriz ASME del proceso con procedimiento rediseñado con ROA - GTZ

En la figura 3.10, se muestra un diagrama de bloques del proceso luego del rediseño. En nuestro caso al momento de plantear el proceso de biblioteca digital distribuida no va a existir un rediseño ya que estamos estructurando desde el inicio la secuencia del flujo del proceso. Sin embargo en situaciones reales en donde ya existen procesos, surge la necesidad de plasmar dichos procesos en forma de diagrama de flujo, de modo que permita detectar los cuellos de botella y proponer mejoras en cuanto a la secuencia de los flujos de información. Una buena práctica consiste en agregar indicadores de costos por cada actividad de un proceso, para cuantificar en números los costos operativos de un proceso en particular.

De ahí la importancia de utilizar ROA-BPM, ya que permite simular un escenario global de un proceso incluyendo a todos los actores, que permita: validar el flujo

así como las mejoras propuestas en un corto plazo, para su posterior implementación.

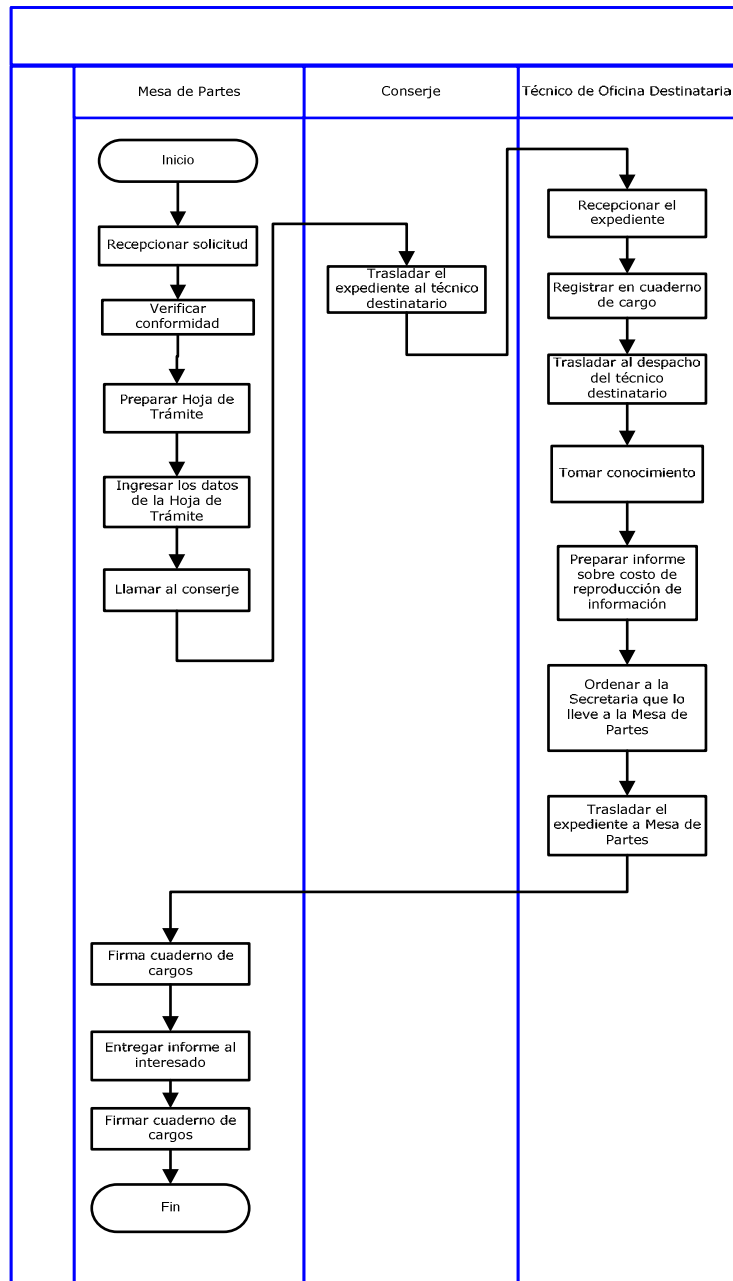


Figura 3.10: Diagrama de actividades del rediseño del proceso

Para el ejemplo del diagrama de bloques, se ve del Cuadro 3.11, que la propuesta de nuevo procedimiento es bastante ventajosa, respecto al

procedimiento actual, haciéndolo más eficiente, más rápido, y permite ahorrar gran cantidad de recursos. Esto es una de las ventajas de utilizar los indicadores de comparación del procedimiento actual versus el procedimiento propuesto.

	Actual	Propuesto	Diferencia
Actividades total	66	16	-50
• Con valor añadido	17	12	-5
• De control	7	4	-3
• Sin valor añadido	42	0	-42
Tiempo	244,5 minutos	111 minutos	-133,5
Cantidad de áreas participantes	5	3	-2
Recursos			
R1	0	0	0
R2	7	1	-6
R3	22	9	-13
R4	26	4	-22
R5	13	2	-11
R6	34	9	-25
R7	1	1	0

Cuadro 3.11: Comparación de indicadores de desempeño entre el procedimiento actual y el procedimiento propuesto con ROA -GTZ

La importancia de ROA es que permite adecuar la metodología GTZ enmarcado en el cuadro ASME con su respectivo diagrama de bloque como paso previo al rediseño del proceso y su posterior pase a la notación BPMN para la automatización del proceso de negocio.

3.2. Diseño de la Biblioteca Digital Distribuida

La principal ventaja de los sistemas distribuidos es la capacidad de compartir y acceder a la información de una forma fiable y eficaz, así como la utilización compartida de los datos y distribución del control.

En la figura que se muestra a continuación se aprecia el flujo de información relacionado con el proceso de atención de una biblioteca digital distribuida. La importancia de la arquitectura ROA basado en BPM es que se puede intercomunicar diversos nodos a través de Web Services vía conectores WSDL.

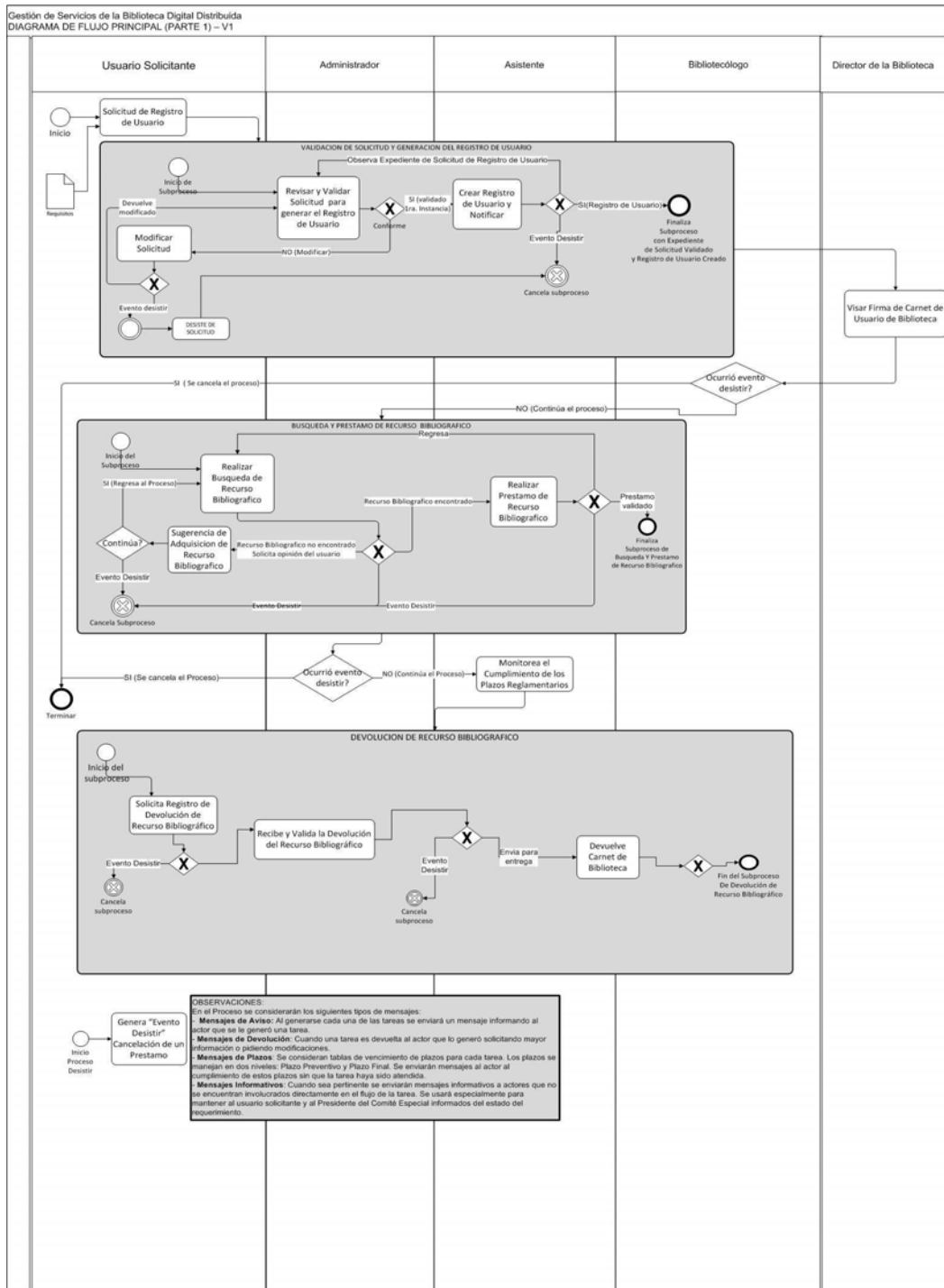


Figura 3.11: Diagrama de bloques utilizado en el modelo ROA-BPM a partir de la metodología GTZ

La ventaja principal de compartir los datos por medio de la distribución es que cada localidad pueda controlar hasta cierto punto los datos almacenados localmente.

En un sistema centralizado, el administrador de base de datos de la localidad central controla la base de datos. En un sistema distribuido existe un administrador global de la base de datos que se encarga de todo el sistema. Parte de esta responsabilidad se delega al administrador de base de datos de cada localidad.

Dependiendo del diseño del sistema distribuido, cada administrador local podrá tener un grado de autonomía diferente, que se conoce como autonomía local. La posibilidad de contar con autonomía local es en muchos casos una ventaja importante de las bases de datos distribuidas.

Si se produce un fallo en una localidad de un sistema distribuido, es posible que las demás localidades (o nodos) puedan seguir trabajando. En particular, si los datos se repiten en varias localidades, una transacción que requiere un dato específico, puede encontrarlo en más de una localidad. Así, el fallo de una localidad no implica necesariamente la desactivación del sistema.

El sistema debe detectar cuando falla una localidad y tomar las medidas necesarias para recuperarse del fallo. El sistema no debe seguir utilizando la localidad que falló. Por último, cuando se recupere o repare esta localidad, debe contarse con mecanismos para reintegrarla al sistema con el mínimo de complicaciones.

La disponibilidad es fundamental para los sistemas de bases de datos que se utilizan en aplicaciones de tiempo real. Si una consulta comprende datos de varias localidades, puede ser posible dividir la consulta en varias subconsultas que se ejecuten en paralelo en distintas localidades.

Sin embargo, en un sistema distribuido no se comparte la memoria principal, así que no todas las estrategias de intersección se pueden aplicar en estos sistemas. En los casos en que hay repetición de los datos, el sistema puede pasar la consulta a las localidades más ligeras de carga.

En contrapartida, la desventaja principal de los sistemas distribuidos es la mayor complejidad que se requiere para garantizar una coordinación adecuada entre localidades.

El aumento de la complejidad se refleja en:

- Costo del desarrollo de software: es más difícil estructurar un sistema de bases de datos distribuidos y por tanto su costo es mayor.
- Mayor posibilidad de errores: puesto que las localidades del sistema distribuido operan en paralelo, es más difícil garantizar que los algoritmos sean correctos.
- Mayor tiempo extra de procesamiento: el intercambio de mensajes y los cálculos adicionales son una forma de tiempo extra que no existe en los sistemas centralizados.

En el presente proyecto, la arquitectura del sistema está basada en la Plataforma BPM de Intalio que se despliega sobre TOMCAT ó JBOSS sobre tecnología J2EE la cual está conformado por un conjunto de estándares para el desarrollo de componentes y para la creación de una plataforma de ejecución basado en lineamientos para el desarrollo de aplicaciones distribuidas, dando información sobre mejores prácticas para el desarrollo de un sistema robusto.

Una implementación de referencia de la plataforma BPM, posee una suite para modelar el proceso, simularlo y desarrollar su ejecutable basado en componentes BPEL. Esto permite modelar objetos y luego enlazarlos a una base de datos convencional, dichos objetos almacenan la información requerida en la

lógica de negocios, con la ventaja de que pueden ser instanciados local o remotamente a través de Web Services vía descriptores WSDL.

Para entender mejor la distribución del empaquetado en las clases Java, en las Figuras 3.3 y 3.4 se aprecian la separación a nivel de componentes, en donde la capa de Integración es la que contiene al componente MOM, encargado de la comunicación entre nodos. Asimismo, la capa de presentación contiene los JSP, y las clases del framework Struts encargadas de las interfaces con el usuario final. En lo que respecta a la capa de Negocio, están contenidas en los Session Beans quienes procesan la funcionalidad interna de los casos de usos implementados.

Dentro de los beneficios que obtenemos con esto podemos mencionar:

- Se evita desarrollar servicios de bajo nivel (que den soporte a componentes de negocio).
- Se cuenta con una arquitectura estable y bien aceptada para elaborar un sistema distribuido.
- Basada en estándares.

3.2.1. Especificaciones de Diseño

Inicialmente el análisis y diseño del proyecto estuvo enfocado sobre un Servidor de Aplicación basado en J2EE utilizando SOA-RUP con UML, posteriormente se migró la solución a BPEL utilizando ROA-BPM con BPMN. Es así que en la Figura 3.13 muestra el esquema de comunicación SOA de la Biblioteca Digital Distribuida. El proceso se inicia cuando un usuario ingresa al sistema en busca de información. Para ello procede a llenar un formulario con los datos del contenido que requiere (Paso 1). Esta información viaja hacia el servidor de aplicaciones revisando si el contenido se halla en la base de datos local (Paso 2). El proceso también puede involucrar realizar búsquedas distribuidas en otras bibliotecas tanto en la misma localidad como a nivel nacional o internacional (esto a disposición del usuario).

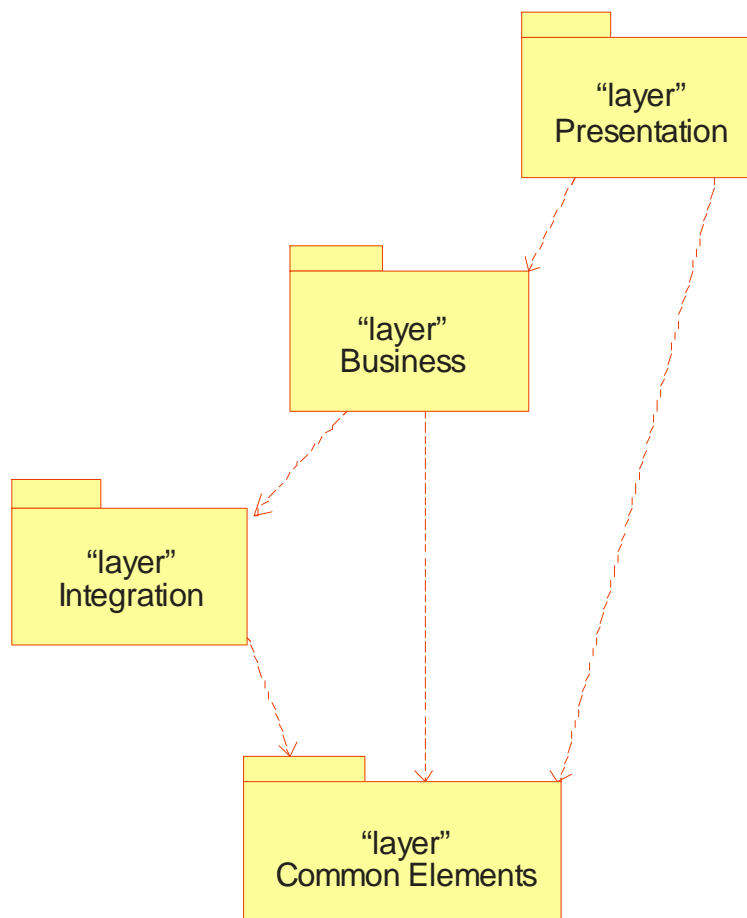


Figura 3.12: Diagrama de componentes SOA-J2EE

Para ello es necesario conocer qué otros sistemas están interconectados a la red. Es necesario entonces contar con un registro de las bibliotecas, el cual es mantenido mediante una base de datos UDDI (Universal Description, Discovery and Intergration). Si no se había realizado anteriormente, se procede a ubicar los nodos activos en la base de datos UDDI (Paso 1). Un mensaje de búsqueda distribuida es enviado hacia el servidor de mensajes local (MOM)(Paso 3), el cual replica el mensaje hacia varios nodos de manera simultánea (Paso 4), siendo recibido por los servidores de mensajes de los nodos suscritos (en la figura: Biblioteca Nacional, un Colegio en Surco, y un Colegio en el Rímac). El

mensaje es transmitido hacia sus servidores de aplicaciones respectivos (Pasos 5, 5'y 5''), realizando luego la búsqueda en sus bases de datos locales (Pasos 6, 6'y 6'') y enviando un mensaje con los resultados a sus servidores de mensajes locales (Pasos 7, 7'y 7''). Los mensajes con resultados son llevados hacia el nodo que realizó la consulta (en la figura, una biblioteca de la UNI-FIIS) en distintos momentos (Pasos 8, 8'y 8''), siendo recibidos por el servidor de mensajes y delegados al servidor de aplicaciones (Paso 9), quien finalmente da un formato adecuado a todos los resultados y lo presenta al usuario (Paso 10).

Una de las piezas claves en este diseño es el relacionado con la mensajería provista por los componentes EJB empresarial (Middleware Orientado a Mensajería - MOM):

- DextersLibExternalNodeManagerEJB.jar
- DextersLibResourceManagerEJB.jar
- DextersLibResourceManagerWebService.jar

De estos es el componente DextersLibResourceManagerEJB.jar el que permite gestionar los recursos independientes de las singularidades de cada nodo, y es el componente DextersLibExternalNodeManagerEJB.jar el que efectúa dichas solicitudes remotas externas y quien posee las funcionalidades MOM.

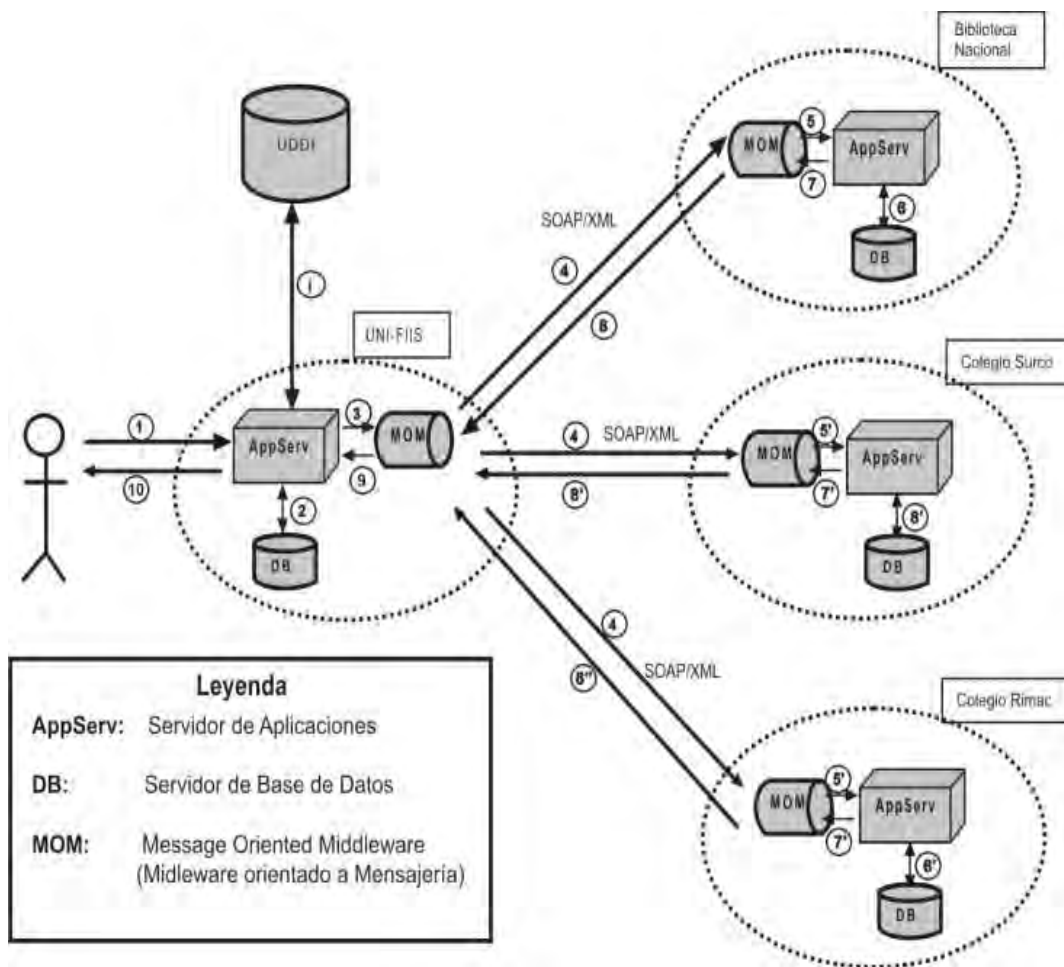


Figura 3.13: Esquema de comunicación entre los nodos del sistema de biblioteca digital distribuida basado en SOA-J2EE

Entre las herramientas de software libre utilizadas (basándonos en la arquitectura anteriormente descrita) se encuentran:

- JBoss (servidor de aplicaciones J2EE) / JBossMQ (Servidor de Mensajes).
- MySQL / Hypersonic SQL Database (Servidor de Base de Datos Relacional).
- Tomcat como contenedor de Servlets y JSP.

- Apache como Servidor Web.
- Axis para implementar el protocolo SOAP.
- Ant para el despliegue de los componentes.
- Struts para la capa de presentación.

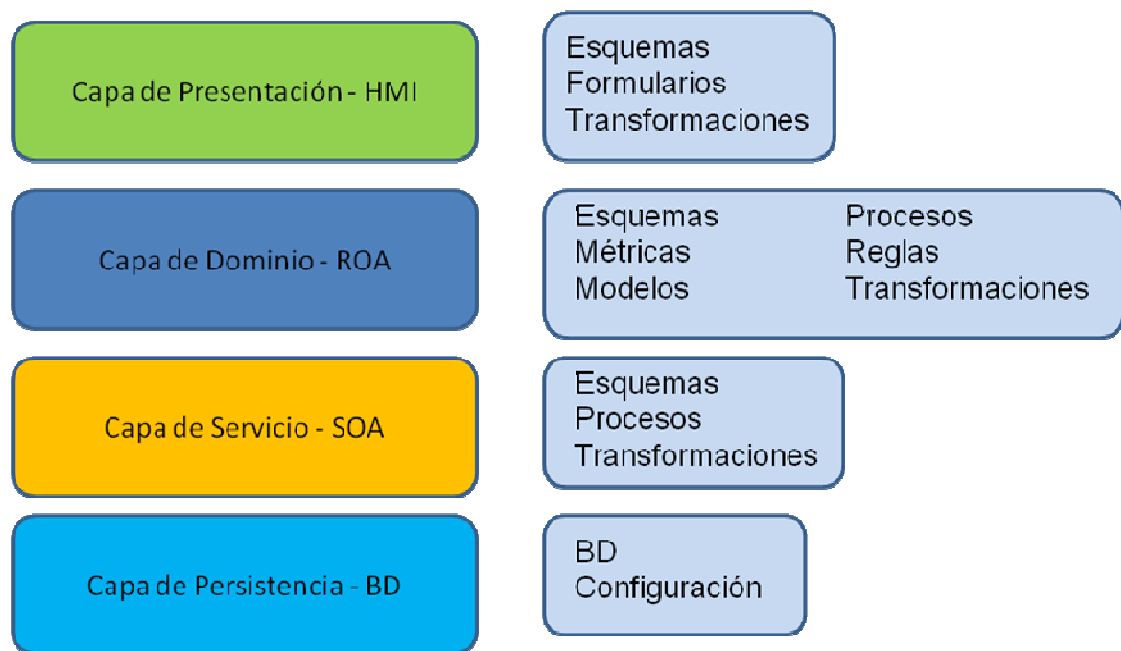


Figura 3.14: Diagrama de capas que compone el sistema de biblioteca digital distribuida basado en ROA-BPM

La contraparte del sistema SOA-J2EE basado en ROA-BPM se muestra en las figuras 3.14 y 3.15; en los cuales se detalla la separación en capas de los componentes o módulos internos del sistema así como su distribución a nivel de diagrama topológico de interconexión. En el Cuadro 3.12 se muestra la correspondencia directa entre un enfoque tradicional SOA-J2EE versus el enfoque ROA-BPM propuesto.

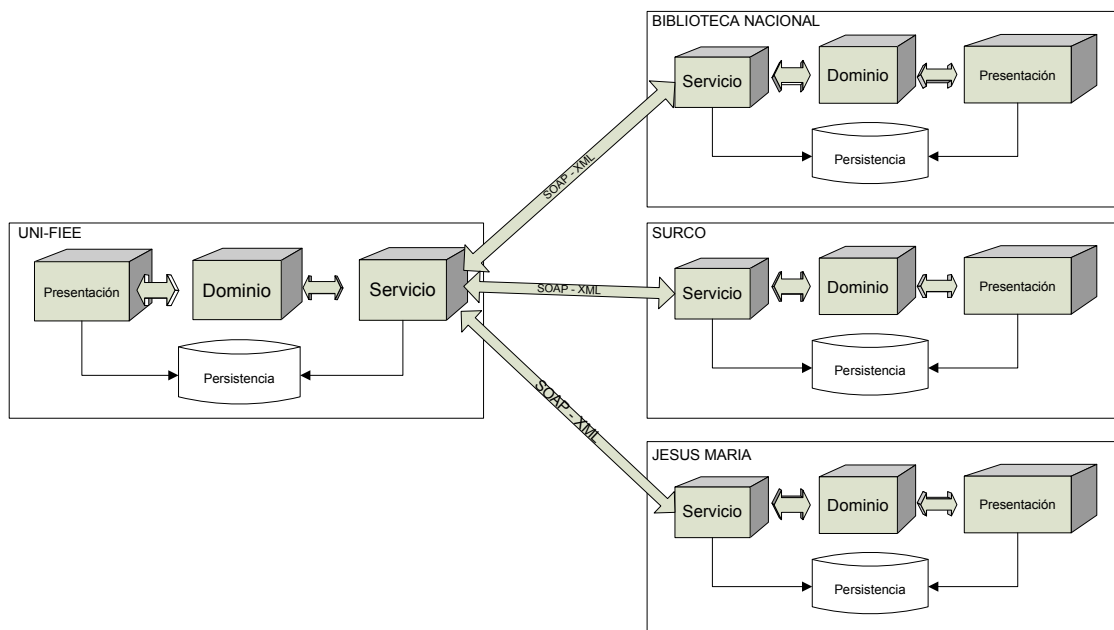


Figura 3.15: Esquema de comunicación entre los nodos del sistema de Biblioteca Digital Distribuida basado en ROA-BPM

Parámetro de Comparación	SOA-J2EE	ROA-BPM	Comentario
Presentación	JSP, Servlets	AJAX, WSDL	ROA se basa en SOA.
Dominio	UML, Java	BPMN, BPEL	
Persistencia	EJB-QL	SQL, XSD	
Servicio	WSDL	WSDL, XSLT	
Metodología	RUP-J2EE	RUP-BPM	
Modelado	UML	BPMN, ASME	
Regulación	NA	ITIL, NTP, etc	Valor agregado ROA.
Gestión	NA	PMBOK, 6Sigma	

Cuadro 3.12: Comparación de tecnologías y metodologías utilizados en las arquitecturas SOA-J2EE y ROA-BPM

3.2.2. Diseño Educativo

Es importante tener en cuenta el Diseño Educativo luego del análisis documentado, ya que se puede empezar a tomar decisiones concretas para el material y por tener un fin educativo se debe considerar todos los elementos que permitirán dar una solución a las necesidades detectadas en la etapa de análisis del proyecto de biblioteca distribuida.

En lo que respecta al diseño educativo, existen dos flujos de procesos en paralelo. El primer flujo de proceso está orientado a la parte administrativa de atención en cara a los usuarios internos y externos de la biblioteca digital distribuida. El segundo flujo de proceso está enmarcado en lo que respecta a la parte del núcleo base del proceso, esto es el producto final en lo que respecta al contenido del material bibliográfico para el usuario. En el proyecto solo se ha considerado el análisis diseño e implementación del prototipo del flujo del proceso administrativo enmarcado en el Registro de Usuario de Biblioteca.

En escenarios donde se implementan la gestión de sistemas de contenido digital, el enfoque varía, ya que se focaliza más en el producto, para lo cual se da énfasis en la etapa de diseño educativo, tales como:

Metas de Aprendizaje: Las metas reflejan las intenciones básicas del material y se formulan en términos de estados o procesos.

Objetivos de Aprendizaje: Los objetivos generales son afirmaciones que especifican lo que el alumno sería capaz de hacer como resultado del aprendizaje, señalan los pasos que conducen a las metas perseguidas expresándolas de manera sistemática y se formulan en términos de productos o resultados.

Decisiones de contenido: En esta etapa se tomarán las decisiones fundamentales para el contenido del material, se decidirán los elementos

específicos del tema escogido que se van a incluir en el material, es importante recordar que es mejor desarrollar los contenidos que solucionen una necesidad educativa específica y no necesariamente cubrir todo el tema en su extensión.

Modelo Cognoscitivo: El desarrollo de modelos cognoscitivos es un aspecto fundamental en la actividad de aprendizaje. La gente desarrolla modelos como resultado de sus experiencias y los utiliza para almacenar información y conocimiento. Pueden también utilizarse para guiar cualquier aprendizaje necesario, por lo que el docente debería escoger el que mejor se ajuste a su población, al tipo de material y al tema trabajado.

Prototipo en Papel: Este prototipo ayudará a plasmar las ideas en papel, el docente podrá realizar el prototipo como desee, siempre que permita visualizar todo lo anteriormente desarrollado en la etapa de análisis y en la de diseño educativo.

Estas consideraciones son de importancia en el caso que estuviéramos implementando un sistema de gestión de contenidos digitales de la biblioteca digital distribuida.

En nuestro caso, estos parámetros son irrelevantes ya que no estamos desarrollando material digital para la biblioteca como parte de un sistema de gestión de contenido, sino más bien nos estamos centrando nuestro alcance en el flujo del proceso de información para soporte administrativo del proceso de biblioteca digital distribuida.

3.2.3. Diseño de Comunicación

La forma en el cual va a interactuar el usuario es a través de una interface web, para lo cual se quiere que el usuario vea e interactúe con estas interfaces, de esta forma se diseñará el ambiente en el que se va a mover el estudiante utilizando un computador.

Requerimientos Funcionales

Se deben conocer y tener en cuenta las características de desarrollo físico y mental de los usuarios para saber de qué dispositivos y ayudas para la comunicación usuario-computador se requieren, estos requerimientos deben estar por escrito. Una vez se haya definido los requerimientos funcionales, la metáfora o paradigma y la interfaz a utilizar, se debe especificar el manejo que tendrá la herramienta, si se utilizará el teclado, el ratón, una pantalla sensible al tacto, etc.

Mapas de Navegación

Para tener una mayor claridad sobre la estructura general, la lógica de funcionamiento y el flujo de información del material es importante realizar un mapa de navegación que consiste en un esquema donde se incluyen todos los sitios, pantallas y rutas posibles de interacción del material.

Pantallas de Esquema

Para ir delimitando todo lo diseñado es importante realizar unos bocetos donde ya apliquemos todo lo pensado en la interfaz, no serán exactos a la interfaz definitiva pero ayudarán al diseñador a aproximarse con mayor precisión a lo que tenemos en mente. Las pantallas de esquema deben representar cada tipo de página que se quiere desarrollar. Con las pantallas de esquema se puede armar una presentación en alguna herramienta de propósito general que permita simular el comportamiento de la herramienta final.

En la plataforma ROA-BPM se cuenta con un módulo denominado Workflow que a diferencia de un portal, permite proporcionar las interfaces de interacción de los usuarios con el sistema automatizado, para lo cual considera tres categorías

de interacción:

1. Procesos
2. Notificaciones
3. Tareas

La herramienta de diseño BPM de Intalio cuenta con un módulo denominado Intalio – Designer, que permite generar prototipos rápidos para esquematizar las pantallas de usuarios que serán utilizados en el workflow. La tecnología que utiliza es AJAX-TIBCO que permite utilizar código de javascript sobre XML para las pantallas gráficas de usuario.

3.2.4. Diseño Computacional

En este punto es importante señalar las alternativas existentes a los Web Services, en el cuadro 3.13 se muestra la comparación entre DCOM, CORBA y Web Services, donde se aprecia la ventaja de la flexibilidad de esta última tecnología usando XML y un acoplamiento ligero cliente servidor. Un punto interesante es que el registro UDDI de IBM fue desarrollado por el grupo de desarrollo de CORBA, quienes reutilizaron varios de los objetos CORBA previamente desarrollados.

Una típica plataforma comercial J2EE incluye uno o más contenedores. Un contenedor J2EE es un periodo de ejecución para gestionar los componentes de la aplicación desarrollados según las especificaciones del API y destinados a proporcionar acceso a las API de J2EE, el que no especifica ninguna identidad para contenedores.

Detalle	DCOM	CORBA	Web Services
Nombrado de Puntos Finales	OBJREF	Referencia de Objeto Interoperable (IOR)	URL/URN
Interfaces	Único	Múltiple	Múltiple WSDL
Tipos Payload	Binario	Binario	Texto
Formato Payload parámetro por valor	Representación de Data de Networking (DR)	Representación de Data Común	XML
Resolución de dirección de Servidor	Directorio	Servicio de Nombrado	Ruteo IP, URN
Despachador de Mensaje	Basado en ID	Basado en ID	Tipos de parámetros y Namespace
Acoplamiento Cliente - Servidor	Fuerte	Fuerte	Débil

Cuadro 3.13: Comparación entre DCOM, CORBA y los Web Services

Del mismo modo, la plataforma BPM de Intalio ofrece comunicación nativa a través de Web Services autogenerados a partir de la notación BPMN ó también a partir de las estructuras SQL. En ambos escenarios, el enfoque ROA-BPM es más ágil en cuanto a tiempos de análisis y diseño que el enfoque tradicional SOA-J2EE.

3.2.5. Preparación y revisión de un prototipo

En el proyecto de Tesis, se preparó inicialmente un prototipo de biblioteca digital distribuida sobre SOA, poniéndose énfasis en la tecnología basada en Web Services, aplicando el patrón Modelo Vista Controlador, esto es: EJB para

el modelo (lógica de negocios a través de los entity beans, modelo de datos a través de los session beans), las vistas y los controladores a través del framework Struts.

Posteriormente se migró el mismo prototipo de biblioteca digital distribuida sobre ROA-BPM, poniéndose énfasis en la tecnología basada en Web Services aplicando patrones de procesos y una estructura de separación por capas, esto es:

- Capa de Presentación: se utiliza Ajax versión TIBCO
- Capa de Dominio: se utiliza BPMN con XSD y XSLT
- Capa de Persistencia: se utiliza BPMN y SQL
- Capa de Servicio: se utiliza BPMN con XML y WSDL

3.3. Desarrollo de la Biblioteca Digital Distribuida

3.3.1. Definición de Estrategia

Según la metodología convencional de Rational Unified Process (RUP), se ha determinado el siguiente cronograma de actividades en lo que a fases y actividades se refiere:

Comienzo Esta etapa contempla:

- Formular la amplitud del proyecto
- Determinar y evaluar los riesgos del proyecto
- Preparar el entorno del proyecto
- Integración de los recursos

Elaboración Esta etapa contempla:

- Definir la arquitectura del sistema

- Refinar la visión del proyecto
- Crear un plan de iteraciones para la fase de Construcción
- Adecuar los componentes a la arquitectura establecida

Construcción Esta etapa contempla:

- Completar el desarrollo de componentes
- Pruebas unitarias
- Generación de releases que cumplan algún criterio de evaluación
- Decidir una ubicación para la instalación del prototipo

Transición Esta etapa contempla:

- Pruebas finales de usuario
- Puesta a punto
- Generación de una versión última de la fase

Estas actividades han sido adecuadas a la nueva arquitectura ROA-RUP considerando roles para el análisis de los procesos, en concreto:

- Analista de Procesos Funcionales
- Analista de Procesos Operativos
- Implementador de Documentación
- Arquitecto de Base de Datos
- Arquitecto de Web Services
- Implementador de Pruebas
- Integrador de Sistemas
- Auditor de Calidad de Software
- Implementador de Sistemas

En la figura 3.16 se aprecia los actores en mención.

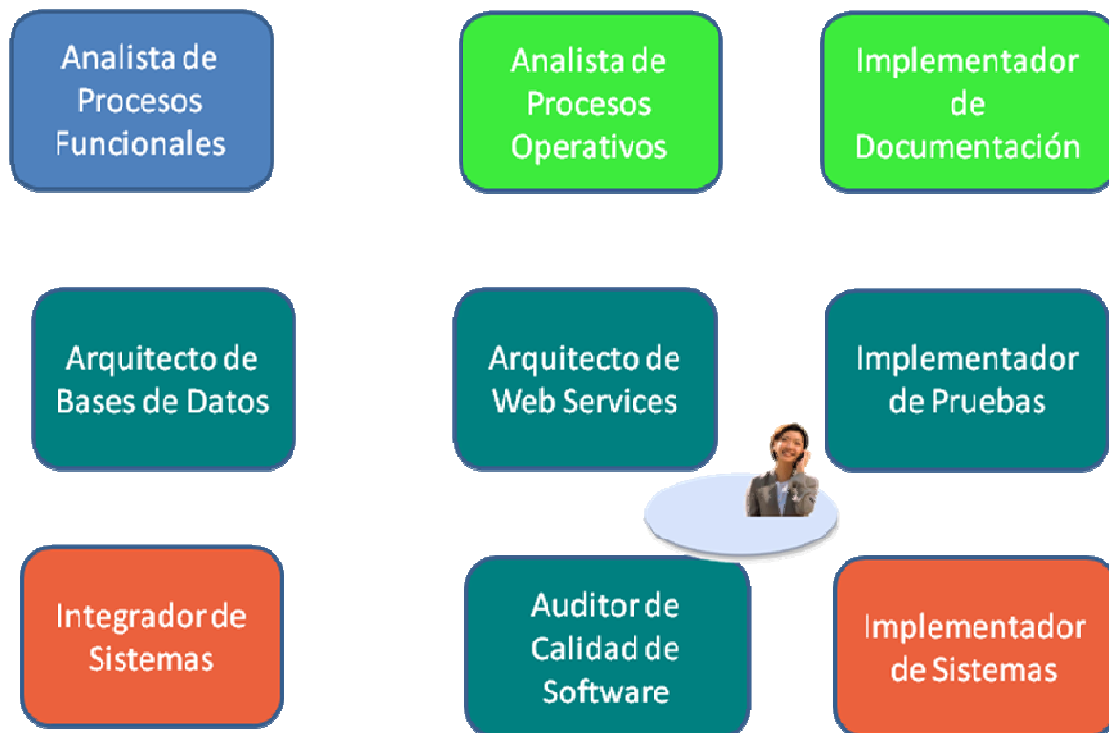


Figura 3.16: Roles de los Actores ROA-BPM utilizado en el desarrollo del sistema de Biblioteca Digital Distribuida

3.3.2. Detalle del Lenguaje SOA-UML versus ROA-BPMN en el diseño del Sistema Dexter's Lib

La finalidad de los diagramas es presentar diversas perspectivas de un sistema a las cuales se les conoce como modelo, es decir nos muestran diferentes aspectos de los sistemas que son modelados, definiendo una serie de vistas, cada una mostrando un aspecto particular del sistema, la cual puede ser percibida como una imagen completa del sistema. Las vistas también enlazan el lenguaje de modelaje al método o proceso escogido para el desarrollo.

Inicialmente se desarrolló el prototipo utilizando el enfoque tradicional SOA-UML, sin embargo debido a que en su mayoría los casos de uso estaban orientados a los procesos administrativos de la institución, resulta más eficaz utilizar ROA-BPM puesto que se centra mayormente en el flujo de información del proceso

administrativo que a los detalles de interfaz gráfica de usuario. Para evaluar la ventaja de esta arquitectura se muestra el diseño del mismo prototipo, pero con los dos enfoques.

Enfoque Tradicional del Prototipo Dexter's Lib con Diagramas SOA-UML

En un escenario típico UML se emplea versiones personalizadas de las siguientes vistas:

- Vistas de Diseño
- Vista de Implementación
- Vista de Casos de Uso
- Vista de Procesos
- Vista de Despliegue

Los cuales utilizan los diagramas que se mencionan a continuación, pero que en un proyecto determinado no necesariamente se utilizan todos sino algunos de los diagramas más importantes para el proyecto, a continuación se lista estos diagramas:

- Diagrama de Casos de Uso
- Diagrama de Clase (incluye implícitamente el Diagrama de Objetos)
- Diagramas de Comportamiento
- Diagrama de Estados
- Diagrama de Actividad
- Diagramas de Interacción
- Diagrama de Secuencia
- Diagrama de Colaboración
- Diagrama de Componentes
- Diagrama de Despliegue

Inicialmente se implementó el prototipo con UML (ver figura 3.22):

- Registrar Recurso.
- Buscar Recursos.
- Agregar Feedback.

Estos mismos diagramas UML han sido implementados vía BPMN a través del flujo del proceso modelados con ROA-BPM (ver figura 3.17).

En el enfoque tradicional UML con J2EE tenemos la siguiente nomenclatura:

Node: Representa los diversos nodos remotos Dexter Lib, ubicados físicamente en cada región.

Resource: Representa los recursos digitales que pueden ser un libro, un video o una animación.

Book: Clase que representa a un libro.

Video: Clase que representa a un video.

Animation: Clase que representa a una animación.

Person: Clase que representa a una persona.

ResourceFeedback: Clase que representa los comentarios de los recursos, contiene objetos del tipo CommentFeedback.

CommentFeedback: Clase que representa el propio comentario.

Los Diagramas de Secuencia y de Colaboración son usados para establecer un escenario del sistema, determinando los objetos y mensajes involucrados. Un Diagrama de Secuencia muestra los objetos de un escenario mediante líneas verticales y los mensajes entre objetos como flechas conectando objetos. El Diagrama de Colaboración modela la interacción entre los objetos de un Caso de Uso, los objetos están conectados por enlaces (links) en los cuales se representan los mensajes enviados acompañados de una flecha que indica su dirección.

El Diagrama de Colaboración ofrece una mejor visión del escenario cuando el analista está intentando comprender la participación de un objeto en el sistema.

El Diagrama de Estados modela el comportamiento de una parte del sistema a través del tiempo.

Típicamente se elabora un diagrama de Estados para cada clase que tenga un comportamiento significativo.

El comportamiento es modelado en términos del estado en el cual se encuentra el objeto, qué acciones se ejecutan en cada estado y cuál es el estado al que transita después de un determinado evento.

Los Diagramas de Actividad son casos especiales de los Diagramas de Estados donde todos los estados o la mayoría son estados de acción. Asimismo todas o la mayoría de las transiciones son **disparadas** como consecuencia de la finalización de la acción.

El Diagrama de Actividades puede estar asociado a:

- Una clase.
- La implementación de una operación.
- Un caso de uso.

Un diagrama de Componentes permite modelar la estructura del software y la dependencia entre componentes. Un componente es un grupo de clases que trabajan estrechamente. Los componentes pueden corresponder a código fuente, binario o ejecutable. El Diagrama de Distribución o Despliegue modela la distribución en tiempo de ejecución de los elementos de procesamiento y componentes de software junto a los procesos y objetos asociados.

Enfoque Propuesto del Prototipo Dexter's Lib con Diagramas ROA-BPM

En un escenario ROA-BPM se emplea la notación BPMN para implementar el flujo del proceso previamente identificado con la metodología ROA-GTZ.

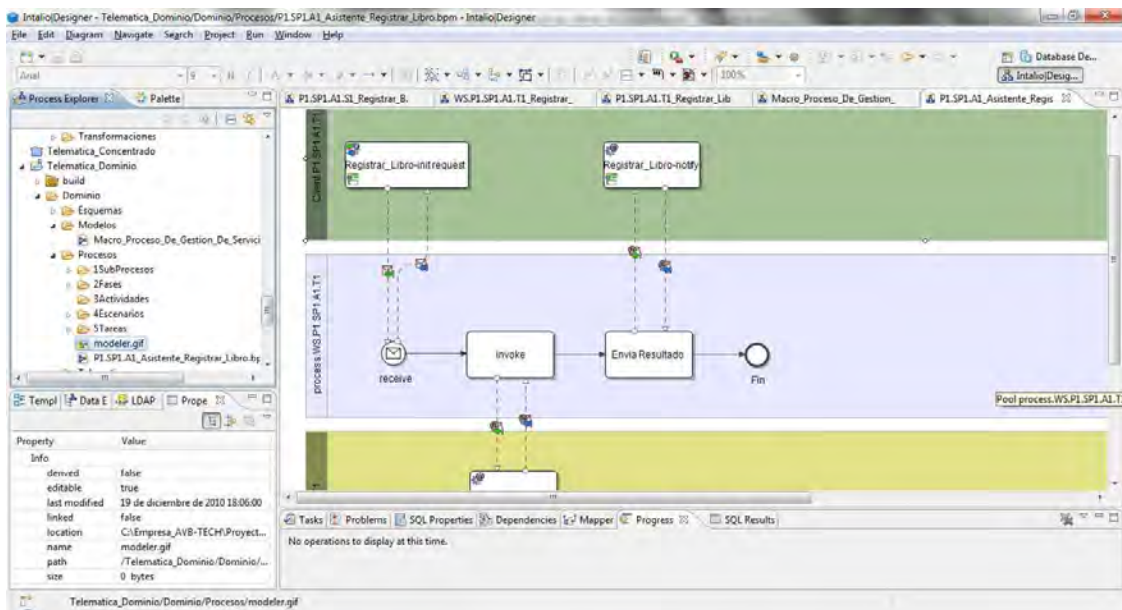


Figura 3.17: Modelado de la capa de dominio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida

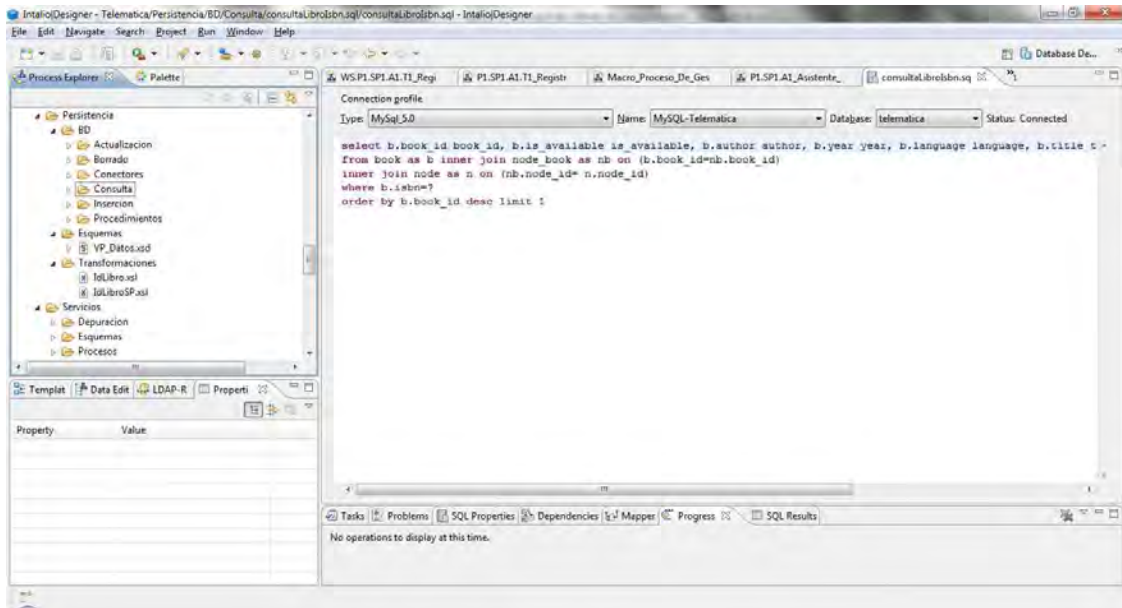


Figura 3.18: Modelado de la capa de persistencia ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida

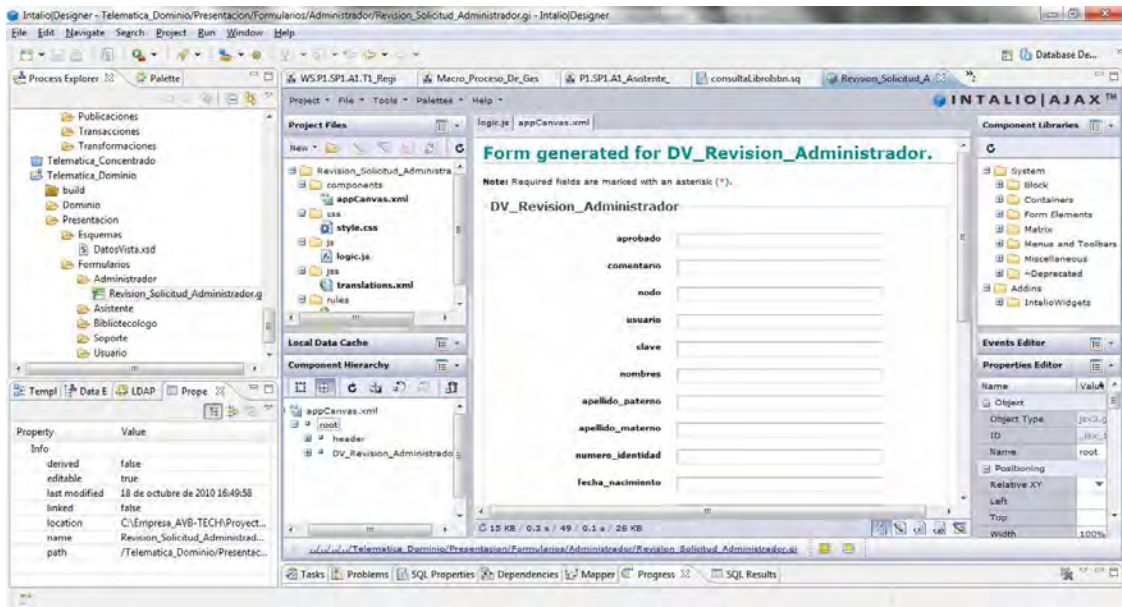


Figura 3.19: Modelado de la capa de presentacion ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida

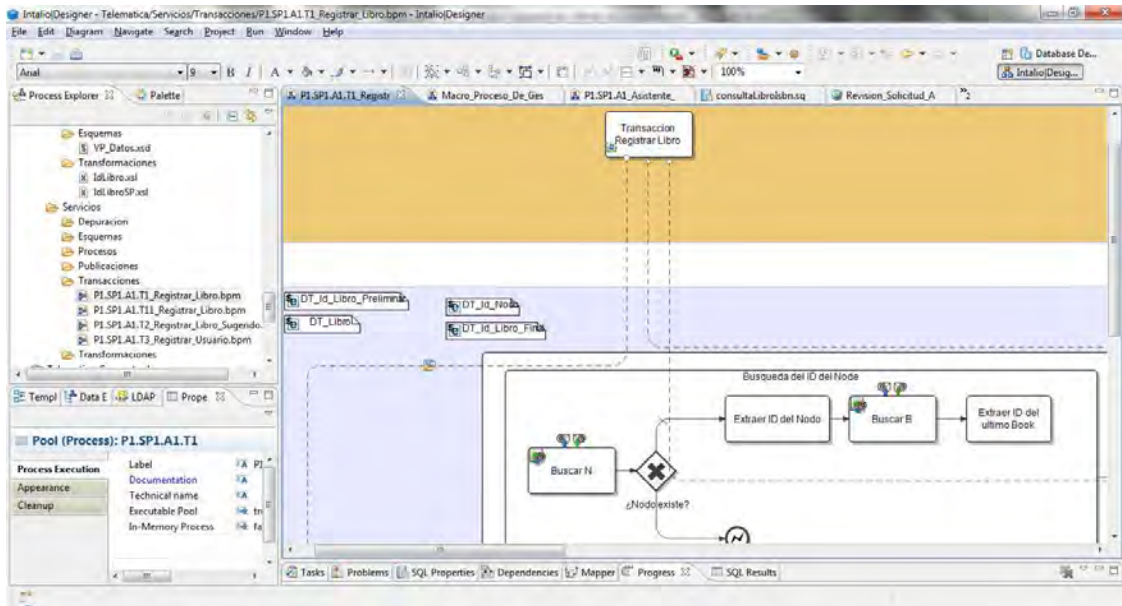


Figura 3.20: Modelado de la capa de servicio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida

Es importante recalcar que los detalles del código ocupan cientos de páginas BPMN y BPEL, de ahí que sólo se ha mostrado aquellas partes más importantes, sugiriendo la lectura del CD de la tesis que contiene la totalidad del código fuente explicada para mayor referencia. Asimismo se han agregado videos de instalación y pruebas, donde se esboza a detalle el procedimiento de despliegue y pruebas remotas.

Comparación entre la Metodología utilizada en la implementación SOA-J2EE versus ROA-BPM

La implementación del prototipo fue desarrollada utilizando una variante de la metodología RUP [ver referencia 18] pero adaptado para ROA, en donde el proyecto de software se divide en fases que sirven para representar el plan estratégico para el proyecto, ya que permite fijar las metas y objetivos de cada ciclo. Cuando se define una metodología es importante tener en cuenta el escenario donde ésta se aplica. En este caso está circunscrito a los sistemas de información. (ver Figura 3.21)

En el cuadro 3.14 se muestra un análisis comparativo de la metodología de implementación tradicional RUP para SOA-J2EE versus RUP para ROA-BPM. Cabe resaltar que ROA supera a SOA debido principalmente a que está concebido como una macro arquitectura que contiene a SOA, agregándole funcionalidades adicionales tales como metodologías, tecnologías regulaciones y líneas de desarrollo base.

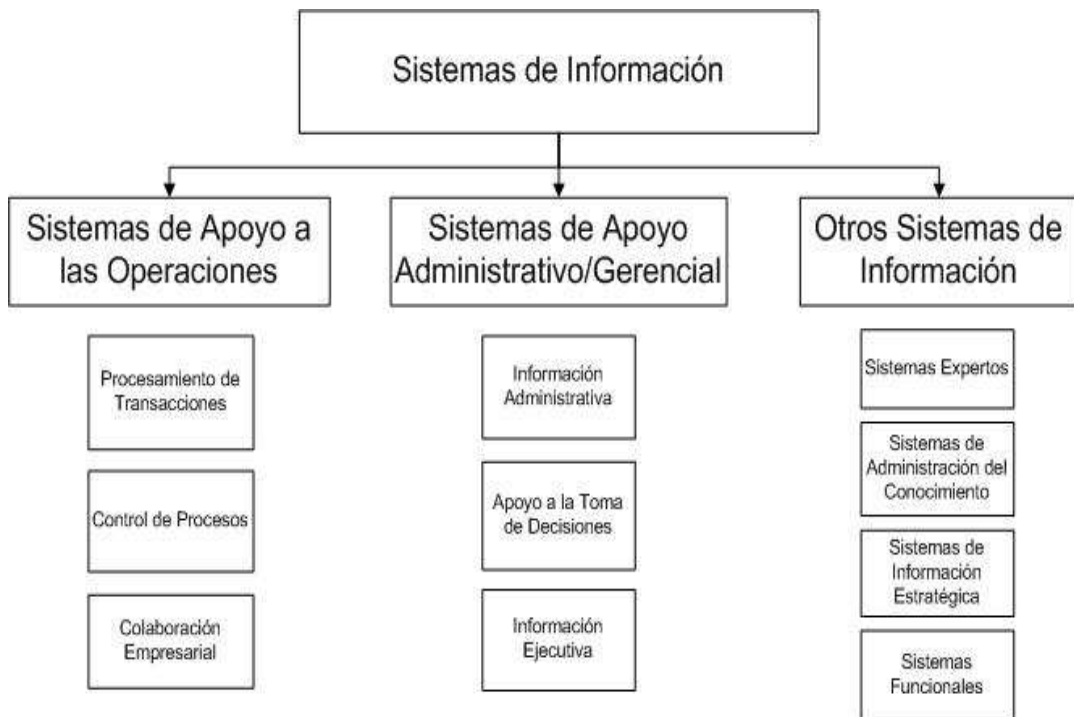


Figura 3.21: Modelado de la capa de servicio ROA-BPM utilizado en el desarrollo del Sistema de Biblioteca Digital Distribuida

Parámetros de Comparación	Arquitectura Orientada a Servicios	Arquitectura Orientada a Regulaciones	Comentarios
Metodologías de Dirección	No está definido.	PMBOK, RUP, CMMI, etc.	ROA es una macro Arquitectura SOA orientado a Procesos y Aplicaciones.
Tecnologías de Soporte	Depende del proyecto.	J2EE, .NET, BPEL, PHP, etc.	
Infraestructuras de Regulación	No está claramente definido.	ITIL, COBIT, 6 SIGMA, NTP, etc.	
Líneas de Desarrollo	Depende del Proyecto.	BPM, CMS, ERP, BI, ESB, etc.	
Soporte a Futuras Mejoras	No	Si	

Cuadro 3.14: Comparación de la Arquitectura Orientada a Servicios SOA con Arquitectura Orientada a Regulaciones ROA

Con RUP se proveen hitos de negocio bien definidos que aseguran que las etapas estén progresando y converjan hacia una solución en lugar de reincidir indefinidamente. En cada fase hay una o más iteraciones. Cada fase concluye con un hito mayor y una evaluación para determinar si se ha cumplido con los objetivos. Una evaluación satisfactoria permite al proyecto moverse hacia la siguiente fase. Las iteraciones están basadas en el tiempo (6 meses como máximo), mientras que las fases están basadas en las metas.

En la Fase de Comienzo: se establece el problema a desarrollar, todos los stakeholders se ponen de acuerdo en los objetivos del proyecto. Asimismo se asegura que el proyecto es de valor y es posible realizarlo.

En la Fase de Elaboración: la arquitectura del software es establecida al proveer los cimientos estables para el diseño e implementación que es realizada durante la fase de Construcción.

En la Fase de Construcción: los requerimientos faltantes han sido clarificados y el desarrollo del sistema es completado basándose en la arquitectura básica establecida en la fase de Elaboración. Entre la fase de Elaboración y Construcción el foco de atención cambia, desde el entendimiento del problema e identificación de elementos principales de la solución, hacia el desarrollo de un producto instalable.

En la Fase de Transición: se asegura que el software está disponible, y es aceptado por los usuarios finales. Es en esta fase donde el sistema será instalado en el entorno del usuario para su evaluación y pruebas. El foco de atención es poner a punto el producto, configurarlo e instalarlo. Todas las inquietudes mayores sobre la estructura del sistema deben haber sido trabajadas mucho antes. Al final de esta fase, el proyecto debería estar en posición de ser cerrado.

3.3.3. Documentación del Proyecto

A continuación se detallan los puntos más relevantes desde el punto de vista de modelamiento y codificación de la biblioteca digital distribuida, es importante señalar que se adjunta un CD con toda la información de codificación del proyecto de software, así como el código fuente BPMN y ejecutable BPEL como parte complementaria del presente trabajo (ver anexo para mayor detalle).

Glosario del sistema Dexter's Lib

El propósito de este glosario es para definir los términos que son usados como referencia en múltiples lugares dentro de la documentación, modelos y demás artefactos del sistema Dexters Lib, a fin de proveer definiciones consistentes.

Este documento es usado para definir términos específicos al dominio del problema, explicando los términos que pueden resultar poco familiares a los lectores de las descripciones de casos de uso y otros artefactos.

Amplitud

Este glosario define los términos más comunes que son usados como referencias en múltiples lugares dentro de la documentación, modelos y demás artefactos del sistema Dexters Lib a fin de proveer definiciones consistentes.

Referencias

Ver documento del procedimiento, así como el video de instalación y despliegue en el CD complementario de la Tesis: Dexter Lib Vision (Versión en Español).

Definiciones

Término	Definición
Búsqueda Local	Se refiere al proceso de ubicación de recursos en el nodo local.
Búsqueda Remota	Se refiere al proceso de ubicación de recursos en nodos remotos.
Calificación en Criterio de Evaluación	Es el puntaje que el consumidor otorga a un recurso en un determinado criterio de evaluación.
Comentario	Es el texto que se agrega al describir un feedback para un recurso.
Consumidor	Se refiere a la entidad que utiliza los recursos para satisfacer una necesidad de información.
Consumidor Humano	Es aquel consumidor que no es un nodo.
Criterio de Evaluación	Es aquella característica de un recurso que puede ser evaluada.
Datos Complementarios	Son datos adicionales de un recurso que amplían el significado del mismo. Por ejemplo, para libros éstos serían: temas, editorial, idioma, tipo, año de publicación.
Datos de Características	Son datos adicionales de un recurso que permiten compararlos con otros similares, como por ejemplo: precio, dificultad, número de ejercicios.
Datos de Identificación	Son datos que permiten identificar a un recurso dentro de un amplio conjunto de recursos. Por ejemplo, para los libros, se tiene: título, autor, ISBN.
Datos de Localización	Son datos que permiten localizar al recurso geográficamente, por ejemplo: Solo en ésta biblioteca (Nodo Local), en mi organización, en mi distrito, en mi departamento, en mi país, en todo el mundo.
Detalles de Recursos	Es el conjunto de los siguientes datos: datos de identificación, datos complementarios, datos de

	características, datos de localización y feedback.
Feedback	Es toda aquella información dejada por los consumidores humanos para dar su opinión respecto de un recurso o comentario y mejorar la experiencia educativa de otros consumidores humanos.
Feedback a Comentario	Consiste en calificar un comentario dejado por un consumidor humano.
Feedback a Recurso	Compuesto por calificaciones en criterios de evaluación, además de un comentario.
Formulatio Registro de Feedback a Recurso	Consiste en calificar un comentario dejado por un consumidor humano.
Mensaje de Agradecimiento	Consiste en dejar mensaje.
Mi Lista de Recursos	Consiste en mostrar listado de Recursos.
Término	Definición
Nodo	Se refiere a una instancia del sistema Dexter's Lib, comúnmente ejecutada en un servidor con una dirección de internet única y cuyo funcionalidad puede ser compartida por otros nodos de Dexter's Lib. Dependiendo del punto de vista, los nodos pueden ser del tipo local o remoto.
Nodo Local	Es el nodo donde el usuario interactúa directamente.
Nodo Remoto	Es todo aquel nodo que coopera con el nodo local. La interacción con el usuario es indirecta.
Proveedor	Se refiere a la entidad que facilita los recursos al sistema Dexter's Lib.
Recurso	Es todo aquel material de consulta. Pueden ser libros electrónicos, libros tradicionales, animaciones, videos, etc.
Resultados Resumidos de la Búsqueda	Título, ISBN, autor(es), tipo, editorial, idioma, temas, calificación, localización.

Cuadro 3.15: Descripción de las definiciones usadas en el Prototipo de la Biblioteca Digital Distribuida

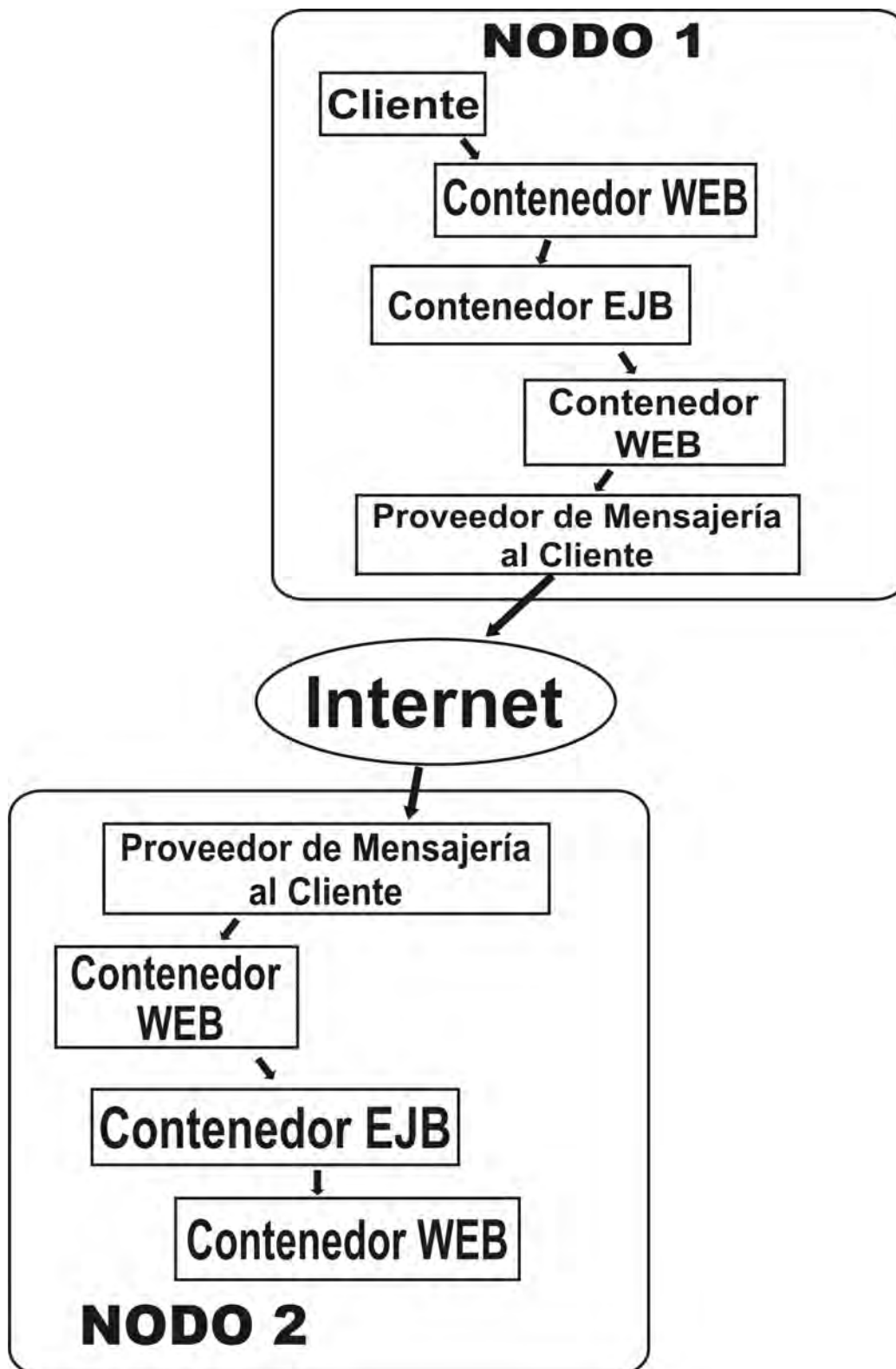


Figura 3.22: Vista lógica de la arquitectura de los componentes

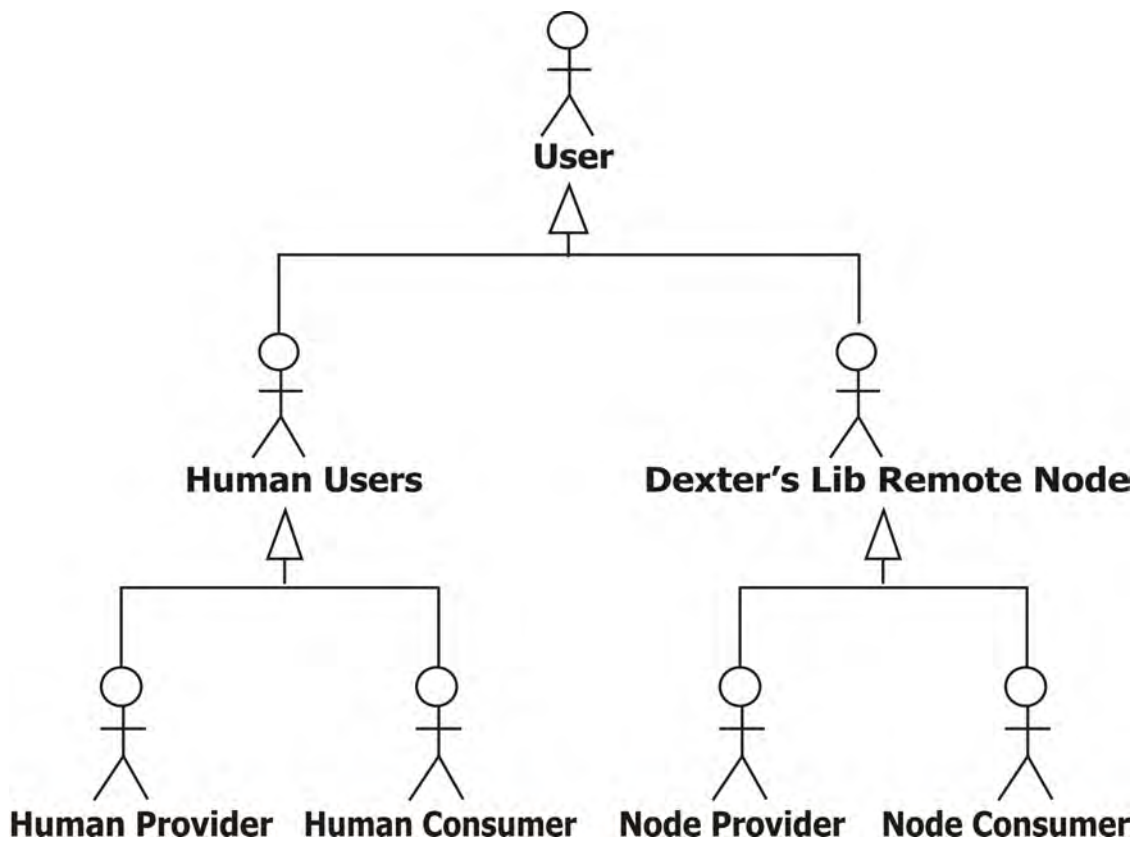


Figura 3.23: Actores del sistema de biblioteca

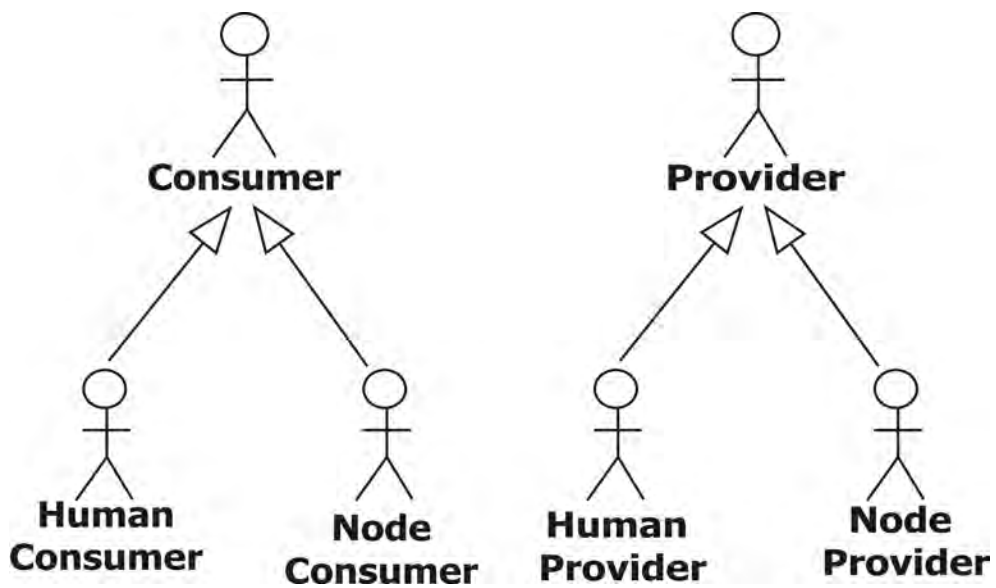


Figura 3.24: Proveedores y consumidores del sistema de biblioteca

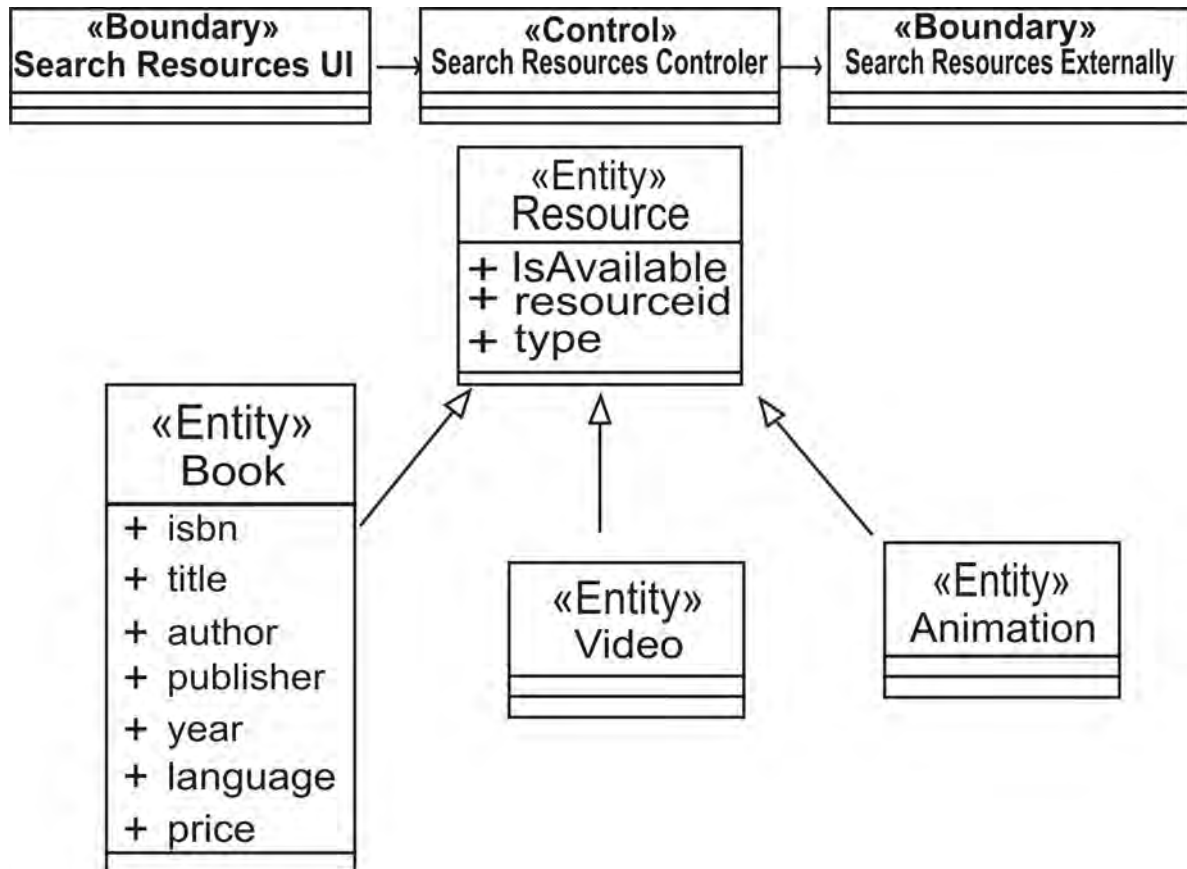


Figura 3.25: Participantes de búsqueda de recurso del sistema de biblioteca

Dexter's Lib Use Case Diagram

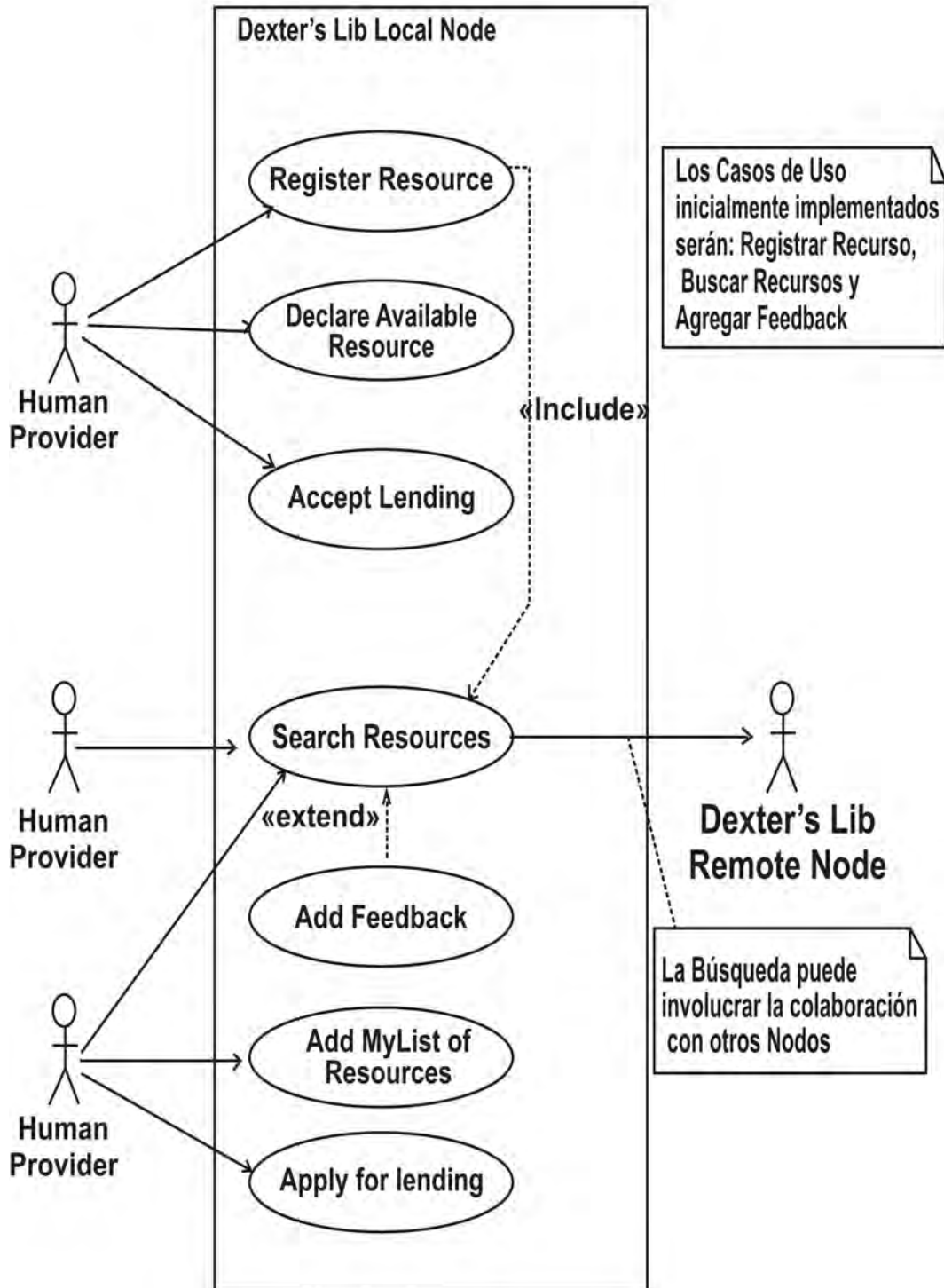


Figura 3.26: Diagrama de casos de uso

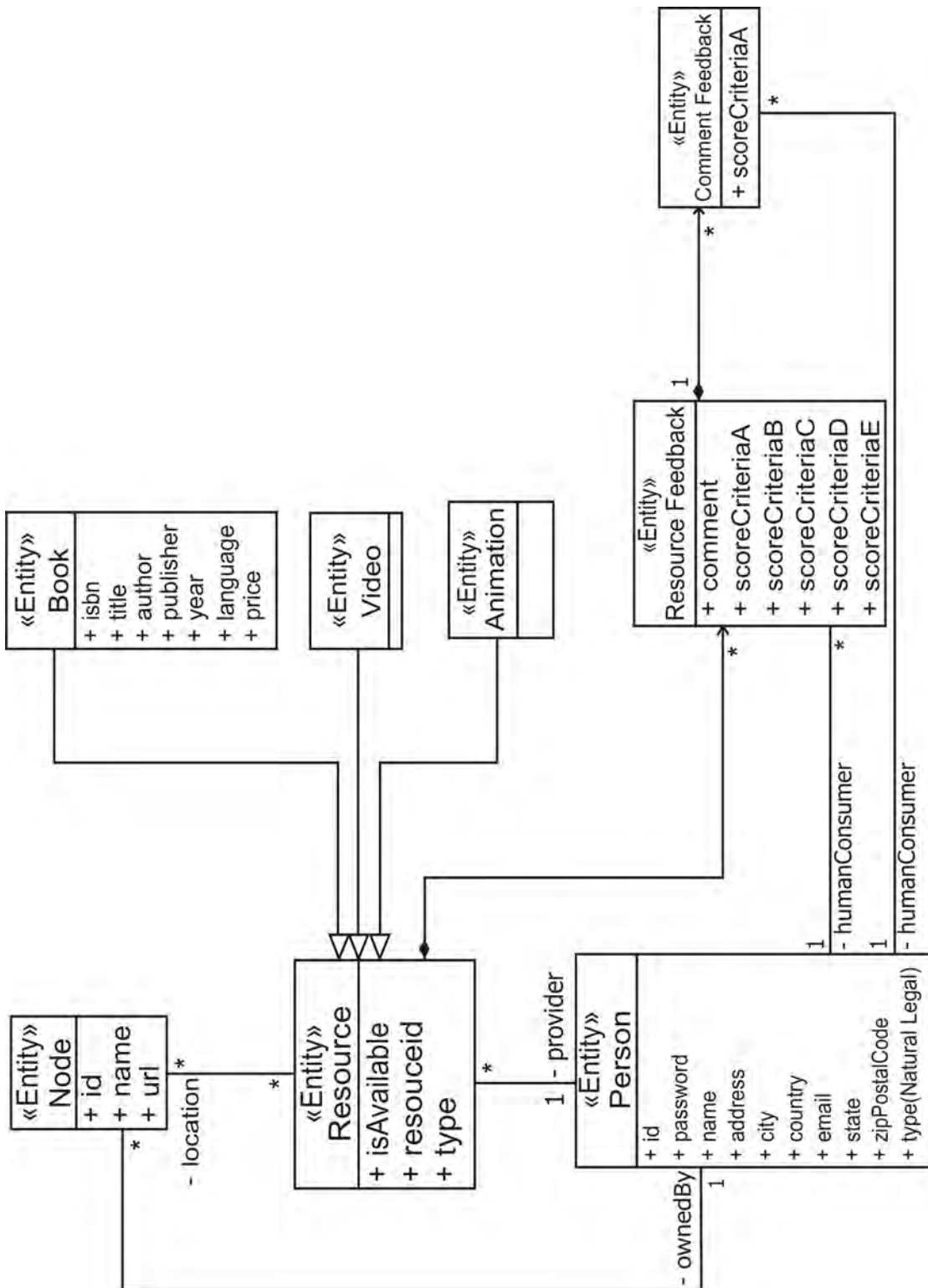


Figura 3.28: Diagrama de clases que se mapean a los Entity Beans, y estos a su vez se mapean a una tabla almacenada en la base de datos

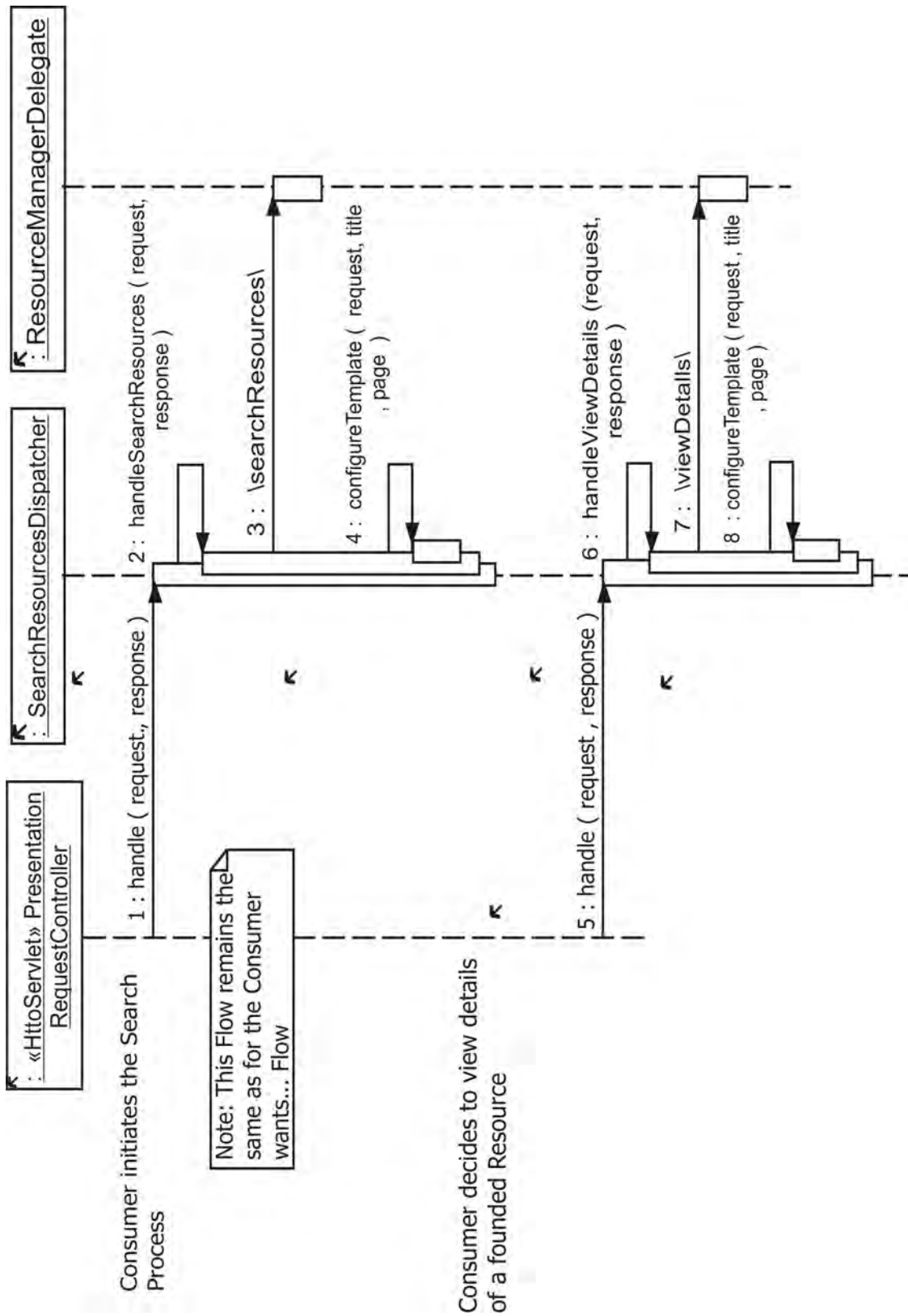


Figura 3.29: Diagrama de secuencia para caso de uso Búsqueda de Recursos

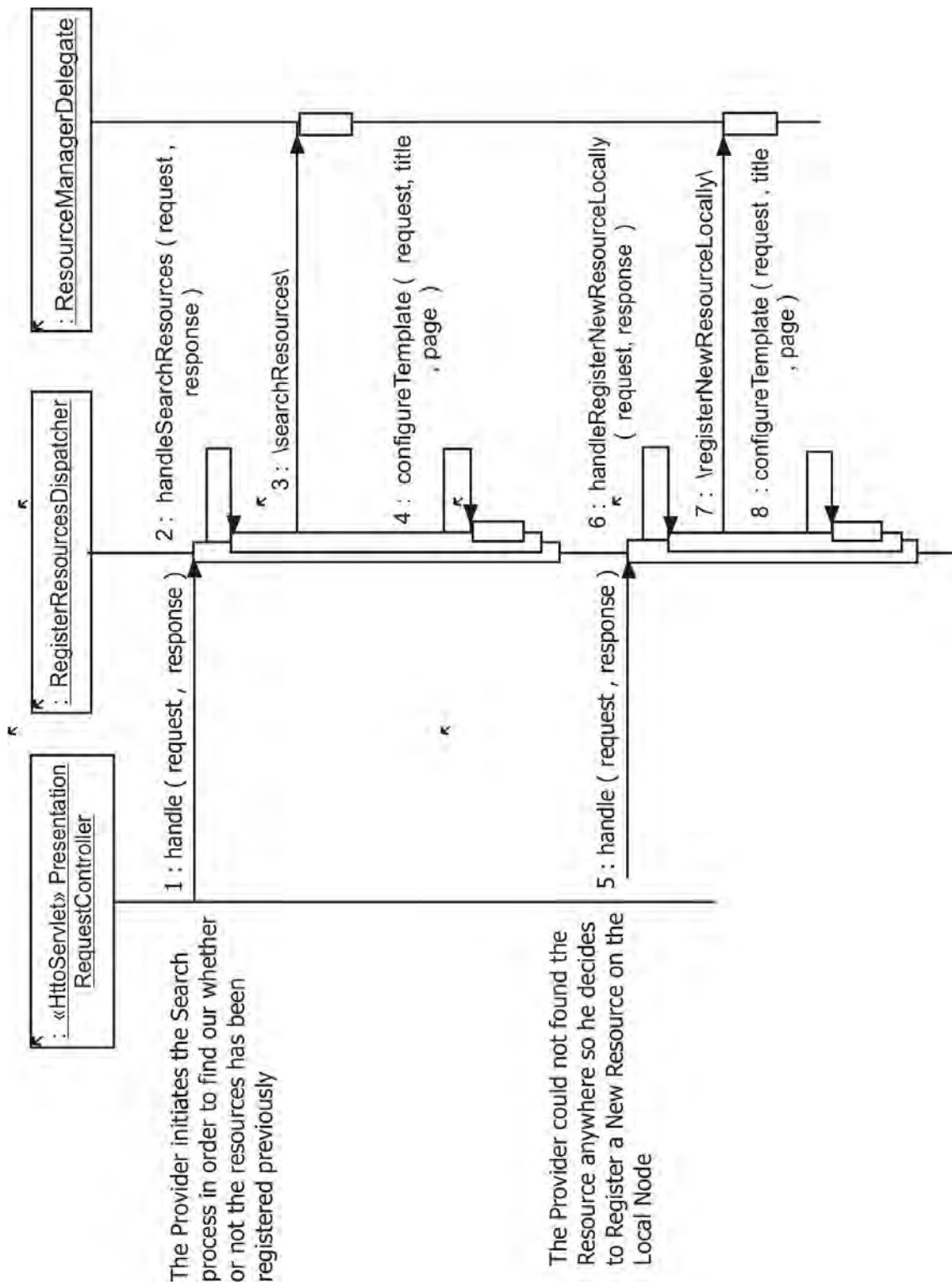


Figura 3.30: Diagrama de secuencia para caso de uso Registrar Recurso

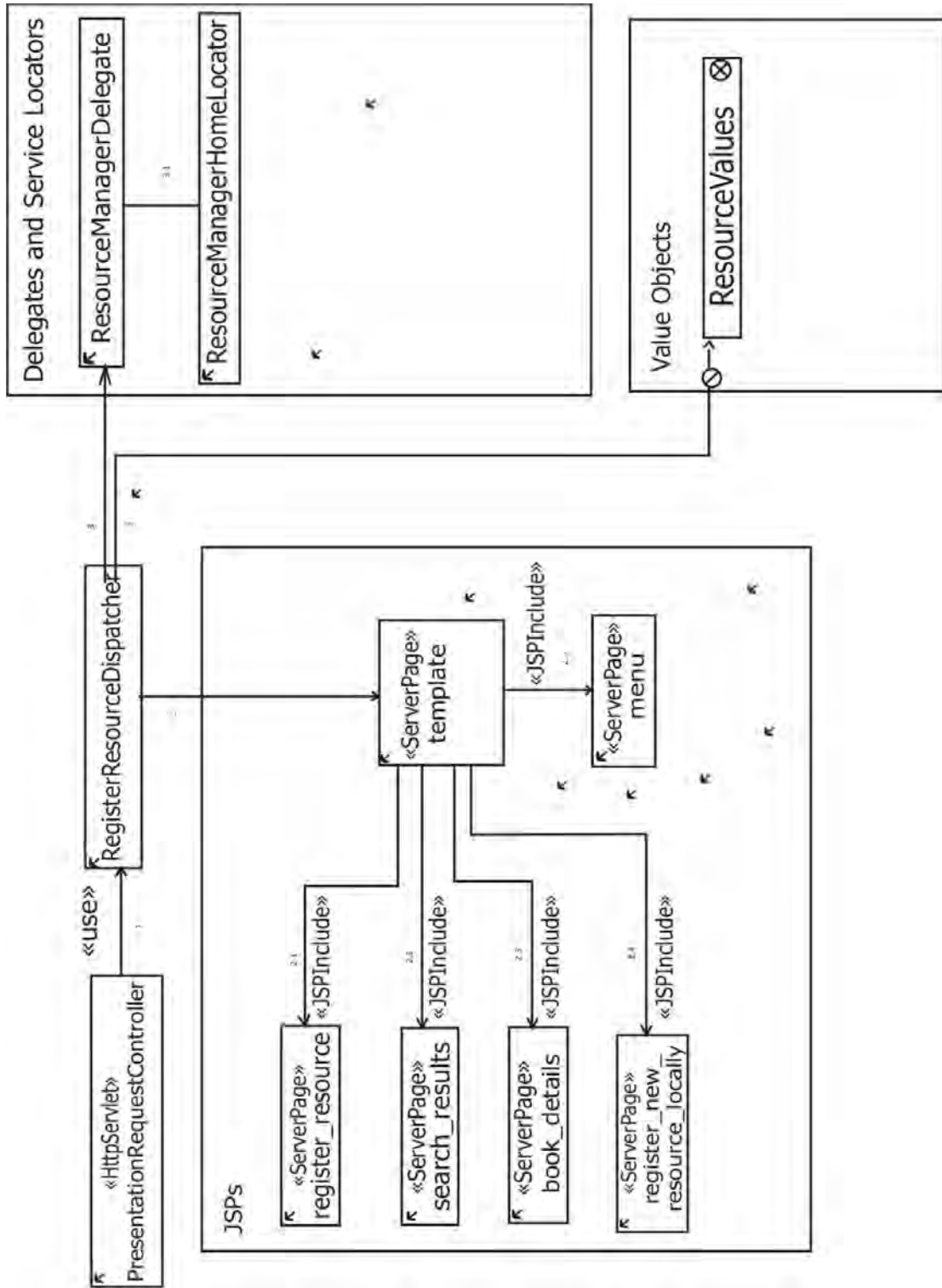


Figura 3.31: Diagrama de colaboración para caso de uso Registrar Recurso

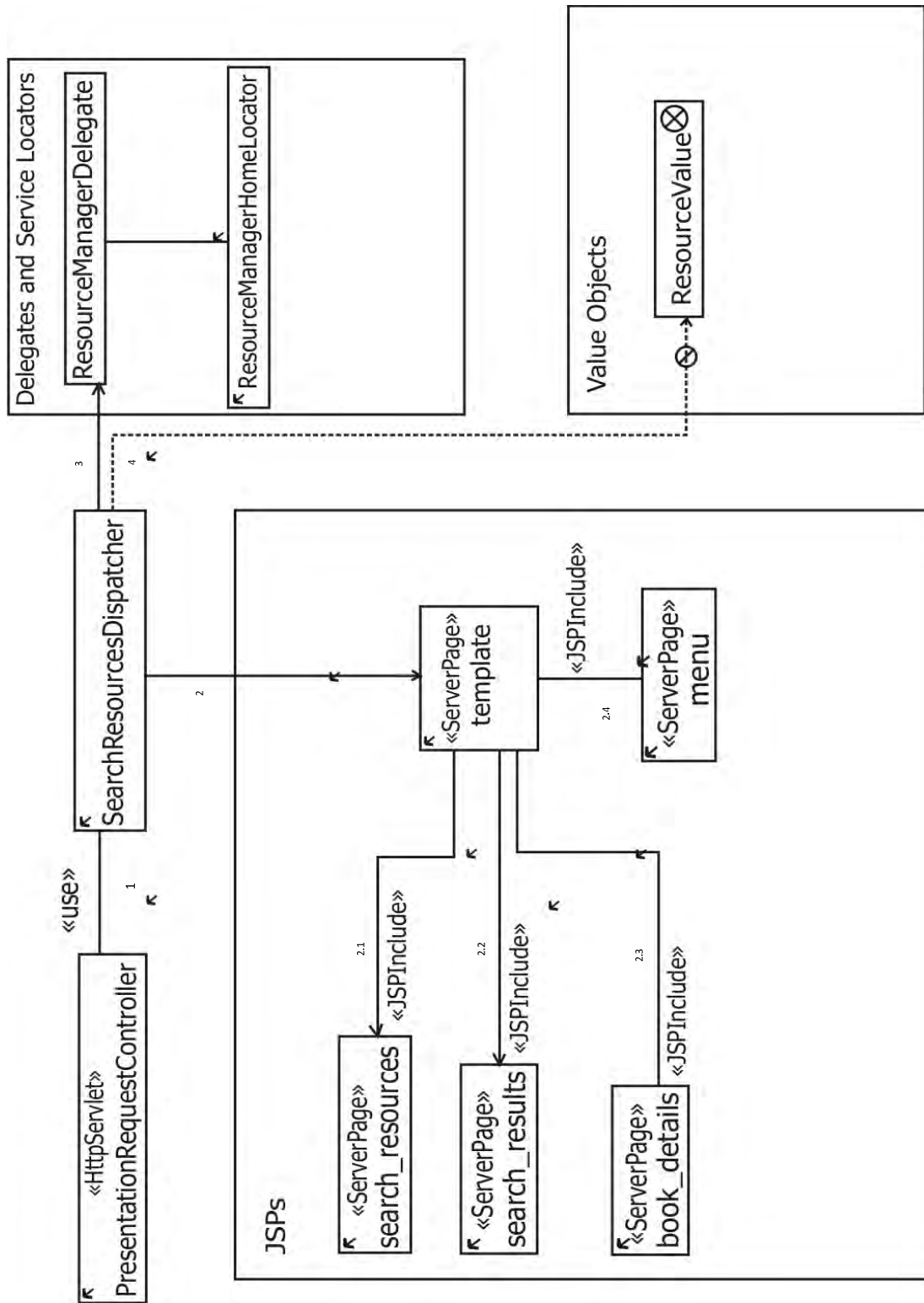


Figura 3.32: Diagrama de colaboración para caso de uso Búsqueda de Recurso

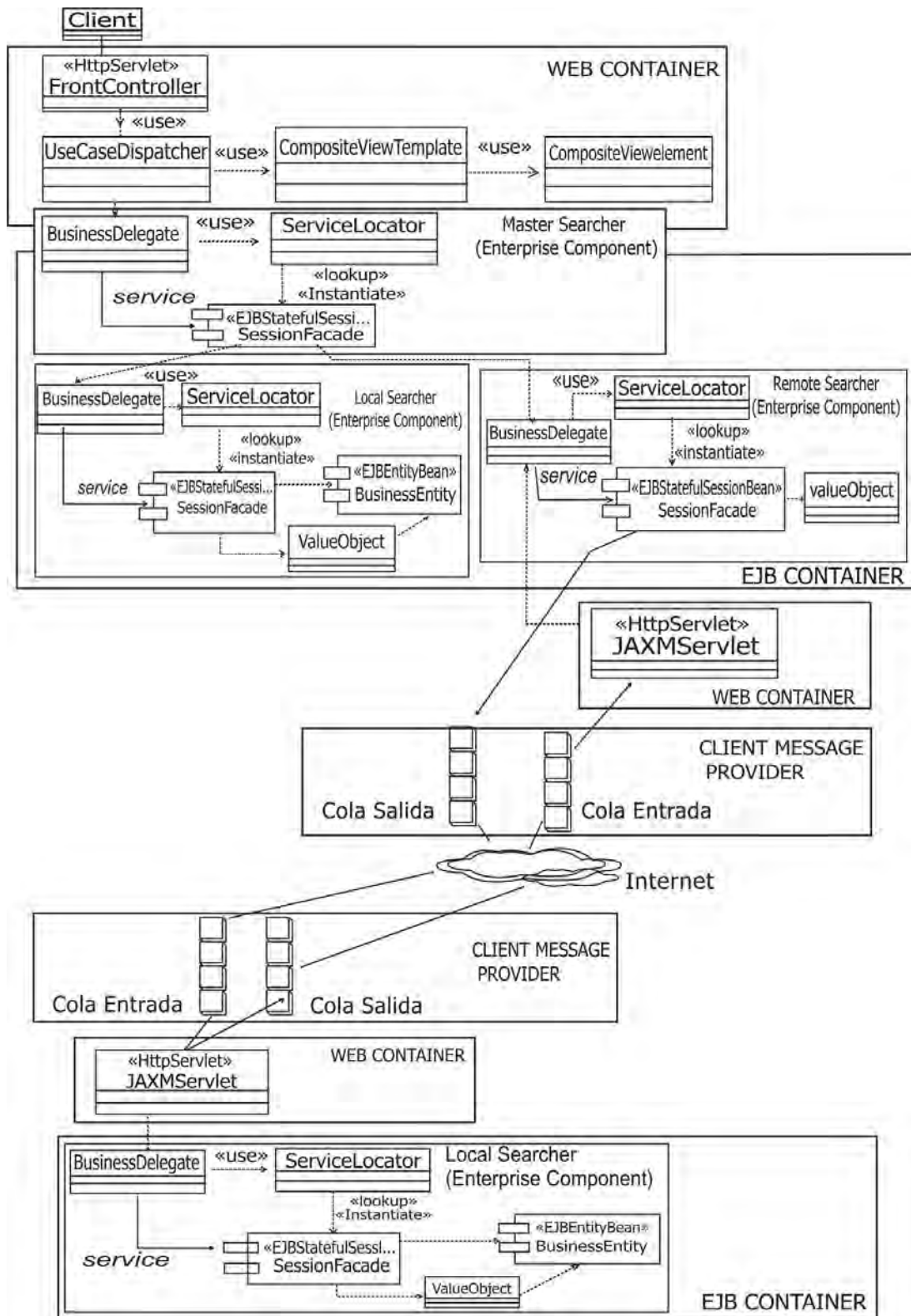


Figura 3.33: Diagrama de interacción entre los componentes J2EE

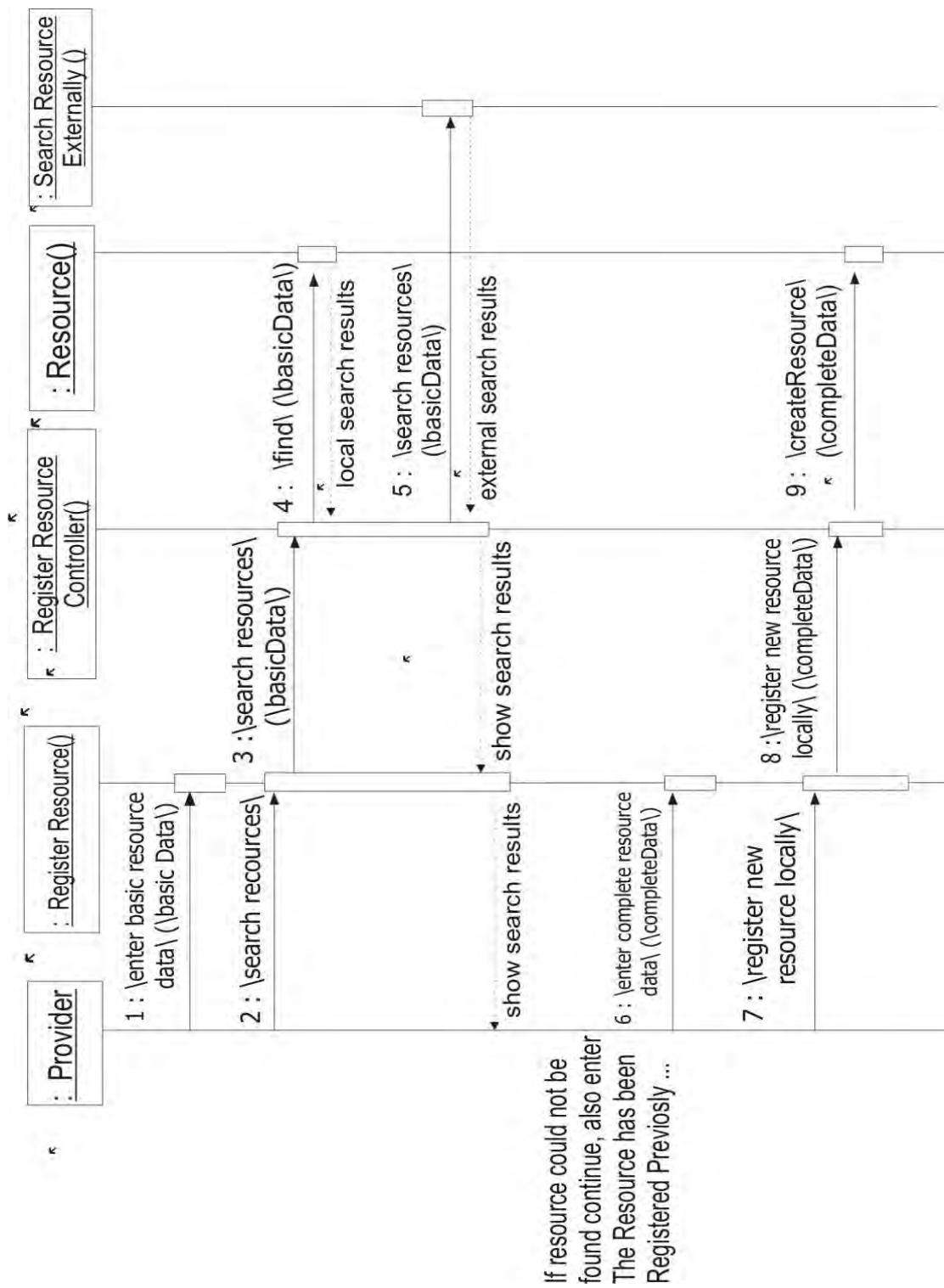


Figura 3.34: Flujo normal del caso de uso: Registrar Recursos del sistema de biblioteca

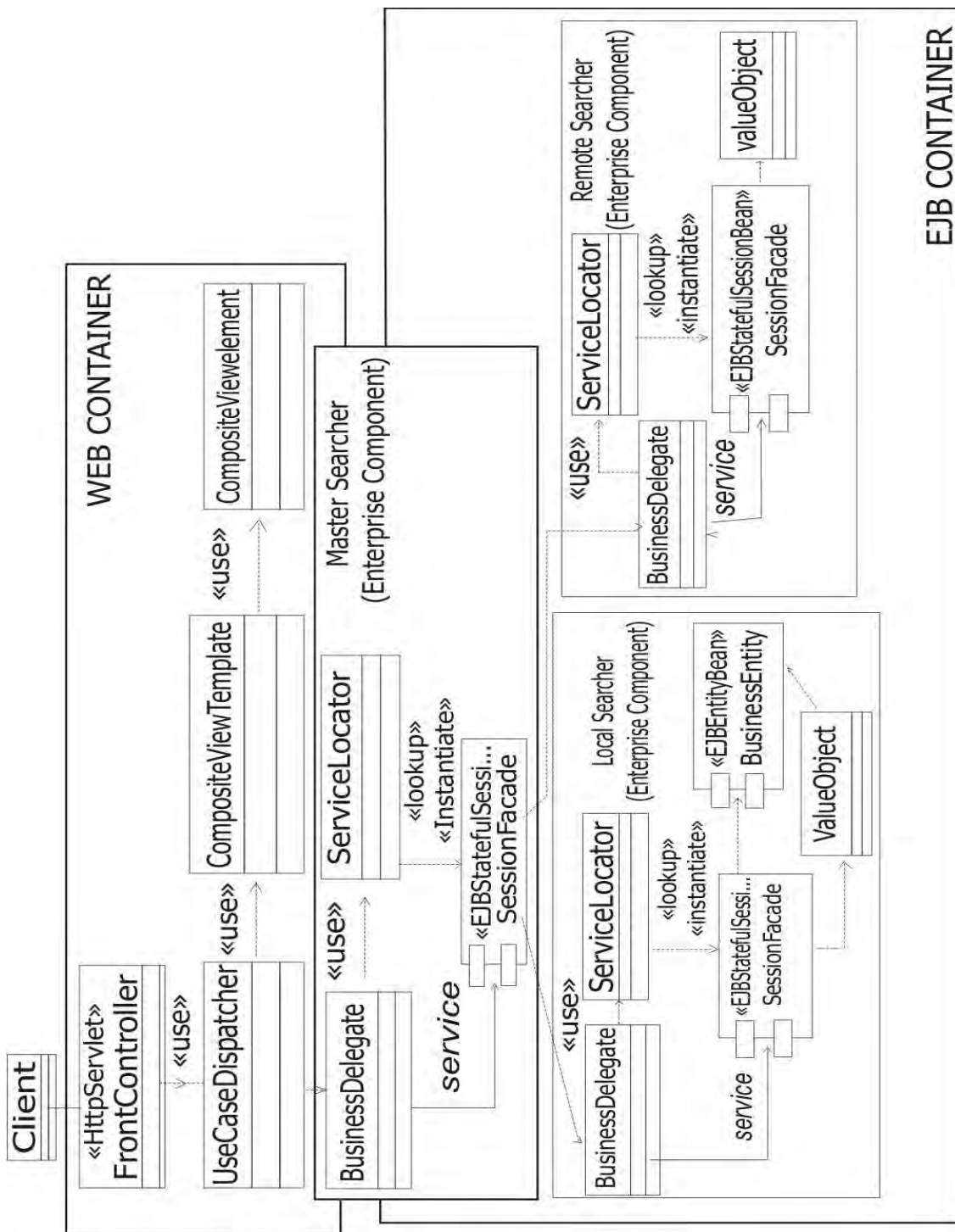


Figura 3.35: Detalle del bloque del nodo cliente en el diagrama de interacción entre los componentes de la Figura 3.25

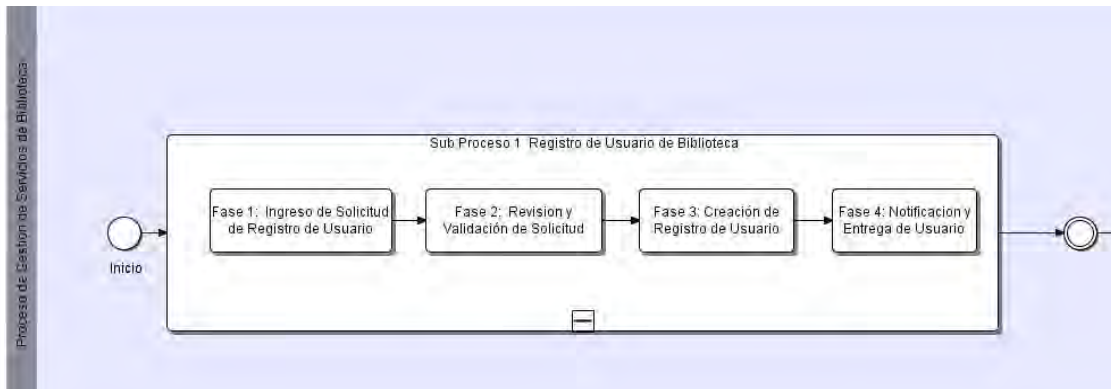


Figura 3.36: Detalle de descomposición de las fases 1, 2, 3 y 4 del proceso de gestión de servicios de biblioteca

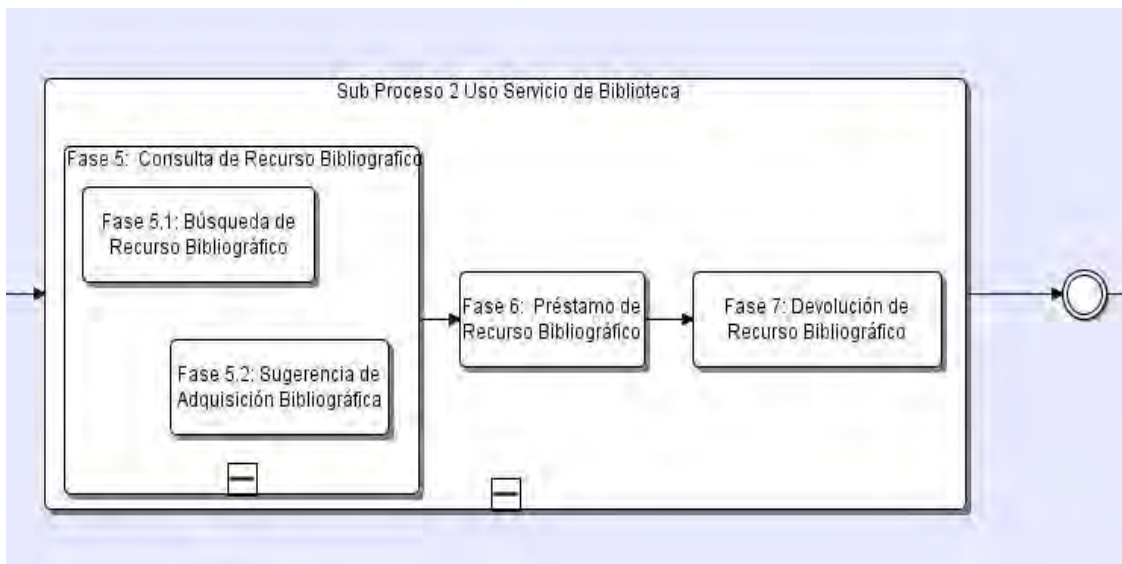


Figura 3.37: Detalle de descomposición de las fases 5, 6 y 7 del proceso de gestión de servicios de biblioteca

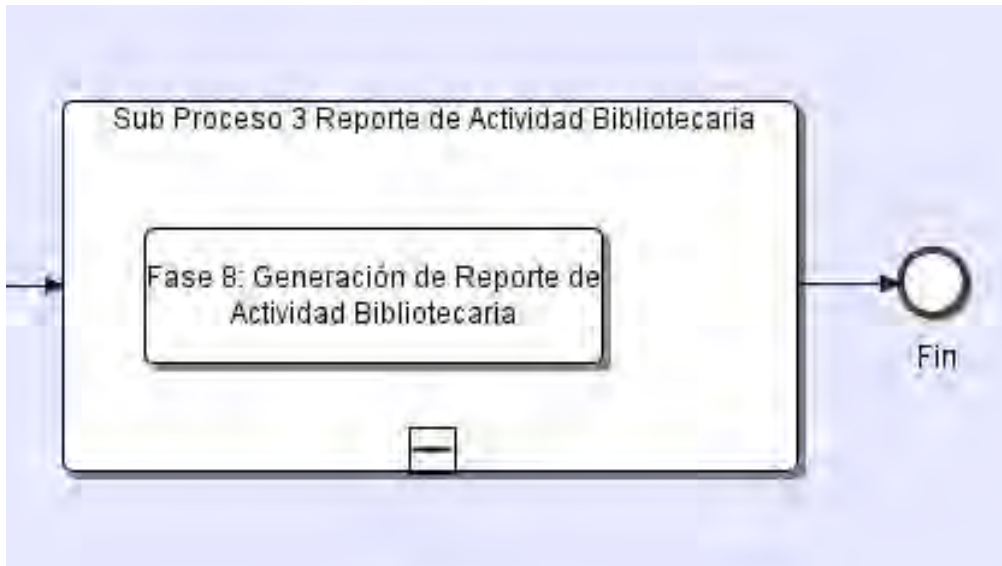


Figura 3.38: Detalle de descomposición de la fase 8 del proceso de gestión de servicios de biblioteca

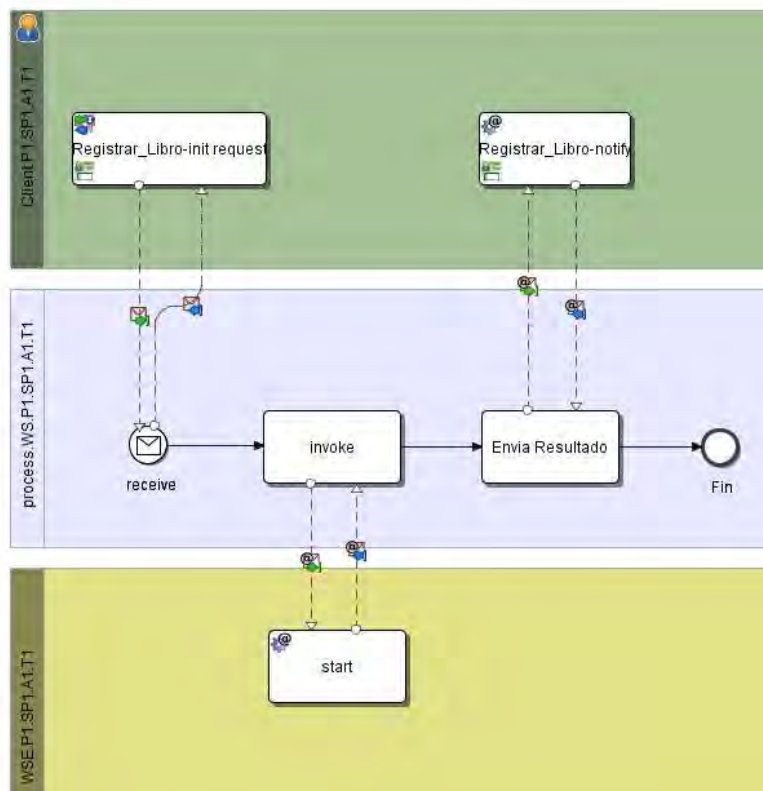


Figura 3.39: Modelado BPMN que invoca el web service de registro de recurso bibliográfico con su instancia libro

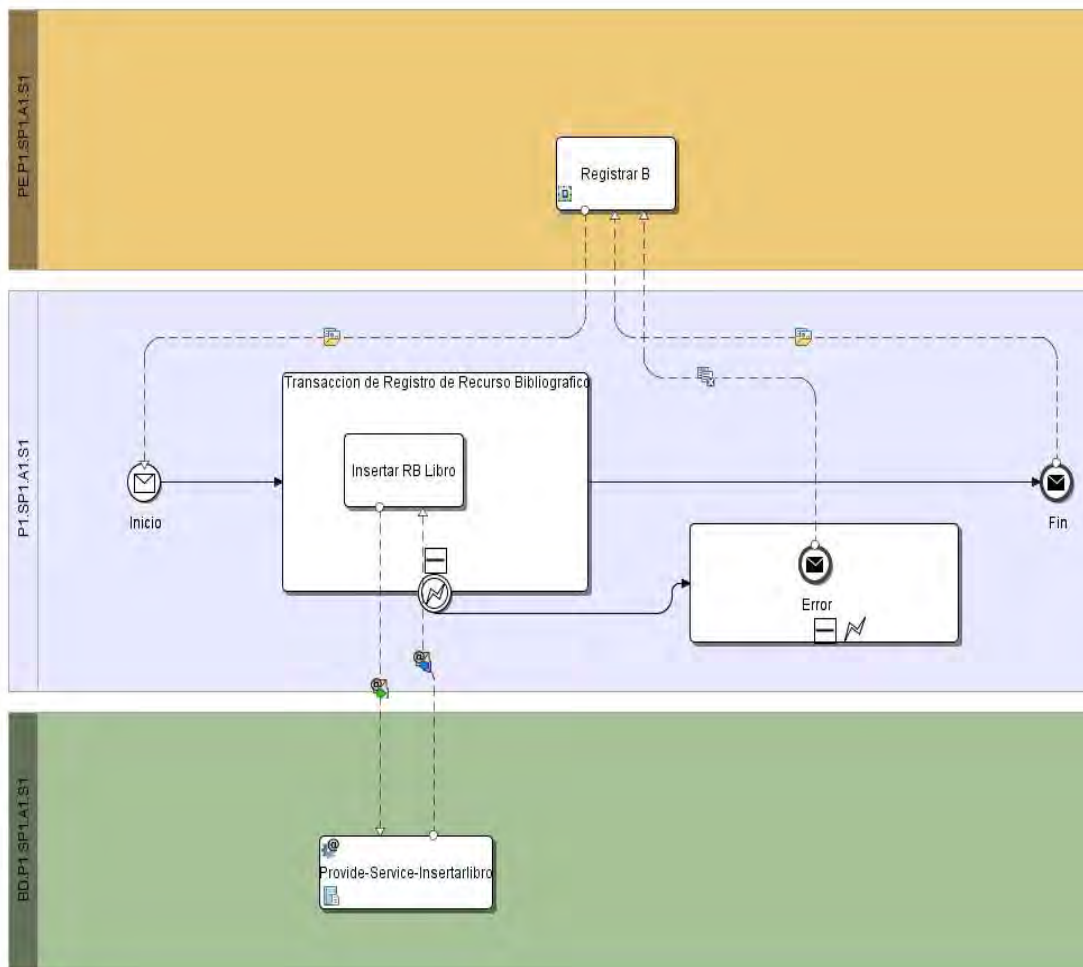


Figura 3.40: Detalle interno del manejo de excepciones BPMN para inserción del recurso bibliográfico libro

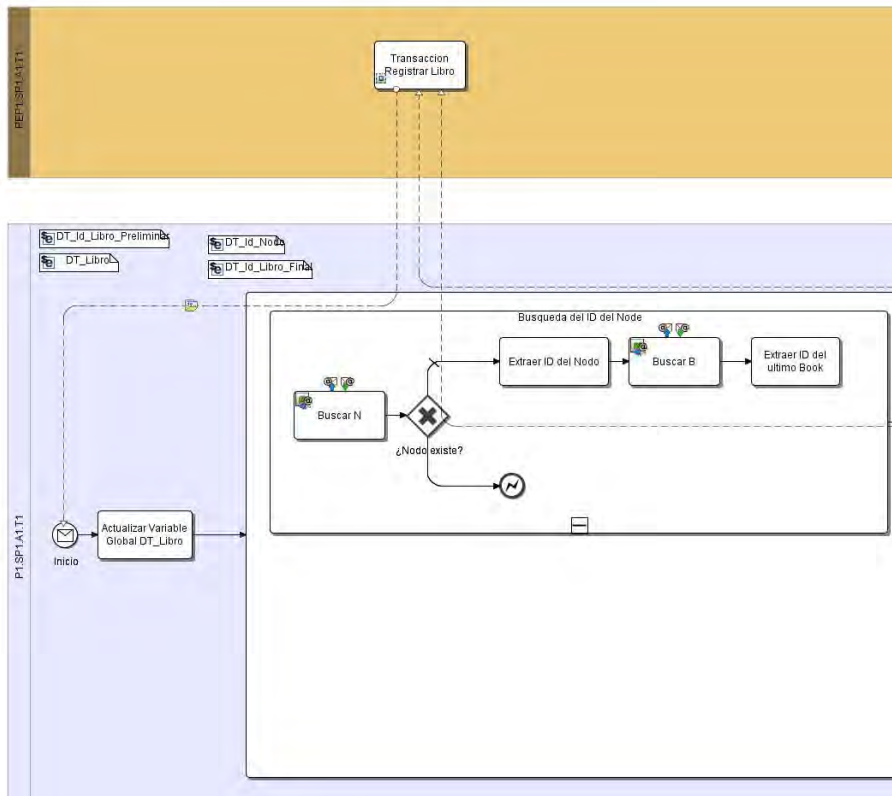


Figura 3.41: Detalle del código BPMN que implementa gráficamente la transacción del registro del recurso bibliográfico libro (Parte 1/2)

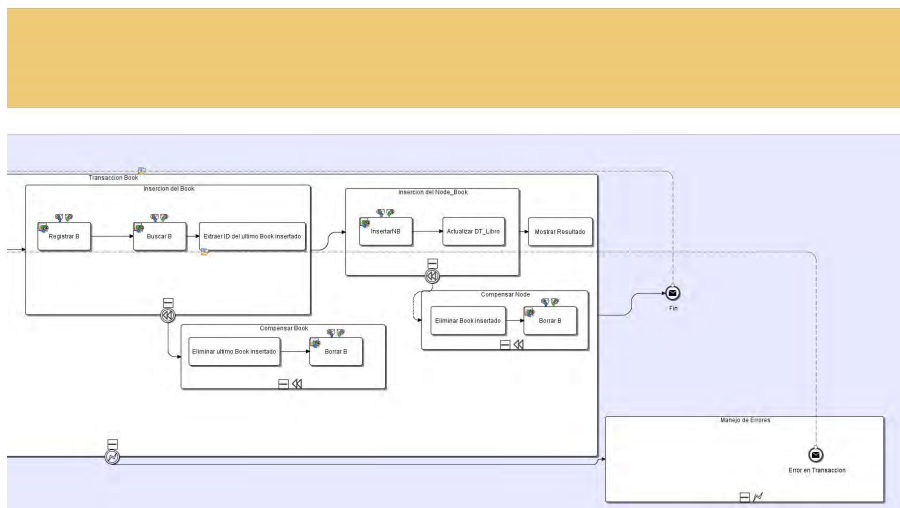


Figura 3.42: Detalle del código BPMN que implementa gráficamente la transacción del registro del recurso bibliográfico libro (Parte 2/2)

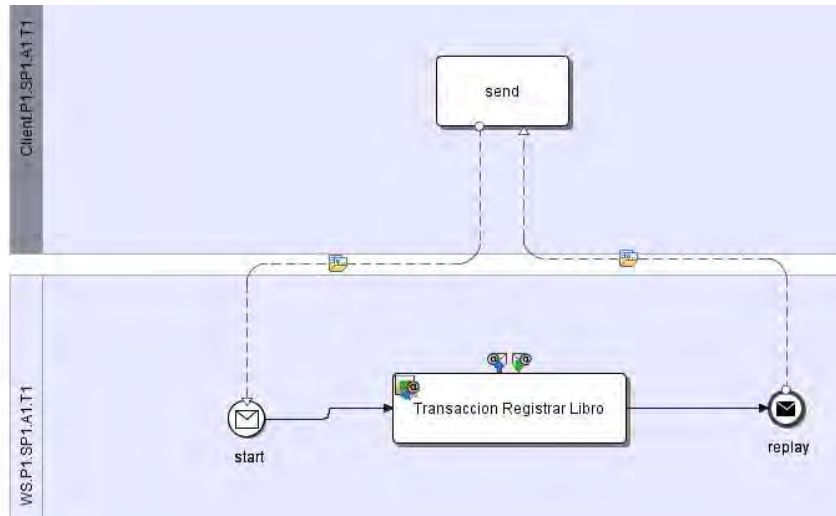


Figura 3.43: Detalle del bloque BPMN que implementa una subrutina que llama a un web service en forma gráfica

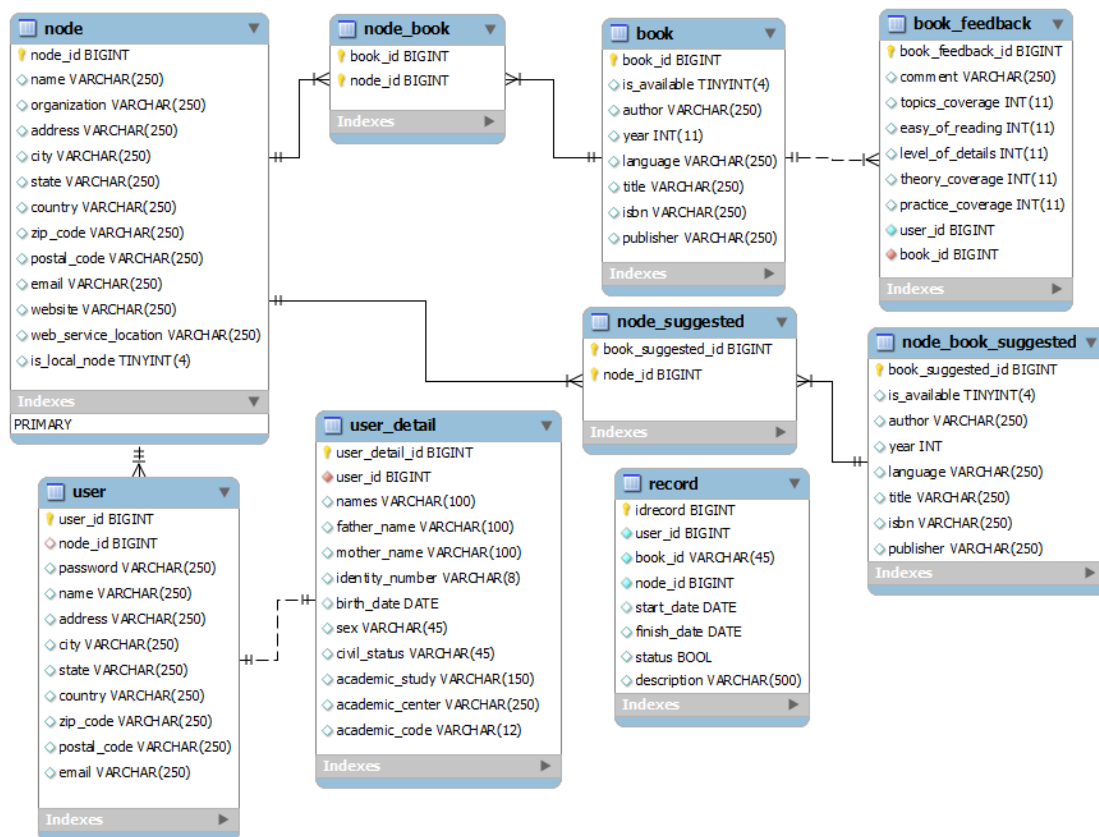


Figura 3.44: Detalle del modelo de base de datos utilizado en el proyecto BPMN de la Biblioteca Digital Distribuida

Capítulo 4

Revisión de la Solución, Pruebas del Prototipo y Resultados

4.1. Revisión de la solución

A continuación, se revisa por capas el diseño del sistema de biblioteca:

Capa de Presentación

La interfaz principal del sistema es a través de navegadores de internet que soportan el estándar Web 2.0. Se utiliza la plataforma BPMS Intalio, el cual ayuda a crear un ambiente extensible de desarrollo para aplicaciones web. El modelo que provee el BPM de Intalio, está basado en AJAX de TIBCO para comunicar el Workflow TEMPO con los usuarios finales. A diferencia de una aplicación convencional, cuando se implementa un proceso usando BPM como parte del proceso mismo, se solapan los casos de uso modelados en el enfoque convencional. En algunos casos las interfaces AJAX deben comunicarse con la capa de persistencia, para acceder a la base de datos que contiene la información necesaria para activar o reactivar cierto estado. En caso que se necesite mejorar la interface gráfica, es posible conectar el proceso modelado a través de un portal de contenido, tal como la tecnología Liferay, pero en la mayoría de los casos es suficiente con la interface AJAX TIBCO; y si se requiere agregar mayores funcionalidades es posible incorporar AJAX DWR, JQUERY, entre otras, como complemento a las librerías existentes.

La fortaleza de la herramienta BPM-Intalio es el utilitario de mapeo XML, que permite enlazar el DOM de una interface web con variables internas AJAX-TIBCO, las que a su vez se enlazan con llamadas a los web services, para que se conecten con la base de datos en caso se requiera vía código Javascript, autogenerado por la misma herramienta (ver Figura 4.1).

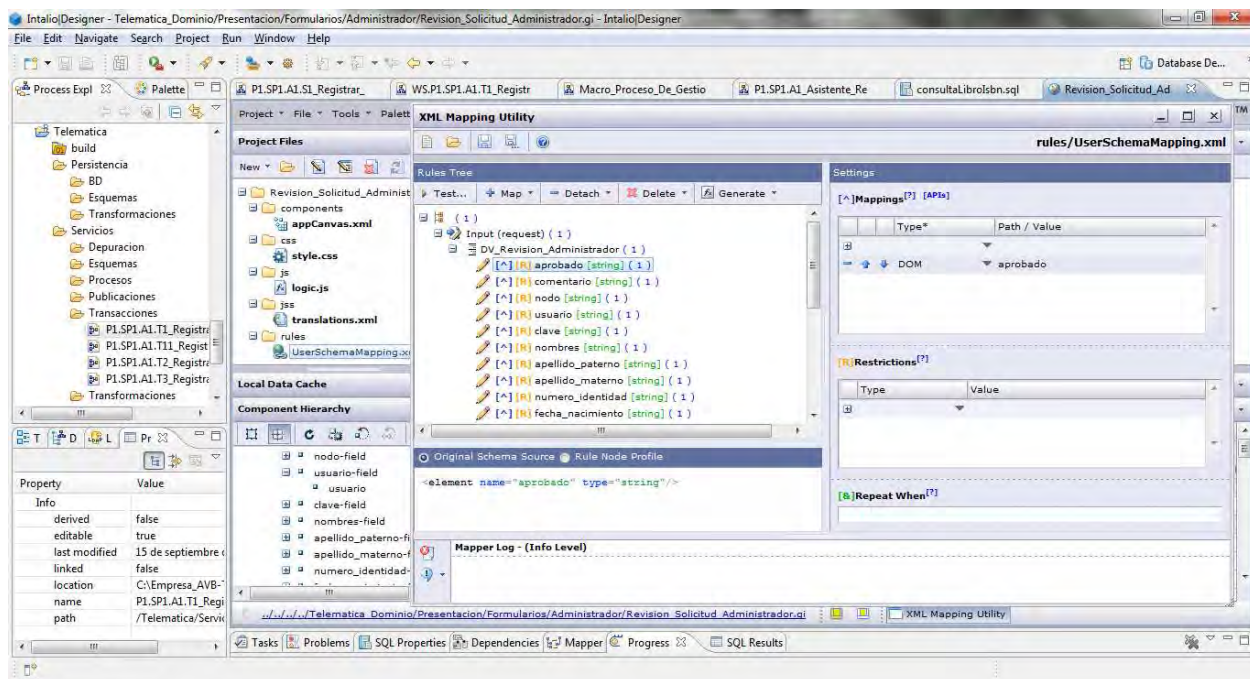


Figura 4.1: Utilitario de Mapeo XML de la interface de Diseño BPM Intalio

Capa de Dominio

En esta capa se implementa a detalle el proceso modelado, en donde se fijan las bifurcaciones por cada fase del modelado del proceso; las actividades que son modelados en BPMN deben ser realizadas por la aplicación para dar soporte a uno o varios casos de uso, generando internamente código BPEL. Se vio conveniente utilizar comunicaciones asíncronas, en escenarios donde se requiere postergar la respuesta por un actor determinado. Asimismo, se han utilizado plantillas BPMN para conectar bloques de códigos BPMN, y en otros casos una interface externa de Web Services entre flujos de sub procesos para comunicar bloques que están contenidos en diferentes proyectos.

En las Figuras 4.2 y 4.3 se muestran un escenario base de comunicación asíncrona, utilizado como referencia en el proyecto de biblioteca digital, para pasar mensajes entre los actores en donde la respuesta podría tomar un tiempo prolongado. La técnica BPMN de comunicación asíncrona necesariamente requiere el uso de plantillas y correlaciones; el primero se refiere a etiquetar un bloque de código BPMN y el segundo se refiere a definir una variable que sirva como patrón de comparación y, en caso coincida, recién se atiende dicho flujo de información.

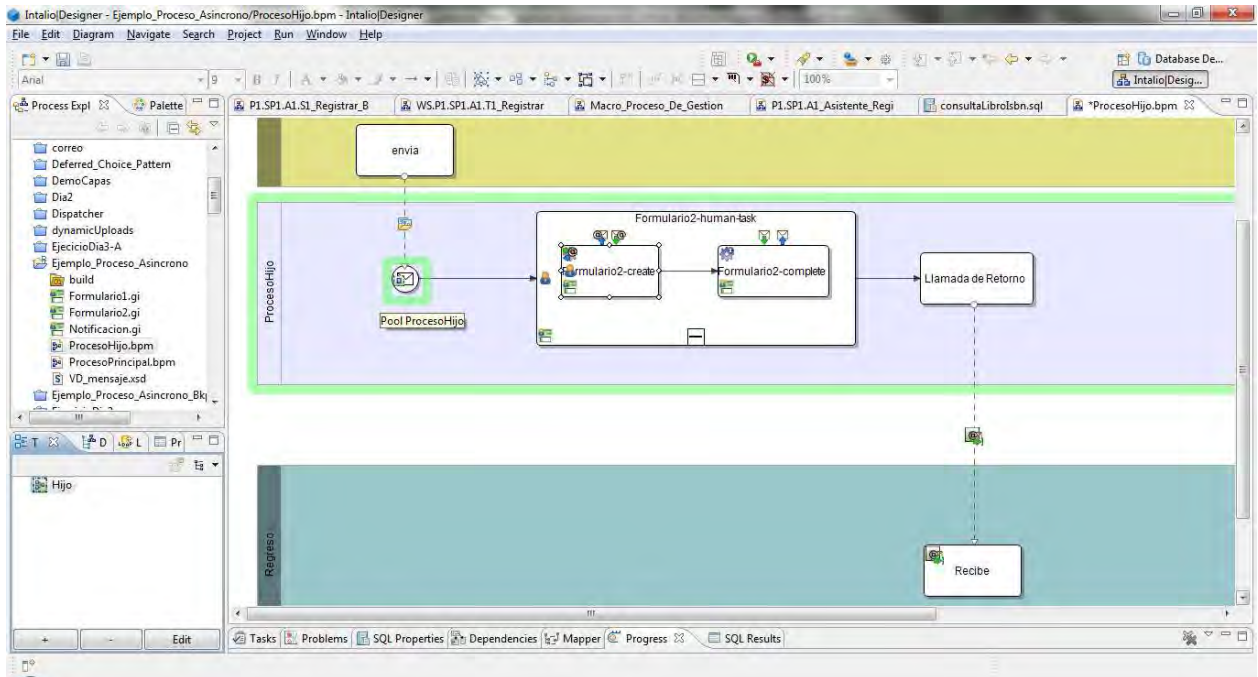


Figura 4.2: Diagrama BPMN base de comunicación asíncrona utilizando la Vista de Correlación y Plantillas (1/2)

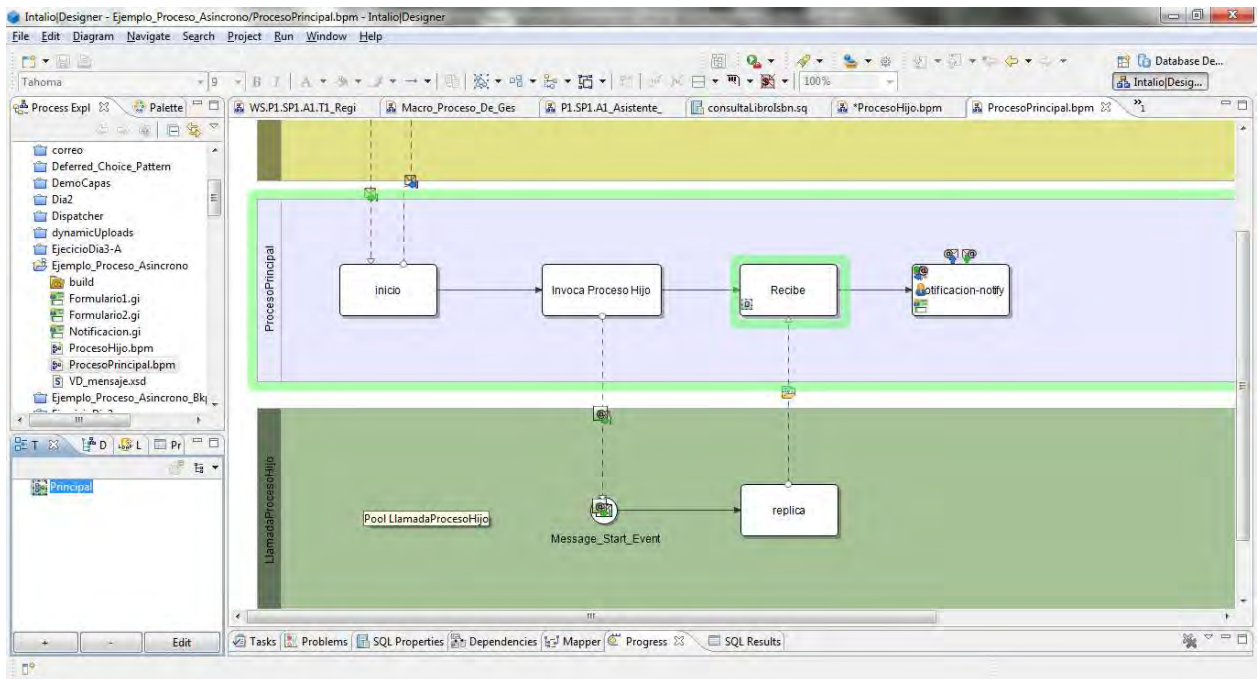


Figura 4.3: Diagrama BPMN base de comunicación asíncrona utilizando la Vista de Correlación y Plantillas (2/2)

Capa de Persistencia

Aquí tienen lugar la persistencia de los datos y el acceso hacia sistema externos. La persistencia de datos se realiza mediante la interfaz JDBC dentro del Diseñador BPM Intalio, que internamente genera conectores WSDL. Para poder representar los datos relacionales como objetos, se crean plantillas que internamente llaman a las interfaces web services creadas. La integración también comprende la comunicación con sistemas externos como son los otros nodos de la Biblioteca Digital Distribuida. Para ello simplemente se referencia a la dirección de la base de datos que se desea comunicar o en su defecto al web service que se desea conectar externamente en la aplicación.

En algunos casos, las sentencias SQL se crean directamente en el entorno de diseño BPM Intalio, en otros, es más práctico invocar a procedimientos almacenados guardados en la misma base de datos, de modo que los detalles de transacciones complejas queden desacoplados de la implementación BPM.

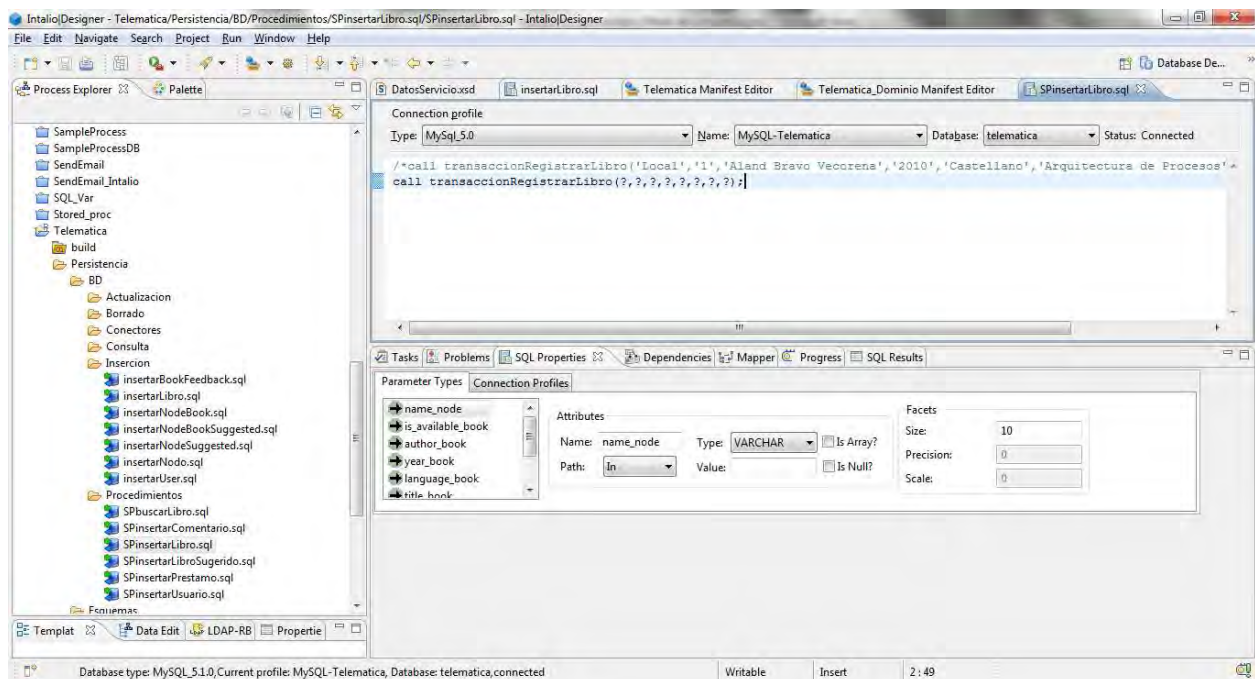


Figura 4.4: Escenario de invocación a un procedimiento almacenado externo

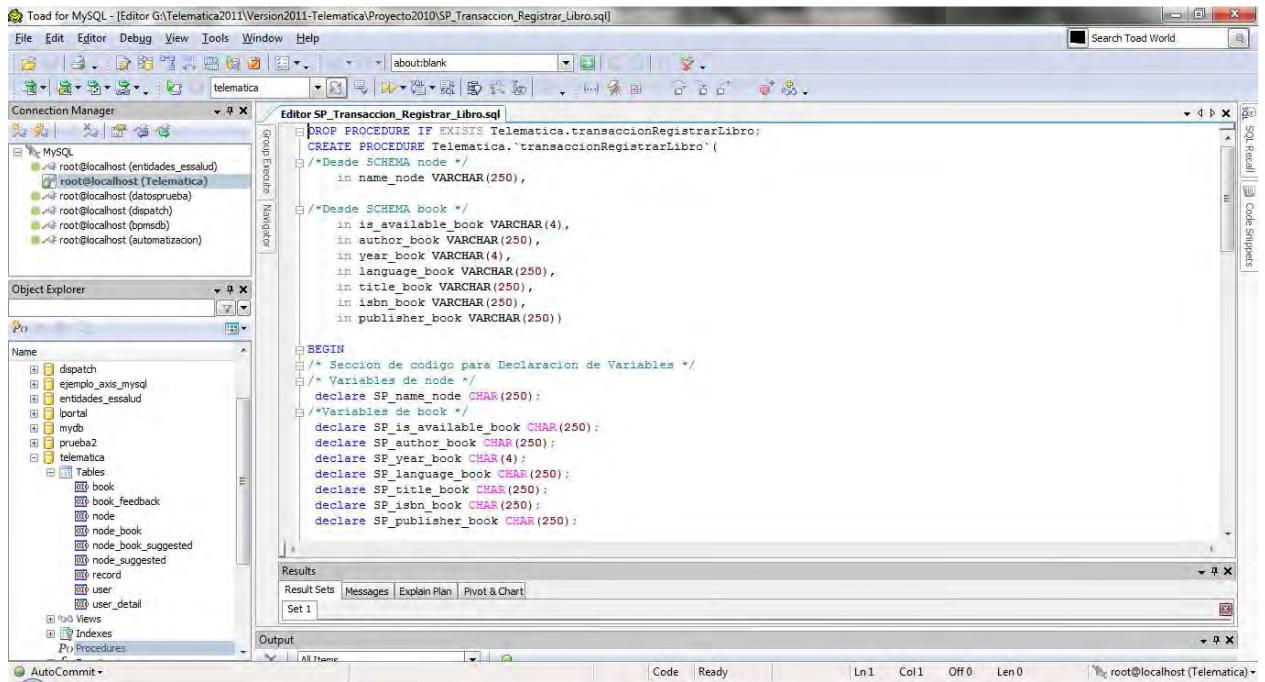


Figura 4.5: Detalle de implementación del procedimiento almacenado externo

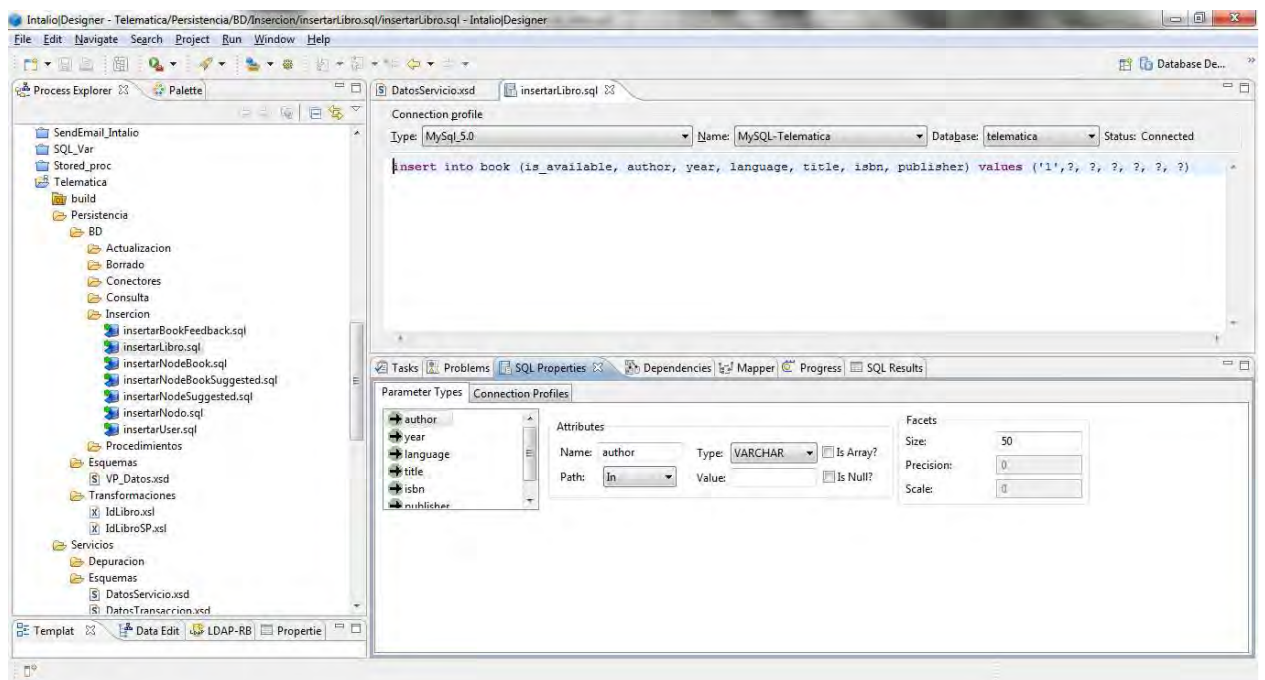


Figura 4.6: Escenario de invocación a un SQL creado directamente en la herramienta de diseño BPM Intalio

Capa de Servicio

Aquí tienen lugar el cómo se hace; es decir el procedimiento interno BPMN que permite procesar el flujo de información del proceso mismo. En la capa de servicio se realiza el enlace entre la Capa de Persistencia de Datos con la Capa de Dominio. Existen varios niveles de abstracción en esta capa, desde servicios para los macro procesos, hasta servicios a nivel de detalle.

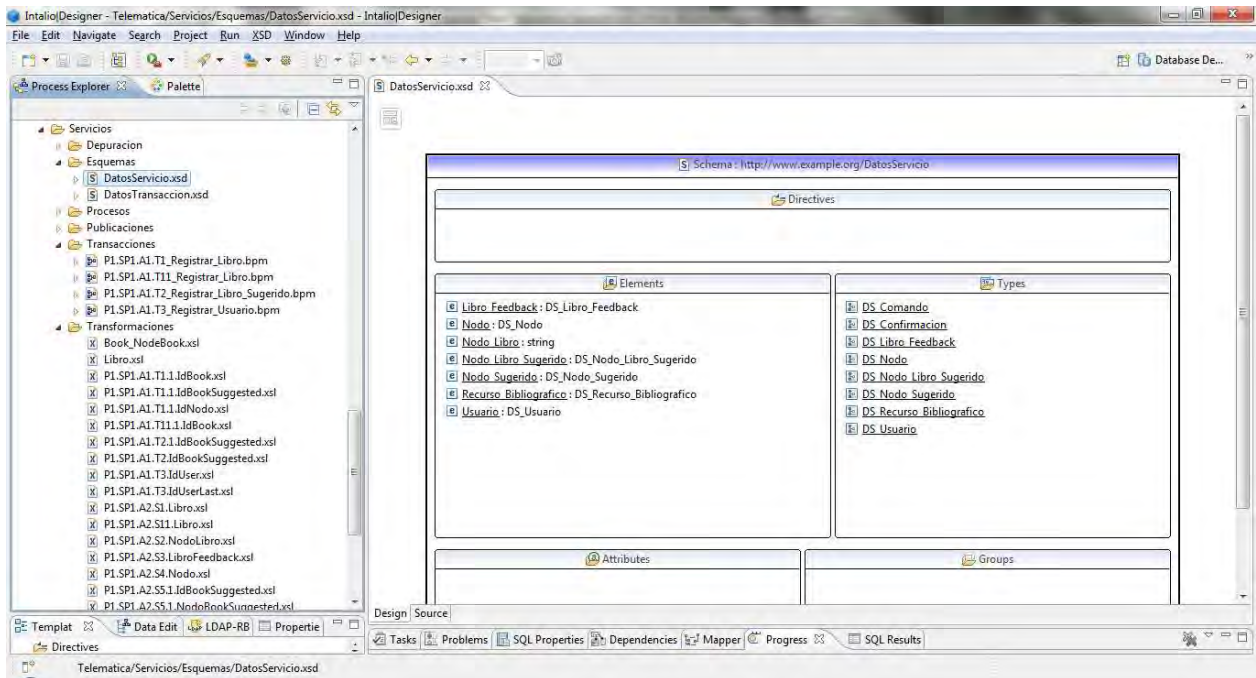


Figura 4.7: Detalle del esquema de datos XSD utilizado en la capa de servicio

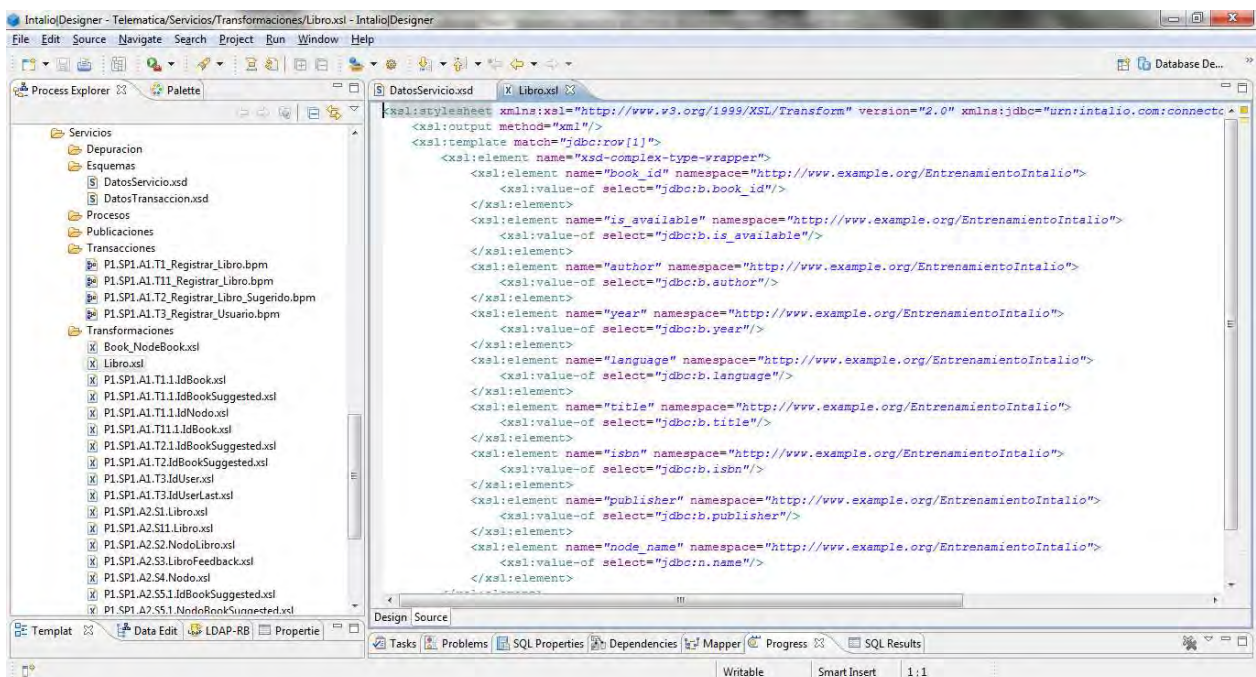


Figura 4.8: Detalle del XSLT utilizado en la capa de servicio

En las Figuras 4.7 y 4.8 se muestran detalles de los elementos XSD y XSLT que se utilizan en conjunto con XML para el manejo de la estructura de los datos a nivel de la Capa de Servicio.

4.2. Prueba Piloto

En la prueba piloto, se configuraron varios nodos o servidores ubicados en el laboratorio del área de ingeniería de software de la Sección de Posgrado y Segunda Especialización FIEE. Es importante señalar que se usó la versión código abierto de la plataforma BPM Intalio sobre JBOSS y TOMCAT.

Ya que la biblioteca digital distribuida es un servicio web, esto es, una unidad de aplicación lógica con objetos y métodos que normalmente es accesible vía HTTP y SOAP; en la prueba piloto, se implementaron varios casos de uso tácitamente a través del proceso automatizado: iniciar sesión, registrar recursos, registrar biblioteca, y registrar usuario; publicando los métodos para poder ser accedidos por web.

La esencia de los Web Services recae en XML (lenguaje de marcas extensibles) en donde todas las operaciones (métodos en analogía con Java) de un servicio web se expresan a través de un archivo WSDL (lenguaje de descripción de servicios web).

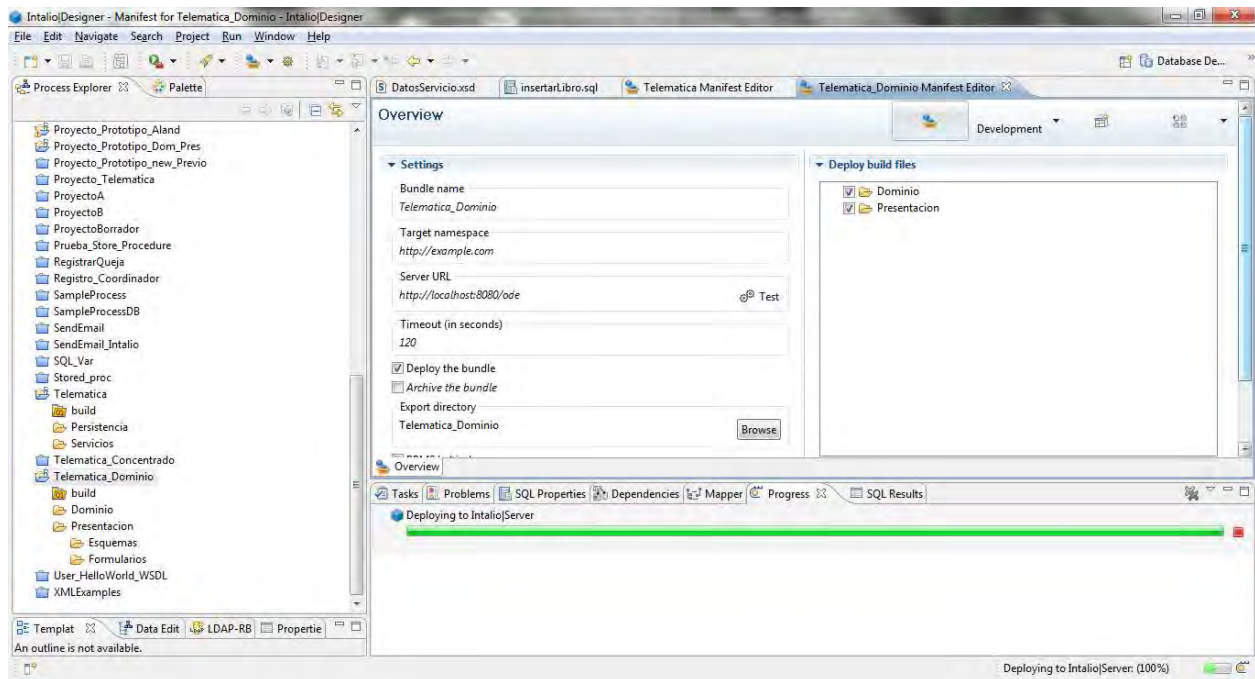


Figura 4.9: Configuración de Despliegue de las Capas de Dominio y Persistencia del Proyecto de Biblioteca Digital Distribuida

El archivo WSDL es usado para acceder y llamar operaciones en servicios web. Axis 1.0 es una de varias aplicaciones de herramientas disponibles para Java de la implementación de servicios web, y probablemente es el más avanzado en términos de características y desempeño. Axis usa el WSDD (descriptor de despliegue de servicio web), el cual permite flexibilidad y características avanzadas. Axis 1.0 permite controlar el tipo de proveedor de los manejadores que son usados para exponer componentes como servicios web.

Hoy en día es frecuente integrar BPM con EJB empresariales y servicios web para proporcionar interoperabilidad entre BPEL, J2EE y otras tecnologías, como .NET, PHP, Perl, etc. Por ejemplo, si se ha escrito un EJB en el back end, y se está usando ASP.NET para desarrollar en el lado front end del cliente, entonces se puede usar servicios web services dentro de la aplicación BPMN para conectar las dos plataformas.

4.2.1. Preparación de la prueba piloto

En el proyecto de Tesis, se usaron las herramientas siguientes para la elaboración del prototipo propuesto:

- JBoss o fuente abierta que alberga al J2EE. Este es un servidor de los objetos de negocio de la aplicación. Para mayores referencias se puede consultar a <http://www.jboss.org>.
- Apache Axis 1.0 para programar SOAP. Se puede consultar mayores detalles en: <http://xml.apache.org/axis>.
- AJAX de TIBCO, que es un framework MVC (patrón modelo vista controlador) para desarrollar aplicaciones web en la capa de presentación.
- SQL como lenguaje de consulta de datos, conjuntamente con XSD y XSLT.
- MySQL como motor para la base de datos.

- Intalio ODE Server como contenedor de procesos que interpreta código BPMN para mapearlos a BPEL.

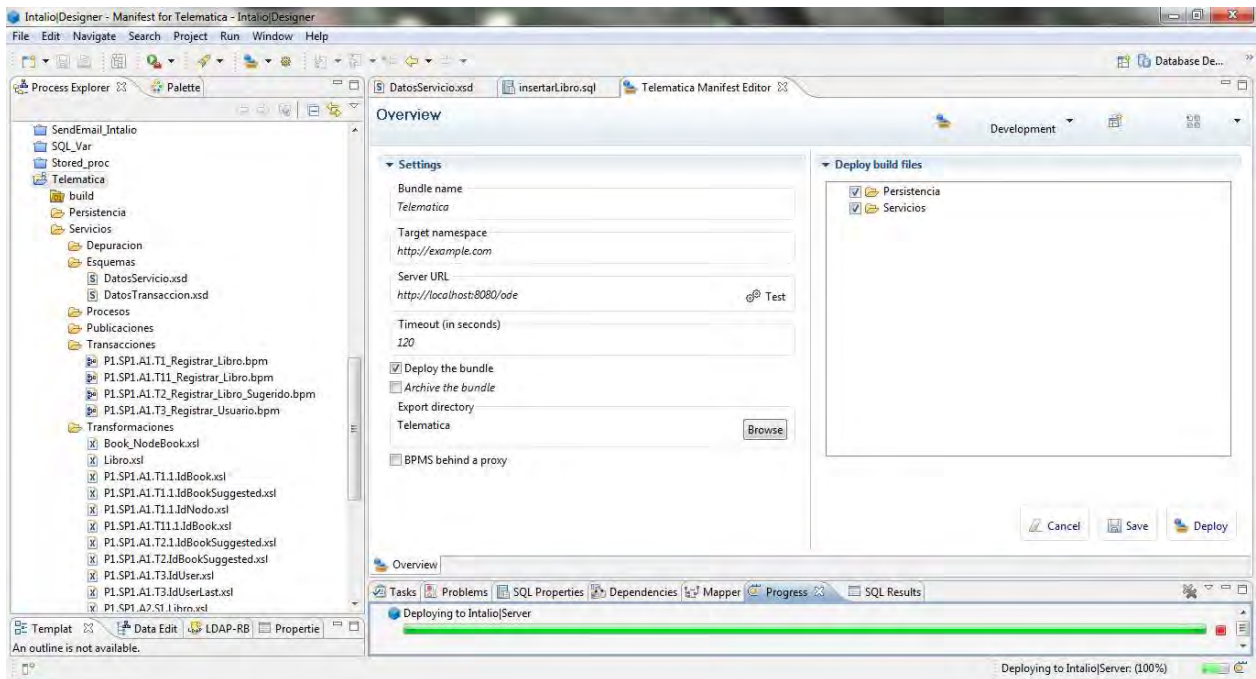


Figura 4.10: Configuración previa de despliegue de las Capas de Persistencia y Servicio del Proyecto de Biblioteca Digital Distribuida

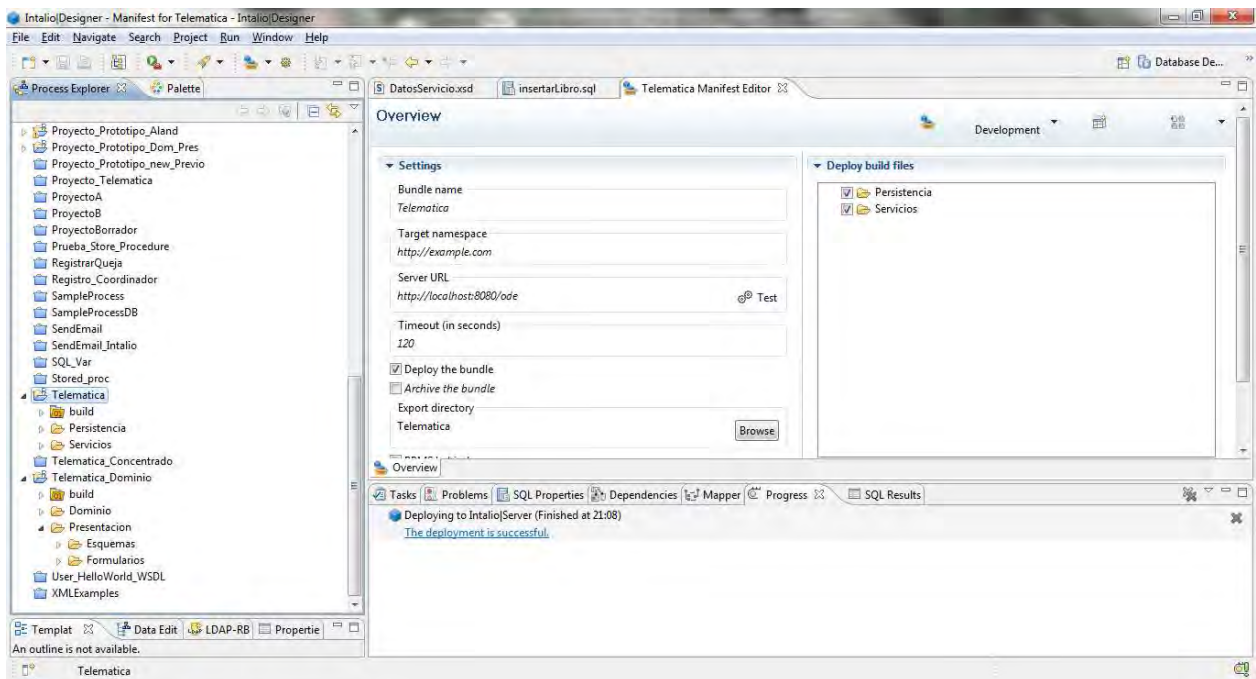


Figura 4.11: Configuración exitosa de despliegue de las Capas de Persistencia y Servicio del Proyecto de Biblioteca Digital Distribuida

4.2.2. Desarrollo de la prueba piloto

Durante el desarrollo de la prueba piloto, se dio énfasis tanto en la parte del prototipo de Biblioteca Digital Distribuida Dexters, así como en la validación de la Arquitectura Orientada a Regulaciones, propuesto para optimizar el proyecto de biblioteca distribuida como parte de trabajos futuros generados como consecuencia del presente proyecto, pero que se detallan como avance en la tesis.

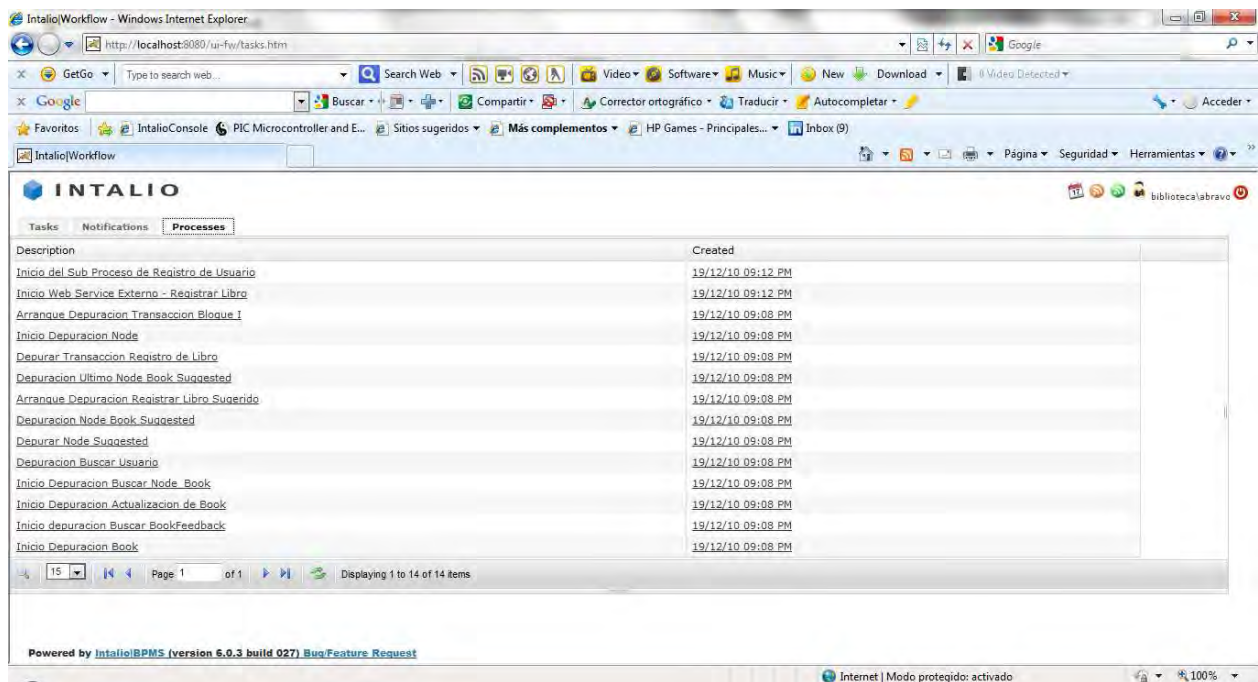
4.2.3. Análisis de resultados de la prueba piloto

Desde el punto de vista de las herramientas tecnológicas empleadas, se ha podido notar lo siguiente:

- Los productos basados en web services usualmente lidian con una arquitectura específica del fabricante. Por ejemplo el servidor de procesos BPM Intalio debe correr sobre JBoss, Tomcat o Gerónimo. De ahí la ventaja de usar ROA-BPM, ya que nos da un marco de referencia unificada para gestionar el proyecto como un todo en su conjunto.
- Los patrones de diseño y de procesos con web services pueden acumularse como parte de la experiencia de despliegue. En ese sentido es útil mantener un catálogo de web services personalizados con el giro de modelo empresarial para posteriores usos.
- Un estructurado framework de la arquitectura web services no solamente provee una metodología precisa para definir componentes web services, sino que ayuda a seleccionar herramientas adecuadas para implementar calidad de servicio durante el desarrollo y despliegue del ciclo de vida.
- La tendencia del mercado: es que algunos vendedores de soluciones automatizadas utilicen estándares al momento de diseñar sus productos middleware legados como conectores de integración a los web services.
- El alcance de la seguridad en web services es un punto a desarrollar, en esto las firmas digitales obedecen al mecanismo de no repudio en los cuales existen otros aspectos de seguridad a ser considerados como complemento.
- El envío de data XML en texto plano es inseguro si no se realiza una previa

encriptación o en todo caso utilizar un ESB que maneje los temas de seguridad en forma separada.

- En el contexto de política tecnológica nacional aún no se percibe madurez, quedando pendientes los temas de leyes que establezcan políticas de intercambio de datos entre las empresas sin repudio entre ellas, por ejemplo a través de certificados digitales (tales como PKI X.509v3 o Kerberos) sobre un escenario de un Bus de Servicio Empresarial ESB, esto permitiría a los usuarios comprobar que las transacciones han sido efectuadas con credenciales válidas establecidas por el gobierno, en calidad de peritos virtuales.



Description	Created
Inicio del Sub Proceso de Registro de Usuario	19/12/10 09:12 PM
Inicio Web Service Externo - Registrar Libro	19/12/10 09:12 PM
Arranque Depuracion Transaccion Bloque I	19/12/10 09:08 PM
Inicio Depuracion Node	19/12/10 09:08 PM
Depurar Transaccion Registro de Libro	19/12/10 09:08 PM
Depuracion Ultimo Node Book Suggested	19/12/10 09:08 PM
Arranque Depuracion Registrar Libro Sugerido	19/12/10 09:08 PM
Depuracion Node Book Suggested	19/12/10 09:08 PM
Depurar Node Suggested	19/12/10 09:08 PM
Depuracion Buscar Usuario	19/12/10 09:08 PM
Inicio Depuracion Buscar Node Book	19/12/10 09:08 PM
Inicio Depuracion Actualizacion de Book	19/12/10 09:08 PM
Inicio depuracion Buscar BookFeedback	19/12/10 09:08 PM
Inicio Depuracion Book	19/12/10 09:08 PM

Figura 4.12: Listado de procesos implementados para el despliegue del proyecto de Biblioteca Digital Distribuida

Registrar Recurso: El proveedor registra su recurso en el nodo del prototipo de biblioteca de su preferencia, probablemente en la biblioteca de su organización. La metadata (información que describe al recurso) y el recurso mismo residen inicialmente en este nodo. La metadata contiene información sobre:

- La(s) localización(es) de(l) recurso(s)
- Información descriptiva propia del tipo de recurso (libro, audio, video, etc.)

- Rating
- Comentarios
- Propietario

Para registrar un recurso, primero se emite una búsqueda general para determinar si el recurso ya ha sido registrado en otro nodo anteriormente (cada nodo que se ha invocado y que efectivamente disponga del recurso, registrará qué nodo(s) poseen el proxy de su metadata). Si el recurso ya existe en otro nodo, la metadata se descarga al nodo local, con lo cual se reduce la duplicación de datos y se ahorra el tiempo en el registro del recurso. La metadata del nuevo recurso del nodo local reside permanentemente en el nodo local, en ella también se registran los otros nodos que tienen el recurso. Una vez completado el registro en el nodo local, se emite un mensaje de actualización a todos los otros nodos que posean el recurso; de manera que registre el nuevo nodo (el local) como otro poseedor más del recurso. El hecho de mantener un registro de los otros nodos que cuentan con un determinado recurso es conveniente puesto que representa un índice que reduciría drásticamente los tiempos de búsqueda de recursos.

Buscar Recurso: El consumidor realiza la búsqueda del recurso que desea. Para la búsqueda se define un ámbito o amplitud geográfica de la misma, la cual puede ser local (en el mismo nodo) o remota (en cualquier otro nodo).

Si el recurso está localizado en un nodo remoto, se descarga la metadata del recurso hacia el nodo local almacenándose en una memoria caché que permitirá que las siguientes búsquedas realicen sólo invocaciones locales.

Si el recurso inicialmente disponible en otro nodo se hace disponible en el nodo local, entonces la metadata obtenida se almacena como proxy en el nodo local, el mismo que será reemplazado de acuerdo al algoritmo LRU.

El nodo remoto que proporciona metadata de un determinado recurso, registra los nodos que actualmente poseen el proxy de la metadata así como sus tiempos de vida.

Agregar Comentario a Recurso: Todo comentario registrado en un nodo, reside permanentemente en él y sólo un proxy del mismo se almacena temporalmente en los

nodos que solicitaron información respecto al recurso.

Es importante recalcar que el acceso a los recursos es un tema que escapa a los objetivos del proyecto. Inicialmente se implementará solo lo referente al registro y búsqueda de recursos, así como la obtención de su metadata. Quedando pendiente lo referente al acceso de los recursos.

Los recursos pueden ser de dos tipos dependiendo de su naturaleza y de acuerdo a su disponibilidad: físico y electrónico.

- Los recursos físicos pueden obtenerse directamente de la organización o persona que los posea o a través de algún servicio de entregas (delivery).
- Los recursos electrónicos pueden ser accedidos directamente del nodo DexterLib que los posea. Si el contenido es libremente accesible, se inicia la descarga del recurso del nodo que lo posee hacia el nodo que lo invocó.

4.2.4. Toma de decisiones

La nueva estructura del sistema educativo peruano, se sustenta en un cambio del paradigma educativo por el que los alumnos pasan a ser el centro del proceso de aprendizaje y los maestros ya no son meros transmisores de información. Como resultado de este cambio de paradigma surge la necesidad de formar maestros no sólo capaces de ayudar a sus alumnos en la construcción de aprendizajes, sino también en su desarrollo personal e interpersonal. Estas preocupaciones parten de la idea de que los maestros en el futuro, aquellos que puedan responder adecuadamente a las necesidades de las personas en el mundo cambiante en que vivimos, serán los que puedan asumir sus propios procesos de aprendizaje y puedan embarcarse en un camino de constante enriquecimiento y reformulación de lo que hacen, todo esto en concordancia con las necesidades de sus alumnos.

El prototipo propuesto, demuestra que es factible implementar una plataforma a escala nacional, que permita a los maestros que vayan más allá de establecer una relación instructiva con sus alumnos para desencadenar procesos comunicativos que permitan un

aprendizaje autónomo y creativo. Se les proporciona una plataforma para que los maestros orienten a sus alumnos en la construcción del conocimiento y tengan en cuenta que la dimensión afectiva está presente y juega un rol fundamental en este proceso. En todo acto educativo, se planifique o no, se desee o no, el maestro interviene, positiva o negativamente en la formación de cada alumno. Basta su presencia para que esto ocurra. El acto educativo implica necesariamente una función tutorial con una plataforma tecnológica altamente integradora de respaldo.

Según la nueva estructura del sistema educativo peruano, la función de los maestros ha cambiado o debería cambiar radicalmente para lograr que el sistema educativo pueda responder adecuadamente a las necesidades de las personas en el mundo actual. La principal característica del nuevo paradigma: es que los alumnos, y ya no los maestros, son los transmisores del conocimiento; y son el centro del proceso educativo.

Esta nueva función de los maestros no sólo implica un cambio a nivel práctico, sino que requiere de ellos una nueva manera de entender el proceso educativo. Es en torno a esto, que el presente proyecto de Biblioteca Digital Distribuida, permitirá ayudar a través de una plataforma flexible y escalable.

Los maestros, antes preparados para transmitir información y para, entre otras cosas, asegurar la disciplina, la buena conducta, la responsabilidad; tienen ahora que actuar como facilitadores de los procesos de aprendizaje de sus alumnos, motivándolos y orientándolos más que dándoles información.

Sin duda, es tiempo de tomar decisiones para poder obtener resultados conducentes al cambio sistémico de nuestra sociedad. Es hora de forjar las nuevas generaciones empleando tecnologías emergentes, que permitan soportar el nuevo enfoque de un mundo globalizado, en el cual todos nosotros formamos parte activa y no pasiva. Es por ello, que el prototipo de la Biblioteca Digital Distribuida, muestra claramente que es posible y necesario la creación de una plataforma nacional en el sector educación, permitiendo grandes beneficios al país en lo que a desarrollo académico se refiere.

Capítulo 5

Conclusiones

5.1. Resumen de los resultados y sus implicancias

Con las pruebas del prototipo de la Biblioteca Digital Distribuida se concluye que es posible integrar sistémicamente a través de la arquitectura orientada a regulaciones ROA, propuesto sobre una plataforma de automatización de procesos de negocios enmarcados dentro de los procesos del sector educativo. Se ha podido validar la metodología de construcción de software ROA-RUP, que permitió especificar los requerimientos del prototipo, lo que implica que es posible realizar la toma de requerimientos en forma integral para efectuar las especificaciones del sistema de información de una biblioteca digital distribuida basada en web services en el sector Educación.

Luego la integración de la tecnología, a través de la notación de gestión de procesos de negocios BPMN usando la edición comunitaria de la plataforma BPMS Intalio, agrega intrínsecamente una robusta plataforma al integrar las mejores técnicas de software con los patrones de procesos. Sin embargo, aún se carece de equipos masivos de desarrolladores de software, capaces de llegar a cabo proyectos complejos, debido a lo nuevo del tema.

En las figuras del 5.1 al 5.15 se han tomado las pantallas de las interfaces de usuario del prototipo del sistema de biblioteca digital para mostrar la implementación de los casos de uso propuestos en el alcance de la tesis que forman parte del flujo del proceso de gestión de la biblioteca.

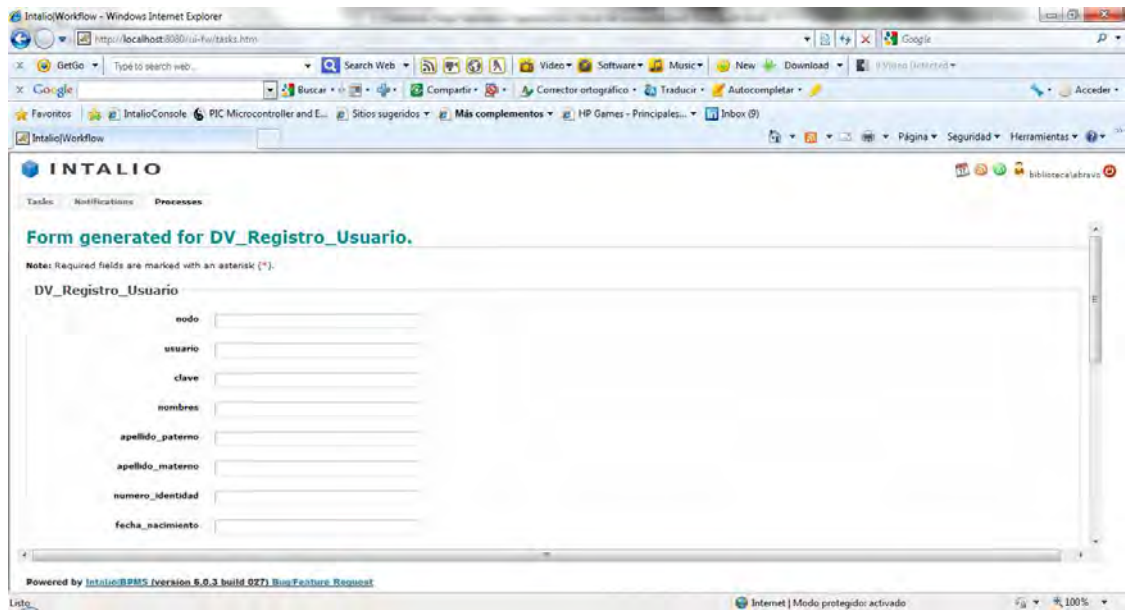


Figura 5.1: Interfaz gráfica de usuario: Registro de Usuario

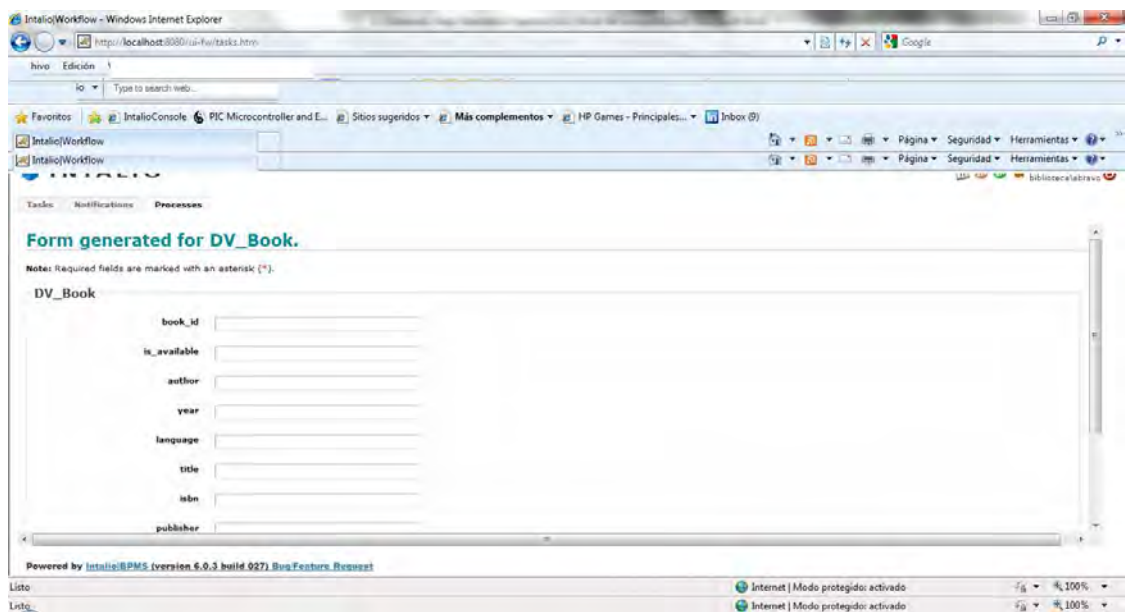


Figura 5.2 Interfaz gráfica de usuario: Registro de Libro

A través de la biblioteca digital distribuida, se demostró la viabilidad técnica de la solución, sin embargo este enfoque no necesariamente es el más adecuado en algunos escenarios; especialmente en aquellos lugares donde se carezca de recursos humanos familiarizados con la plataforma ROA-BPM.



Figura 5.3 Interfaz Gráfica de Usuario: Búsqueda de contenidos

Asimismo, el presente proyecto de Tesis provee un aporte a través de una propuesta integral tanto metodológica como tecnológica para el sector educativo. Un especial beneficio sería obtenido en programas como Huascarán, donde se hace necesario contar con una plataforma de soporte administrativo a los procesos educativos, o también en la propia biblioteca virtual de la UNI (ver Figura 5.3), en donde se hace necesario integrar las bibliotecas de cada facultad en forma horizontal y no en forma aislada como está esbozado inicialmente.

Asimismo, el hecho de plantear la solución con una plataforma netamente de código abierto, de por sí es una ventaja frente a soluciones propietarias, ya que es mejor desarrollar un software a la medida que comprarlo, pues se tiene la posibilidad de tener los códigos fuentes para realizar mejoras; además ningún software de caja se ajusta 100 % a una institución. Siempre se encuentran aspectos que se requieren ajustar y que si no está contemplado en el contrato implica desembolsos adicionales. A esto se suma la diversidad cultural y de género en nuestras regiones, que de por sí necesitan adaptaciones precisas para ingresar a sus micro sociedades.

También es importante recalcar que esta tecnología puede ser empleada en otros sectores diferentes al sector educativo: como por ejemplo el sector de telecomunicaciones, donde actualmente las tecnologías NGN utilizan para su plataforma de red inteligente la arquitectura J2EE con énfasis en web services, como es el caso de la empresa Telefónica del Perú [ver referencia 29].

En el contexto internacional, la mayoría está de acuerdo en que a partir del año 1994 se ha impulsado la creación de bibliotecas digitales, con la cual fue iniciativa en los Estados Unidos de Norte América del proyecto de biblioteca digital generada en DARPA, la Fundación Nacional de Ciencias y la NASA. En dicho proyecto el gobierno subvencionó a seis universidades para realizar la investigación referida a las bibliotecas digitales, basadas en el crecimiento de Internet y los navegadores web.

Existen exitosos casos referidos al diseño e implementación de una Biblioteca Digital, tales como:

- ◆ Repositorio Digital de la Universidad de Stanford, USA
- ◆ Biblioteca Digital del Instituto Tecnológico de Massachusetts, USA
- ◆ Repositorio Digital de la Universidad Humboldt en Berlín, Alemania

Para mayor información acerca de los casos de éxito referidos a la implementación de bibliotecas digitales, se sugiere revisar las referencias de Wikipedia.

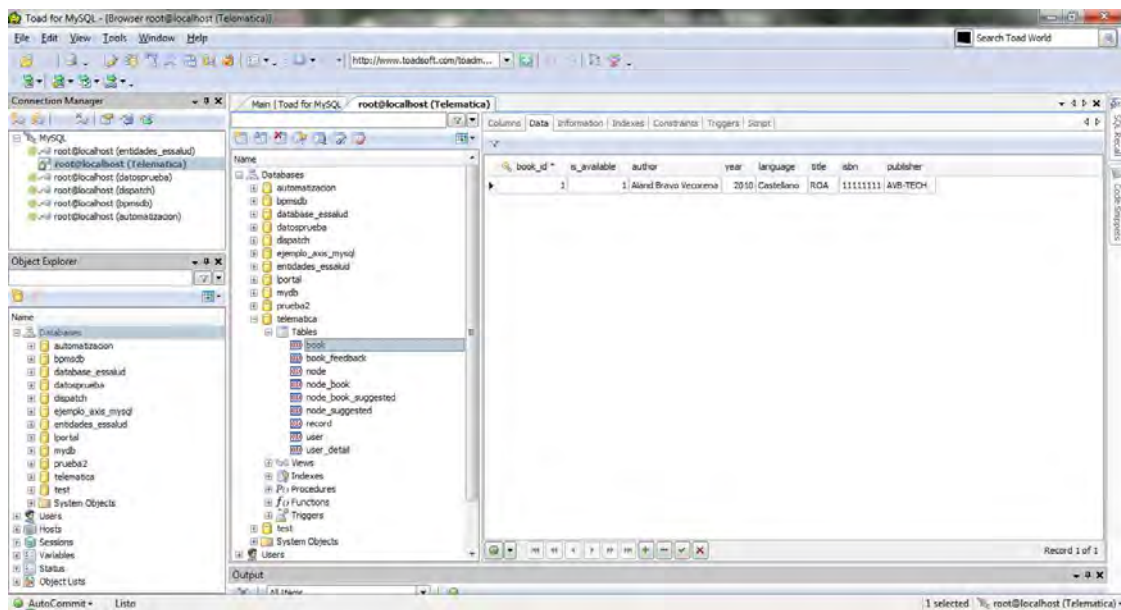


Figura 5.4 Validación en base de datos de la Interfaz gráfica de usuario:
Registro de Recurso Bibliográfico

En los últimos años, la Biblioteca Central UNI ha mejorado notablemente; sin embargo, desde el punto de vista de software gestión bibliotecaria, aún se requiere mejorar sus procesos, correspondiendo a las autoridades tomar las decisiones para implementar una estrategia integral que permita mejorar la productividad entre las áreas. Cabe resaltar también que la adquisición de un software no soluciona el problema por sí mismo, tal como se demostró, de ahí la propuesta de la Arquitectura Orientada a Regulaciones que permite incluir metodológicas y tecnológicas con cierto grado de automatización de procesos que es deseable y necesaria para que la universidad fortalezca su equipo de trabajo; el cual permita desarrollar un sistema propio que forme parte de un sistema general que soporte los distintos procesos de la universidad. Esto es alcanzable, ya que se cuenta con material humano de alta calificación ya sea en las escuelas de posgrado o a través del centro de cómputo. No obstante, se debe tener presente que el software es una herramienta y no reemplaza a las personas que forman parte del sistema.

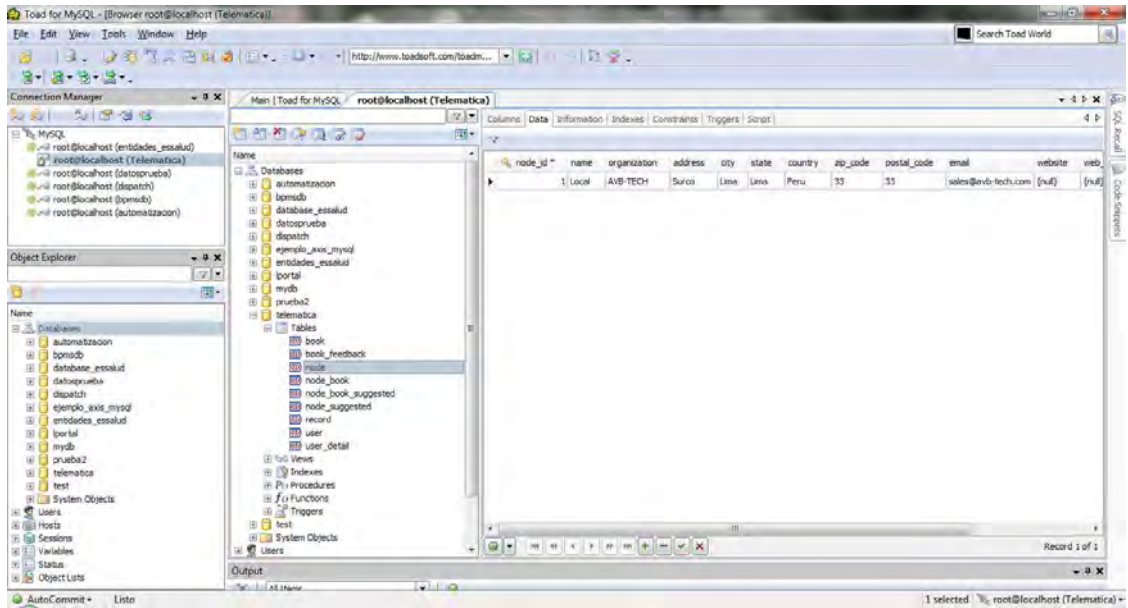


Figura 5.5 Validación en base de datos de la interface gráfica de usuario:
Registro de Nodo

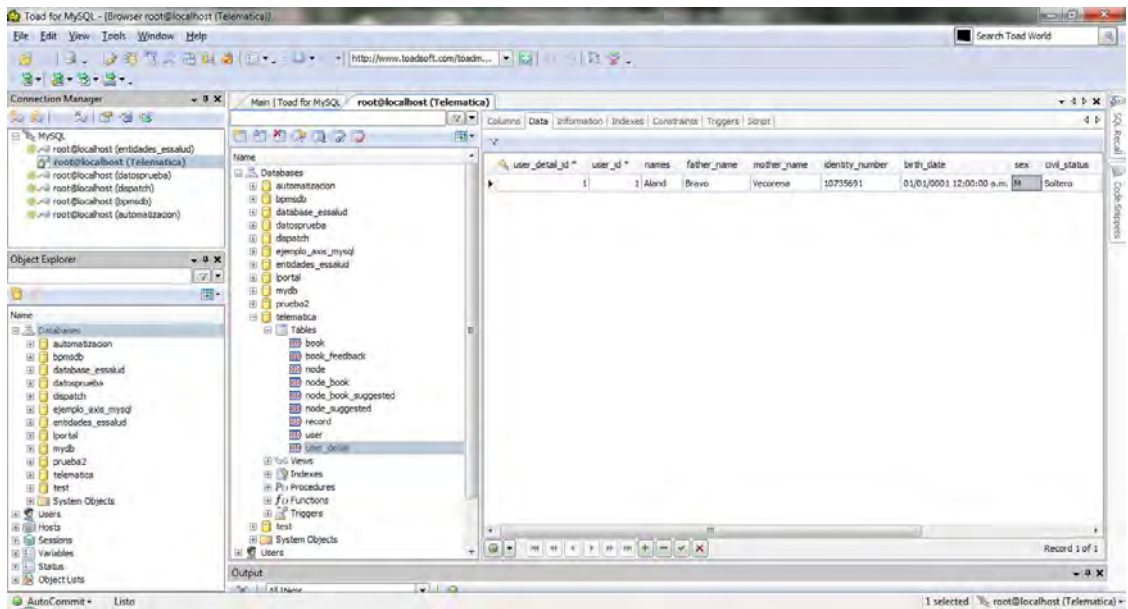


Figura 5.6 Validación en base de datos de la interfaz gráfica de usuario:
Registro de Usuario

Desde el punto de vista técnico, es recomendable utilizar la metodología GTZ-ROA con una versión modificada de la tabla ASME, ya que nos permite descubrir detalles del proceso. Luego de modelar en diagrama de bloques la totalidad del proceso, se procede a modelar tanto el macro proceso como el proceso de negocio principal utilizando la notación BPMN.

En las figuras del 5.4 al 5.6 se muestran el resultado de las consultas a nivel de base de datos al sistema de biblioteca digital distribuida a través de las pantallas de las interfaces gráficas del Workflow Tempo de la plataforma BPMS de Intalio. Un detalle importante es el nombre comercial del producto prototipo de software de la biblioteca digital distribuida: Dexter Lib, pues esto obedece a la idea de captar aceptación de los usuarios de nivel primaria, secundaria y superior; donde una de las caricaturas animadas que cumple en cierto modo con este espectro es el personaje Dexter (ver página web de cartoon network: <http://www.cartoonnetwork.com/tv-shows/dexter/>).

De este modo la primera impresión con el sistema de gestión administrativa de biblioteca digital distribuida le resultaría familiar a la comunidad estudiantil, permitiendo su rápida inserción en su rol protagónico de usuario final.

A continuación se muestra la salida de la configuración del servidor del web service durante el despliegue en tiempo real donde se verifica la conectividad, con un tiempo de despliegue aproximado de 57471 ms en una Pc HP AMD Triple Core con 4 G RAM.

```
16:37:20,951 0 INFO [org.apache.catalina.startup.Catalina] (main:) Initialization
processed in 1622 ms
16:37:41,372 20421 WARN [org.springframework.jmx.support.JmxUtils] (main:) Found more
than one MBeanServer instance. Returning first from list.
16:38:18,656 57705 INFO [org.apache.catalina.startup.Catalina] (main:) Server startup in
57471 ms
```

5.2. Recomendaciones para trabajos futuros

Las aplicaciones basadas en sistemas distribuidos ofrecen una excelente oportunidad de poner las bases tecnológicas de un futuro centro de telemática, que preste servicios de información no sólo al pabellón central sino a toda la universidad a escala nacional; fomentando la creación de cursos virtuales a ser dictados por nuestros docentes que generen recursos a las facultades, de modo que permita un sistema de conocimiento autosostenido.

El aporte inédito de la Arquitectura Orientada a Regulaciones ha surgido de la experiencia del autor al momento de desarrollar e implementar sistemas complejos sobre BPM, de modo que queda por descubrir su aplicación en diversos escenarios: por ejemplo: ROA sobre tecnologías J2EE, .NET, PHP, entre otras.

En la universidad los estudiantes requieren experiencia en desarrollo de aplicaciones complejas de software. Siendo la universidad por sí misma un buen candidato para consumir aplicaciones, es factible que se pueda organizar un grupo de este abundante recurso humano y tener un ecosistema donde todos ganen. La universidad como proveedor de recursos humanos de alta calidad (y con experiencia adquirida por los docentes especialistas en el tema) ganaría en desarrollos de software que le son necesarios, los alumnos que recién comienzan bajo supervisión ganarían experiencia, y las empresas privadas que hayan consolidado una alianza con la institución de la UNI, ganarían experiencia como incubadora de productos de software que podrían ser financiados como posibles temas de investigación aplicada o tesis. En el marco actual de la ley universitaria, se están cambiando los paradigmas de la universidad, para insertar conceptos como valles del silicio, parques tecnológicos o incubadoras de empresas, así como la existencia de cursos y syllabus dinámicos.

Desde el punto de vista técnico, se recomienda antes de usar una herramienta

de desarrollo, aprender una metodología, luego profundizar en las técnicas de patrones de procesos en un lenguaje de procesos estándar tal como BPMN; con ello se asegura el uso eficiente de las herramientas de alto nivel; que de otro modo se podrían tornar complicados de entender. Como parte de los entregables de la Tesis: el autor ha preparado un CD, donde se detalla a nivel de código, la implementación de los casos de usos utilizados internamente en el proceso de gestión descritos en el volumen de tesis; esto es útil en la medida que se separa las especificaciones del proceso basado en el nuevo enfoque ROA-BPM con los casos de uso del enfoque tradicional SOA-J2EE.

En la etapa de implementación de código se recomienda guardar cada cierto tiempo backups de versiones preliminares. Asimismo se recomienda cerrar los puertos que no están en uso para servicios de red a nivel de sistema operativo.

Como regla general es preferible proteger sólo los archivos de configuración como sólo lectura. Paralelamente se sugiere utilizar librerías especializadas para validar la consistencia de los módulos de software; así como agilizar la compilación de las mismas. En el caso de ROA, se sugiere utilizar repositorios como Maven. También es importante recalcar lo siguiente:

- BPMS no sustituye al desarrollo de sistemas, sino se complementa como contraparte de los procesos de negocios.
- El portal se integra al BPMS para agregar funcionalidad distribuida, pero la solución total se torna más compleja.
- El Workflow del BPMS tiene funcionalidades de interface de usuario para los actores del BPMS.
- El prototipo desarrollado cumple con el objetivo propuesto de prueba de concepto.
- ROA es una nueva arquitectura que complementa el enfoque propuesto con BPMS-Intalio.

En el contexto histórico, en el año 2002 fecha donde se inició el presente proyecto de tesis, los web services tuvieron varias innovaciones y cambios sustantivos. En este contexto el presente proyecto da como resultado la habilidad conjunta de la tecnología ROA-BPM aplicado a diversos ámbitos; tales como educación, gobierno, etc. Un especial interés está surgiendo en los teléfonos móviles que soportan Java (usualmente se conoce como J2ME) en donde gran parte de la estrategia utilizada en el prototipo de Biblioteca Digital Dexter's Lib es aplicable a esta tendencia emergente, donde el acceso a los Web Services juega un papel importante. A esto se suma el advenimiento de dispositivos de 3G que converge voz, video y datos en forma compacta, lo cual permite dar valor añadido a los sistemas tradicionales.

Finalmente, nuestros gobiernos que enrumban el camino de nuestra sociedad peruana, han privilegiado su acción gubernamental a adquisición de infraestructura pública, carreteras, colegios, computadoras; pero han olvidado lo más importante: que es usar las tecnologías de sistemas de información para automatizar procesos empresariales e industriales; de modo que tanto los procesos administrativos del Estado Peruano como de las empresas sean cada día más eficientes. De ahí la importancia de esta propuesta, enmarcada en un sistema de información encaminada al diseño e implementación de una Biblioteca Digital Distribuida con Web Services basado en ROA-BPM, para integrar a las regiones y localidades a nivel nacional e internacional.

Apéndice A

Proyecto Huascarán

Introducción

En el marco de la política educativa nacional, se busca contribuir a mejorar la calidad de la educación en un contexto de equidad. Pretende ser un factor de innovación educativa, de cierre de las asimetrías sociales, en particular, de la brecha digital, y un catalizador en el tránsito hacia la sociedad del conocimiento, mediante la incorporación de las nuevas tecnologías de la información y la comunicación al sistema educativo. El Plan Estratégico se define con un horizonte de 8 años, dentro de la declaratoria de Emergencia de la Educación definida por el Gobierno del Perú. Así, el Plan Estratégico del Programa Huascarán se concibe como un proceso participativo, mediante el cual se articula la contribución al sistema educativo peruano, por medio de la incorporación sistemática y el uso extendido de las TIC en los procesos de enseñanza-aprendizaje.

Visión y Misión

La Visión del Programa Huascarán: La comunidad educativa peruana tiene pleno acceso a las TICs, las usa intensamente y las incorpora gradualmente en su actividad cotidiana, para potenciar las capacidades de socialización del conocimiento, creatividad e innovación y participar en el desarrollo global de la sociedad del conocimiento en igualdad de condiciones.

La Misión del Programa Huascarán: Programa Estratégico, especializado en el uso educativo de las TICs; que promueve y desarrolla investigaciones e innovaciones para la integración de las TICs a la educación peruana, en el marco de la interculturalidad, de acuerdo a normas y estándares internacionales,

para facilitar el cambio cualitativo en la forma de educar, permitiendo la construcción del conocimiento significativo e incorporando los valores postulados en la política educativa peruana, expresados en el Acuerdo Nacional.

Objetivos Generales

- Contribuir a ampliar la cobertura y mejorar la calidad de la educación mediante el uso de las TICs y de programas pedagógicos de educación a distancia, permitiendo así su descentralización, democratización y equidad; dando prioridad a las áreas donde exista mayor concentración poblacional, las zonas rurales, de selva, de frontera y urbano marginales; dotándolas de información intercultural sustentada en valores y con visión productiva de manera tal que se generen sociedades del conocimiento realmente democráticas y disminuir, por tanto, la brecha digital, entre los peruanos y entre el Perú y los países más desarrollados.
- Implementar una plataforma tecnológica que permita mejorar la capacidad comunicativa y de conectividad de las entidades del Estado, de los centros educativos y de la comunidad en general, que permita optimizar los servicios educativos y desarrollar programas multisectoriales con la participación de educación, salud, agricultura, defensa civil, etc.
- Desarrollar y proveer a la comunidad educativa sistemas de información con recursos especializados en las TICs, para fortalecer el proceso de enseñanza y aprendizaje; apoyando la democratización del acceso a Internet y otras TICs y promoviendo la actualización tecnológica y operativa de los sistemas de información que se implementen.
- Promover el acceso, la capacitación y el perfeccionamiento de los docentes en el uso de las TICs con fines educativos.
- Propiciar la eficiencia en el proceso de administración de la educación mediante el uso de las TICs mejorando así los resultados y reduciendo los costos de operación.

Modelo de Programación

La misión del Programa Huascarán se hará efectiva mediante el impulso de ocho grandes subprogramas, los cuales cubren las diversas etapas y articulan diversos componentes:

- Planificación y atención a la demanda
- Plataforma pedagógica y de capacitación
- Plataforma tecnológica y conectividad
- Plataforma de producción y difusión multimedia
- Infraestructura física
- Gestión administrativa
- Movilización de recursos
- Subprogramas transversales (intersectorialidad)

La puesta en ejecución de los subprogramas se realiza por medio de proyectos ligados a componentes y la estructura administrativa del Programa.

Periodización

El Plan Estratégico considera tres grandes etapas:

Etapas: **A:** Acciones Inmediatas (Relanzamiento del programa). Período comprendido entre el 01 de octubre del 2002 y 31 de marzo del 2003.

Etapa B: Intervención acelerada orientada a completar la infraestructura y la conectividad a Internet, impulsar la difusión multimedia e iniciar el desarrollo de aplicaciones. Abarca el periodo 01 de abril al 31 de diciembre del 2003.

Etapa C: Ampliación y consolidación incluye 2 grandes fases:

- Ampliación de atención a la demanda que va del 01 de enero del 2004 al 31 de diciembre del 2005.
- Consolidación y preparación para la sostenibilidad que va del 01 de enero del 2004 al 31 de diciembre del 2005.

El Programa continuará en su etapa de expansión y consolidación hasta el 2010, integrado ya como un programa permanente del sistema educativo peruano.

Modelos de Atención

El Programa Huascarán brinda atención bajo los siguientes modelos:

Informática Educativa - Espacios de innovación sustentados en aula de innovación de informática educativa convencional o móvil.

Acceso a cabinas públicas o megacentros - Espacios en los cuales alumnos y docentes pueden tener acceso a Internet (aulas de innovación, cabinas, megacentros y unidades móviles).

Televisión educativa - Establece franjas horarias claramente segmentadas por niveles educativos, modalidades y grados de acuerdo al currículo. Se contemplarán espacios para estudiantes con dificultades de aprendizaje y para el apoyo de la educación no formal en el país. Contempla también el establecimiento de alianzas con otras redes de televisión educativa a nivel nacional, regional y mundial.

Radio educativa - Programas radiales multipropósito e interactivos diseñados por niveles y grados con miras a un aprendizaje más efectivo, complementado con libros.

Áreas del Programa Huascarán

Pedagógica. Área que propicia la innovación y diseña las metodologías de aplicación de las TIC con el objetivo de mejorar la calidad de la educación.

Capacitación. Esta área realiza las acciones de capacitación intensiva para el personal y docentes, coordinadas como acciones dentro del esquema pedagógico.

Producción de recursos y portal web. Área que produce y distribuye recursos y herramientas TIC a través de la Internet para el desarrollo de capacidades tanto en docentes como en alumnos.

Plataforma tecnológica y redes. Para garantizar el adecuado acceso y uso de las TIC es necesario contar con la infraestructura y el conocimiento adecuado.

Estructura organizativa. Concentrada en crear una organización sólida tanto en el ámbito local y regional como en el nacional. Ello asegurará la permanencia en el tiempo.

Estructura Organizativa

La dirección del Programa Huascarán está integrada de la siguiente forma:

Comité Intersectorial - Integrado por el Director Ejecutivo, quien lo preside; el Director Pedagógico; el Director de Informática y comunicación; un representante del Vice Ministerio de Gestión Pedagógica del MED; y un representante del Vice Ministerio de Comunicaciones del MTCVC.

Dirección Ejecutiva - De la cual dependen la Dirección Pedagógica y la Dirección de Informática y Comunicación.

Dirección Pedagógica Informática y de Telecomunicaciones - Quienes tienen a su cargo el planeamiento y diseño de estrategias en sus respectivas áreas. En tal sentido la Dirección Pedagógica identifica los contenidos para el aprendizaje a ser incorporados en la red nacional y la Dirección de Informática y Telecomunicaciones planifica la infraestructura y diseña los sistemas de Telecomunicaciones y Telemática del Programa.

La Estructura Nacional que sostiene el Programa y los agentes participantes son los siguientes:

a) Organización Central del Programa Huascarán - La Organización Central del Programa Huascarán se encuentra actualmente ubicada en Lima. Entre sus funciones se encuentra la de prestar apoyo a las Organizaciones Departamentales.

Aunque se trabajará con una arquitectura descentralizada, la sede de la Organización Central albergará una parte importante de la plataforma tecnológica; sobre todo en lo que se refiere a servidores centrales, parte de la plataforma de desarrollo de contenidos, estaciones de transmisión de video y, en algunos casos, infraestructura de acceso a Internet. Asimismo, se desarrollarán materiales para cursos que alimentarán los contenidos pedagógicos del programa. La Organización Central del Programa Huascarán interconectará las diferentes instancias administrativas del sistema de educación nacional, para que puedan cumplir mejor todas sus funciones tanto pedagógicas como de gestión administrativa, facilitando el intercambio de información y promoviendo el desarrollo del conocimiento sectorial.

b) Organizaciones Departamentales - Las Organizaciones Departamentales

replican muchas de las funciones y características de la Organización Central del Programa Huascarán. En principio, estas organizaciones estarán superpuestas en los órganos Intermedios (Regionales, Sub regionales, ADES, USES, etcétera) del Ministerio de Educación.

c) Módulos Concentradores Locales - Los Módulos concentradores locales sirven directamente a los módulos de usuario final (centros educativos) de su radio de acción. La cantidad de módulos de usuario final a los que servirán, dependerá de consideraciones técnicas y administrativas.

d) Centros Educativos del Programa Huascarán - Un Centro Educativo del Programa Huascarán es aquel que está en capacidad de aprovechar los contenidos pedagógicos del programa. Esto quiere decir que, con excepción de posibles casos particulares, un centro educativo Huascarán cuenta, como mínimo, con personal docente capacitado, algún medio de acceso a la Plataforma Tecnológica en sus instalaciones y los materiales de apoyo correspondientes.

En principio se preferirá que los centros educativos cuenten con sus propios módulos Huascarán como medio de acceso de la plataforma, aunque, cuando sea necesario, el colegio podrá utilizar alianzas con cabinas públicas locales u otras alternativas para llevar a cabo dicho acceso. En todos los casos se instalarán sobre un sistema de redes sociales que potenciarán la cobertura del programa atendiendo un número determinado de centros educativos cercanos.

e) Docente del Programa Huascarán - El docente del programa Huascarán es aquel que ha sido capacitado y que formará parte de un proceso de formación continua en el uso de las TIC. Este docente estará en capacidad de utilizar los contenidos pedagógicos y la plataforma tecnológica del programa en beneficio de sus alumnos, de su comunidad y de sí mismo.

f) Alumno o alumna del Programa Huascarán - El alumno o alumna del Programa Huascarán es aquel que gozará de una educación renovada, gracias a las nuevas competencias de sus docentes. Asimismo, contará con herramientas basadas en las TIC para mejorar la calidad y eficiencia de sus procesos de estudio e investigación y formará parte de comunidades de estudiantes locales, nacionales e internacionales, gracias a las aplicaciones de comunicación que tendrá a su disposición.

g) Comunidad del Programa Huascarán - Será aquella que se beneficiará del esfuerzo inicial del Programa Huascarán y que se integrará en el gran reto de mantener y continuar con el mismo los siguientes años. La comunidad Huascarán comparte con el Estado la responsabilidad de desarrollar y dinamizar los mercados locales alrededor de las herramientas puestas a su disposición de tal manera que se asegure la sostenibilidad del programa sobre la base de la generación de valor al interior de cada comunidad.

Apéndice B

Glosario de Términos

A continuación se detalla los términos usados en el presente proyecto:

Término	Definición
SIE	Servicio de Información Estudiantil
SB	Servicio de Biblioteca
SE	Software Engineering o Ingeniería de Software
FIIS	Facultad de Ingeniería Industrial y de Sistemas
UML	Lenguaje de Modelado Unificado
TIC	Tecnología de Información y Comunicación
J2EE	Java para Edición Empresarial
JVM	Máquina Virtual de Java
JSP	Página de Servidor Java
J2SE	Java para Edición Estándar
J2ME	Java para Micro Edición
EAI	Enterprise Application Integration ó Integración de Aplicación Empresarial
RDF	Resource Description Framework
W3C	World Wide Web Consortium
RTP/RTCP	Real Time Protocol / Control Protocol
TCP/IP	Protocolo de Control de Transmisión / Protocolo de Internet
RPC	Llamada a Procedimiento Remoto
WSDL	Lenguaje de Descripción de Servicios Web
MIME	Multipurpose Internet Mail Extensions
MTCVC	Ministerio de Transportes, Comunicaciones, Vivienda y Construcción
CAP	CAMEL Application Part
CAMEL	Customized Applications for Mobile Network Enhanced Logic
INAP	Intelligent Application Part
MAP	Mobile Application Part
WIN	CDMA Wireless intelligent Protocol
IEEE	Instituto de Ingenieros Electricistas y Electrónicos

Término	Definición
IMS	Sistema de Administración de Información
QTI	Question and Test Interoperability
AICC	Aviation Industry CBT (Computer Based Training) Committee
ADL	Advanced Distributed Learning
SCORM	Sharable Content Object Reference Model
CMI	Computer Managed Instruction
SOAP	Simple Object Access Protocol
EJB	Enterprise Java Bean
LOM	Learning Object Metadata
JAX Pack	Paquete Java XML, este paquete brinda varios estándares para XML (SAX, DOM, XSLT, SOAP, UDDI, ebXML y WSDL)
URI	Uniform Resource Identifier, identificador unificado de recursos
UN/SPSC	United Nations Standard Products and Services Code
BPEL4WS	Business Process Execution Language for Web Services
VPN	Virtual Private Network
JMS	Java Message Service es parte del paquete de la tecnología J2EE
JNDI	Java Name Directory Interface
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
XKMS	XML Key Management Service
SAML	Security Assertion Markup Language
WSIL	Web services inspection language
WSIF	Web services invocation framework
WSGW	Web Services Gateway
WSFL	Web services flow language
OMG	Object Management Group
ORBs	Object Request Brokers
IOR	Interoperable Object Reference
AIX	Versión del sistema operativo UNIX que IBM desarrolla
POP	Point of Presence
MOM	Message-oriented middleware
P2P	Peer-to-peer

Término	Definición
LMS	Sistema de Administración del Conocimiento
AIC	Advanced Industrialized Countries
TIC	Tecnologías de la información y la comunicación
RPC	Llamada a Procedimiento Remoto
TCP	Protocolo de Control de Transmisión
IP	Protocolo de Internet
RMI	Java Remote Method Invocation
CORBA	Common Object Request Broker Architecture
OMG	Grupo de Gestión de Objetos
DCOM	Modelo de Objetos de Componentes Distribuidos
RMI-IIOP	Internet Inter-Orb Protocol
JAF	JavaBeans Activation Framework
JMS	Java Message Service
JTA	API para transacciones en Java
JNDI	Interfaz de Nombrado y Directorio Java
JDBC	Java Database Connectivity
JTS	Java Transaction Service
ebXML	Electronic Business eXtensible Markup Language
BPMN	Business Process Management Notation
BPMS	Business Process Management Server
RUP	Rational Unified Process
URL	Universal Resource Locator
BI	Business Intelligent
BR	Business Rules
BPEL	Business Process Execution Language
Vo-IP	Voice over Internet Protocol

Apéndice C

Código Fuente

A continuación se detalla las partes más resaltantes del código fuente del prototipo de la Biblioteca Digital Distribuida Basada en Web Services sobre la Arquitectura Orientada a Regulaciones ROA-BPM:

Detalles de Codificación

A continuación el detalle de codificación de las consultas SQL:

Procedimiento Almacenado Buscar Libro:

```
select b.book_id, n.name, b.is_available, b.author, b.year, b.language, b.title, b.isbn, b.publisher from book
as b left join node_book as nb on
nb.book_id=b.book_id left join node as n on n.node_id = nb.node_id
where n.name = ? and b.title = ?;
```

Procedimiento Almacenado Insertar Usuario:

```
/*call transaccionRegistrarUsuario('Local','miflaca','pamela','La Molina','Lima','Lima','Peru','33','Lima
33','psumarriba@hotmail.com','Pamela','Sumarriba','Hibaja','66666666','30','Femenino','Soltera','Bachiller','U
PC','2010');*/
call transaccionRegistrarUsuario(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?);
```

Procedimiento Almacenado Registrar Préstamo:

```
/* call transaccionRegistrarPrestamo ('abravo', 'ROA','Local', '2010/10/10', '2010/10/20', 'Verdadero','El
prestamos ha sido realizado sin ningun inconveniente'); */
call transaccionRegistrarPrestamo(?,?,?,?,?,?);
```

Detalle del Procedimiento Almacenado de la Transacción Registrar Libro:

```
DROP PROCEDURE IF EXISTS Telematica.transaccionRegistrarLibro;
CREATE PROCEDURE Telematica.`transaccionRegistrarLibro` (
/*Desde SCHEMA node */
in name_node VARCHAR(250),
/*Desde SCHEMA book */
in is_available_book VARCHAR(4),
in author_book VARCHAR(250),
in year_book VARCHAR(4),
in language_book VARCHAR(250),
```

```

        in title_book VARCHAR(250),
        in isbn_book VARCHAR(250),
        in publisher_book VARCHAR(250))

BEGIN
/* Seccion de codigo para Declaracion de Variables */
/* Variables de node */
declare SP_name_node CHAR(250);
/*Variables de book */
declare SP_is_available_book CHAR(250);
declare SP_author_book CHAR(250);
declare SP_year_book CHAR(4);
declare SP_language_book CHAR(250);
declare SP_title_book CHAR(250);
declare SP_isbn_book CHAR(250);
declare SP_publisher_book CHAR(250);

/*Variables internas para los id */
declare SP_node_id CHAR(22);
declare SP_book_id CHAR(22);

/* Manejo de la Tabla Node */

/* set @SP_name_node = 'Local'; */
set @SP_name_node = name_node;
set @node_id = (SELECT n.node_id from Telematica.node as n
                where (n.name = @SP_name_node) );
/* select @node_id; */

/* Manejo de la Tabla Node_Book */

/* set @SP_is_available_book = '1'; */
set @SP_is_available_book = is_available_book;
set @is_available = cast(@SP_is_available_book as CHAR(4));
/* select @is_available;*/

/* set @SP_author_book = 'Aland Bravo Vecorena'; */

```

```

set @SP_author_book = author_book;
set @author = @SP_author_book;
/* select @author;*/

/* set @SP_year_book = '2010'; */
set @SP_year_book = year_book;
set @year = @SP_year_book;
/* select @year;*/

/* set @SP_language_book = 'Castellano'; */
set @SP_language_book = language_book;
set @language = @SP_language_book;
/* select @language;*/

/* set @SP_title_book = 'Arquitectura de Procesos'; */
set @SP_title_book = title_book;
set @title = @SP_title_book;
/* select @title;*/
/* set @SP_isbn_book = '33333333'; */
set @SP_isbn_book = isbn_book;
set @isbn= @SP_isbn_book;
/* select @isbn;*/

/* set @SP_publisher_book = 'LIBUN'; */
set @SP_publisher_book = publisher_book;
set @publisher = @SP_publisher_book;
/* select @publisher;*/
start transaction;

INSERT INTO telematica.book(is_available, author, year, language, title, isbn, publisher)
VALUES (@is_available, @author, @year, @language, @title, @isbn, @publisher);
set @book_id = LAST_INSERT_ID();

INSERT INTO telematica.node_book(book_id, node_id) VALUES (@book_id, @node_id);
commit;
END;

```

Detalle del Procedimiento Almacenado de la Transacción Registrar Libro:

```
DROP PROCEDURE IF EXISTS Telematica.transaccionRegistrarPrestamo;
CREATE PROCEDURE Telematica.`transaccionRegistrarPrestamo` (
/*Desde SCHEMA user */
    in name_user VARCHAR(250),
/*Desde SCHEMA book */
    in title_book VARCHAR(250),
/*Desde SCHEMA node */
    in name_node VARCHAR(250),
/*Desde SCHEMA record */
    in start_date_record VARCHAR(12),
    in finish_date_record VARCHAR(12),
    in status_record VARCHAR(9),
    in description_record VARCHAR(250))
BEGIN
/* Seccion de codigo para Declaracion de Variables */
/* Variables de user */
    declare SP_name_user CHAR(250);
/*Variables de book */
    declare SP_title_book CHAR(250);
/*Variables de node */
    declare SP_name_node CHAR(250);
/*Variables de record */
    declare SP_start_date_record DATE;
    declare SP_finish_date_record DATE;
    declare SP_status_record CHAR(9);
    declare SP_description_record CHAR(250);
/*Variables internas para los id */
    declare SP_user_id CHAR(22);
    declare SP_book_id CHAR(22);
    declare SP_node_id CHAR(22);
/* Manejo de la Tabla User */
/* set @SP_name_user = 'abravo'; */
    set @SP_name_user = name_user;
    set @user_id = (SELECT u.user_id from Telematica.user as u
        where (u.name = @SP_name_user) );
```

```

/* select @user_id; */

/* set @SP_title_book = 'ROA'; */
set @SP_title_book = title_book;
set @book_id = (SELECT b.book_id from Telematica.book as b
                where (b.title = @SP_title_book) );
/* select @book_id; */
/* set @SP_name_node = 'Local'; */
set @SP_name_node = name_node;
set @node_id = (SELECT n.node_id from Telematica.node as n
                where (n.name = @SP_name_node) );
/* select @node_id; */
/* set @SP_start_date = '04-10-2010'; */
set @SP_start_date = start_date_record;
set @start_date = cast(@SP_start_date as DATE);
/* set @start_date = STR_TO_DATE(cast(@SP_start_date as DATE),'YYYY-MM-DD');*/
/* select @start_date; */
/* set @SP_finish_date = '06-10-2010'; */
set @SP_finish_date = finish_date_record;
set @finish_date = cast(@SP_finish_date as DATE);
/* select @finish_date; */
/* set @SP_status_record = 'Verdadero'; */
/* set @SP_status_record = 'TRUE'; */
set @SP_status_record = status_record;
set @status= UPPER(@SP_status_record);
set @var1 = (select UPPER(@SP_status_record ));
set @status= (SELECT IF ( STRCMP(@var1, 'VERDADERO'),'0','1')) or (SELECT IF
( STRCMP(@var1, 'TRUE'),'0','1')) or (SELECT IF ( STRCMP(@var1, '1'),'0','1'));
/* select @status; */
/* set @SP_description_record = 'El prestamos ha sido realizado sin ningun inconveniente'; */
set @SP_description_record = description_record;
set @description = @SP_description_record;
/* select @description; */

start transaction;
/* set @user_id = LAST_INSERT_ID(); */
INSERT INTO telematica.record(user_id, book_id, node_id, start_date, finish_date, status,

```


description)

VALUES (@user_id, @book_id, @node_id, @start_date, @finish_date, @status, @description);

commit;

END;

Detalle del Procedimiento Almacenado de la Transacción Registrar Libro:

```
DROP PROCEDURE IF EXISTS Telematica.transaccionRegistrarUsuario;
```

```
CREATE PROCEDURE Telematica.`transaccionRegistrarUsuario` (
```

```
/*Desde SCHEMA node */
```

```
in name_node VARCHAR(250),
```

```
/*Desde SCHEMA user */
```

```
in password_user VARCHAR(250),
```

```
in name_user VARCHAR(250),
```

```
in address_user VARCHAR(250),
```

```
in city_user VARCHAR(250),
```

```
in state_user VARCHAR(250),
```

```
in country_user VARCHAR(250),
```

```
in zip_code_user VARCHAR(250),
```

```
in postal_code_user VARCHAR(250),
```

```
in email_user VARCHAR(250),
```

```
/*Desde SCHEMA user_detail */
```

```
in names_user_detail VARCHAR(250),
```

```
in father_name_user_detail VARCHAR(250),
```

```
in mother_name_user_detail VARCHAR(250),
```

```
in identity_number_user_detail VARCHAR(12),
```

```
in age_user_detail VARCHAR(3),
```

```
in sex_user_detail VARCHAR(9),
```

```
in estado_civil_user_detail VARCHAR(250),
```

```
in estudio_academico_user_detail VARCHAR(250),
```

```
in centro_academico_user_detail VARCHAR(250),
```

```
in codigo_academico_user_detail VARCHAR(250))
```

```
BEGIN
```

```
/* Seccion de codigo para Declaracion de Variables */
```

```
/* Variables de user */
```

```
declare SP_name_node CHAR(250);
```

```
declare SP_password_user CHAR(250);
```

```
declare SP_name_user CHAR(250);
```

```
declare SP_address_user CHAR(250);
```

```
declare SP_city_user CHAR(250);
```

```
declare SP_state_user CHAR(250);
```

```
declare SP_country_user CHAR(250);
```

```
declare SP_zip_code_user CHAR(250);
```

```
declare SP_postal_code_user CHAR(250);
```

```
declare SP_email_user CHAR(250);
```

```
/* Variables de user_details */
```

```
declare SP_names CHAR(200);
```

```
declare SP_father_name CHAR(250);
```

```
declare SP_mother_name CHAR(250);
```

```

declare SP_identity_number CHAR(12);
declare SP_age CHAR(3);
declare SP_sex CHAR(9);
declare SP_estado_civil CHAR(250);
declare SP_estudio_academico CHAR(250);
declare SP_centro_academico CHAR(250);
declare SP_codigo_academico CHAR(250);
/*Variables internas para los id */
declare user_id VARCHAR(22);

/* Manejo de la Tabla User */

/* set @SP_name_node = 'Local'; */
set @SP_name_node = name_node;
set @node_id = (SELECT n.node_id from Telematica.node as n
                where (n.name = @SP_name_node) );
/* select @node_id; */

/* set @SP_password_user = 'abravo1'; */
set @SP_password_user = password_user;
set @password = @SP_password_user;
/* select @password; */

/* set @SP_name_user = 'aladino'; */
set @SP_name_user = name_user;
set @name = @SP_name_user;
/* select @name; */

/* set @SP_address = 'Nelson Guia Gonzales 415'; */
set @SP_address_user = address_user;
set @address = @SP_address_user;
/* select @address; */

/* set @SP_city_user = 'Huanuco'; */
set @SP_city_user = city_user;
set @city = @SP_city_user;
/* select @city; */

/* set @SP_state_user = 'Andres Avelino Caceres'; */
set @SP_state_user = state_user;
set @state = @SP_state_user;
/* select @state; */

/* set @SP_country_user = 'Peru'; */
set @SP_country_user = country_user;
set @country = @SP_country_user;
/* select @country; */

/* set @SP_zip_code_user = 'Lima 33'; */
set @SP_zip_code_user = zip_code_user;
set @zip_code = @SP_zip_code_user;
/* select @zip_code; */

```

```

/* set @SP_postal_code_user = 'aladino'; */
set @SP_postal_code_user = postal_code_user;
set @postal_code = @SP_postal_code_user;
/* select @postal_code; */

/* set @SP_email_user = 'aladino'; */
set @SP_email_user = email_user;
set @email = @SP_email_user;
/* select @email; */

/* Manejo de la Tabla User_Detail */
/* set @SP_names = 'Aland'; */
set @SP_names = names_user_detail;
set @names = @SP_names;
/* select @names; */

/* set @SP_father_name = 'Aland'; */
set @SP_father_name = father_name_user_detail;
set @father_name = @SP_father_name;
/* select @father_name; */

/* set @SP_mother_name = 'Aland'; */
set @SP_mother_name = mother_name_user_detail;
set @mother_name = @SP_mother_name;
/* select @mother_name; */

/* set @SP_identity_number = 'Aland'; */
set @SP_identity_number = identity_number_user_detail;
set @identity_number = @SP_identity_number;
/* select @identity_number; */

/* set @SP_age = 'Aland'; */
set @SP_age = age_user_detail;
set @age = @SP_age;
/* select @age; */

/* set @SP_sex = 'Aland'; */
set @SP_sex = sex_user_detail;
set @sex = @SP_sex;
/* select @sex; */

/* set @SP_estado_civil = 'Aland'; */
set @SP_estado_civil = estado_civil_user_detail;
set @civil_status = @SP_estado_civil;
/* select @civil_status; */

/* set @SP_estudio_academico = 'Aland'; */
set @SP_estudio_academico = estudio_academico_user_detail;
set @academic_study = @SP_estudio_academico;
/* select @academic_study; */

/* set @SP_centro_academico = 'Aland'; */
set @SP_centro_academico = centro_academico_user_detail;
set @academic_center = @SP_centro_academico;

```

```

/* select @academic_center; */

/* set @SP_codigo_academico = 'Aland'; */
set @SP_codigo_academico = codigo_academico_user_detail;
set @academic_code = @SP_codigo_academico;
/* select @academic_code; */

start transaction;
INSERT INTO telematica.user (node_id, password, name, address, city, state, country, zip_code, postal_code, email)
VALUES (@node_id, @password, @name, @address, @city, @state, @country, @zip_code, @postal_code, @email);

set @user_id = LAST_INSERT_ID();

INSERT INTO telematica.user_detail(user_id, names, father_name, mother_name, identity_number, age, sex, civil_status, academic_study,
academic_center, academic_code)
VALUES (@user_id, @names, @father_name, @mother_name, @identity_number, @age, @sex, @civil_status, @academic_study,
@academic_center, @academic_code);
commit;

END;

```

Para mayor detalle, revisar el CD de la tesis que contiene la totalidad del código fuente del proyecto, tanto a nivel de diseño como a nivel de implementación y despliegue.

Bibliografía

- [1] Tom Debevoise and Rick Geneva. *The Microguide of Process Modeling*. Tipping Point, 2010.
- [2] Anil Gurnani. *Web Development with TIBCO General Interface*. Addison Wesley, 2009.
- [3] Tong Ka lok. *Developing Web Services with CXF and Axis 2*. Tip Tec Development, 2010.
- [4] Bruce Tate, Mike Clark, Bob Lee, y Patrick Linskey. *Bitter EJB Manning*, 2003.
- [5] Maruyama, Tamura, Uramoto. *Creación de sitios Web con XML y Java*, Prentice Hall 2000.
- [6] Laura Lemay y Rogers Cadenhead. *Aprendiendo Java 2 en 21 Días*, Pearson, Prentice Hall 1999.
- [7] H. M. Deitel y P. J. Deitel. *Cómo programar en Java*, Prentice Hall 1998.
- [8] Subrahmanyam Allamaraju, Cedric Beust, John Davies, Tyler Jewell, Rod Johnson, Andrew Longshaw, Ramesh Nagappan, Daniel O'Connor, Dr. P. G. Sarang, Alex Toussaint, Sameer Tyagi, Gary Watson, Mark Wilcox, Alan Williamson. *Programación Java Server con J2EE Edición 1.3*, Anaya Multimedia 2002.
- [9] David Pitts, y Bill Ball. *Red Hat Linux 6 a fondo*, Anaya Multimedia 1999.
- [10] Paul DuBois. *MySQL*, Prentice Hall 2001.

- [11] H. M. Deitel, P. J. Deitel, and S. E. Santry. *Advanced Java 2 Platform: How to Program*, Prentice Hall 2002.
- [12] Ralph P. Grimaldi. *Matemática Discreta y Combinatoria*, Addison Wesley Longman 1998.
- [13] Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Fundamentos de Bases de Datos*, Mc Graw Hill 2002.
- [14] Ramesh Nagappan, Robert Skoczylas, y Rima Patel Sriganesh. *Developing Java Web Services*, Wiley 2003.
- [15] Ed Roman, Scott Ambler, Tyler Jewell. *Mastering Enterprise JavaBeans*, John Wiley & Sons, Inc 2002.
- [16] Andrew W. Appel and Jens Palsberg. *Modern Compiler Implementation in Java*, Cambridge University Press 2002.
- [17] Ray Lai. *J2EE Platform Web Services*, Prentice Hall & Sun Microsystems Press 2004.
- [18] I. Jacobson, Booch, J. Rumbaugh. *Unified Modeling Language User Guide*, Addison Wesley 1998.
- [19] Ministerio De Educación, Vice Ministerio de Gestión Pedagógica. *Proyecto Huascarán Reglamento de Organización y Funciones del Proyecto Huascarán*. Decreto Supremo N 015-2005-ED.
- [20] Madeleine Zúñiga Castillo y Juan Ansión Mallet. *Interculturalidad y educación en el Perú*, <http://macareo.pucp.edu.pe/jansion/Interculturalidad/Libro/IC-Intr.htm>.

- [21] Ministerio De Educación. *Proyecto Huascarán Orientaciones para el Desarrollo de las Actividades Pedagógicas en las Instituciones Educativas atendidas por el Proyecto Huascarán*, Directiva N 9 - 2006.
- [22] Enciclopedia Libre. *Wikipedia*, <http://es.wikipedia.org/wiki/Portada>.
- [23] Sheng Liang. *The Java Native Interface*, Addison-Wesley 1999.
- [24] Instituto Nacional de Estadística e Informática. *Norma Técnica para Elaboración de Cuadros Estadísticos*, <http://www.inei.gob.pe> .
- [25] Francisco Sagasti. *El Sistema de Innovación Tecnológica en el Perú*, Foro Nacional de Agenda Perú, 2003.
- [26] UNI, Facultad de Ingeniería Industrial y de Sistemas, Instituto de Investigación. Proyecto de Investigación: *Web Services y su Aplicación en la Educación*, <http://www.fiis.uni.edu.pe/iifiis/equipos/webservices.html>. Jefe de Proyecto: Aland Bravo Vecorena.
- [27] Empresa de video caricaturas de Cartoon Network. *El Laboratorio de Dexter*, <http://www.cartoonnetwork.com.mx/watch/tvshows/dexter/index.html>.
- [28] LTSC del Instituto de Ingenieros Electricistas y Electronicos IEEE. *The IEEE Learning Technology Standards Committee*, <http://ieeeltsc.org>.
- [29] Red Inteligente NGN TELLIN UNICA de Huawei Technologies. *Huawei Technology Standards Committee*, <http://huawei.com>