



UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGÍA
COORDINACIÓN DE POSTGRADO
Maestría en Ciencias de la Computación



**PROPUESTA DE INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN
BASADOS EN ROA. CASO: UNEXPO**

JOSÉ VALENTÍN LÓPEZ MONTILLA

Barquisimeto, 2011



UNIVERSIDAD CENTROCIDENTAL
"LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGÍA
COORDINACIÓN DE POSTGRADO
Maestría en Ciencias de la Computación



**PROPUESTA DE INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN
BASADOS EN ROA. CASO: UNEXPO**

Trabajo de grado presentado para optar al grado de
Magister Scientiarum
en Ciencias de la Computación
Mención Ingeniería de Software

Por: JOSÉ VALENTÍN LÓPEZ MONTILLA

Barquisimeto, 2011

**PROPUESTA DE INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN
BASADOS EN ROA. CASO: UNEXPO**

Por: JOSÉ VALENTÍN LÓPEZ MONTILLA

Trabajo de grado aprobado

(Jurado 1)

(Jurado 2)

(Jurado 3)

Barquisimeto, ____ de _____ de 2011

INDICE GENERAL

	PAG.
DEDICATORIA	vi
AGRADECIMIENTO	vii
LISTA DE FIGURAS	viii
LISTA DE CUADROS	ix
RESUMEN	x
INTRODUCCIÓN	12
CAPITULO	
I	EL PROBLEMA 14
	Planteamiento del Problema 14
	Objetivos de la Investigación 20
	Objetivo General 20
	Objetivos Específicos 20
	Justificación e Importancia 21
	Alcance y Limitaciones 22
II	MARCO TEORICO 24
	Antecedentes de la Investigación 24
	Bases Teóricas 29
III	MARCO METODOLOGICO 61
	Naturaleza de la Investigación 61
	Fases del Estudio 62
	Fase I: Diagnóstica 62
	Fase 2: Estudio de Factibilidad 65
	Factibilidad Técnica 65
	Factibilidad Institucional 66
	Factibilidad Operativa 66

	Factibilidad Económica	67
IV	FASE III: DISEÑO DEL PROYECTO	69
V	CONCLUSIONES Y RECOMENDACIONES	94
	BIBLIOGRAFIA	97
	ANEXOS	101

DEDICATORIA

A Dios, por darme la ecuanimidad y paciencia necesaria para vencer cada obstáculo.

A mis amados padres: Carmen, regalándome como de costumbre su gran amor, y enseñándome al mismo tiempo que el verdadero amor no conoce ni de distancia, ni de dimensión. Ángel, icono de trabajo, constancia y rectitud; a ellos, que recuerdan mis preferencias, me enseñan con su ejemplo, escuchan mis historias y me regalan su tiempo.

A mi adorada Enny, por esa compañía llena de tanta dulzura y bondad, que me ha permitido crecer como ser humano.

A mis hermanos Isidro, Isrrael y Adrian, nuevamente por ese gran respaldo, credibilidad y confianza.

A mi querida sobrina Sofia, que con su picardía y suprema inteligencia, me llena de alegría e inspira al mismo tiempo.

A Mariela de mi corazón, en primer lugar por permitirme el honor de estar a su lado, y en segundo lugar por esa llave maestra, basada de su amable compañía y generoso amor, que una vez más me dan la fortaleza necesaria para no declinar y seguir siempre adelante.

AGRADECIMIENTO

A Dios, porque simplemente no existen palabras para plasmar el infinito agradecimiento por hacer todo esto posible.

A la UNEXPO, por la gran oportunidad de permitirme desarrollar este proyecto.

A mi tutor Edgar González, que como de costumbre demuestra en todo momento su gran disposición y profesionalismo.

A mi estimada tutora Maiba Rodríguez, por la gran colaboración aportada de su talento, así como también de su calidad humana e inimitable conducción.

Al Econ. Ender Macias, por su capacidad humana y por su gran respaldo, además de estar siempre dispuesto a apoyar los proyectos dentro de la UNEXPO.

A mis estimados compañeros de trabajo Maribal y Leiban, por creer y apoyar este proyecto.

A todas aquellas personas que de una u otra forma realizaron su aporte significativo a este proyecto.

LISTA DE FIGURAS

Figura		pp.
1	Sistemas de Información dentro de la UNEXPO.	17
2	Proceso actual de la Unidad en Estudio.	18
3	Putting the WS-* vs. REST decision in the context of application integration styles.	26
4	A collaborative Bite flow. Bite composes a flow for releasing software to the public by combining unstructured collaboration activities, back-end services, and user interactions with Web forms.	27
5	Reference architecture for the BPEL for REST extensions.	29
6	Disciplinas del proceso InDoCaS.	36
7	Análisis del dominio InDoCaS.	38
8	InDoCaS: Diagrama de actividades de la Disciplina de análisis del dominio.	39
9	InDoCaS: Diagrama de actividades subproceso para Síntesis arquitectural.	40
10	InDoCaS: la disciplina de Diseño del dominio, subproceso para Evaluación arquitectural del Dominio.	41
11	Dependencias encontradas en la evaluación arquitectural del dominio.	42
12	Un motor de búsqueda sin estado.	50
13	Primer plano de una página de resultados de búsqueda de Google.	50
14	El camino desde la raíz de servicio a un restaurante cerca del Monte Rushmore.	53
15	Esquema de componentes de SOA.	56
16	Capas y componentes de una SOA.	58
17	Un Simple escenario de ESB.	58

18	Vista de Alto Nivel de un ESB.	59
19	Organización de MOM.	60
20	Colas de mensajes MOM.	61
21	Arquitectura de la Propuesta.	90
22	Proceso Propuesta de Integración	91
23	Opciones del módulo Integrador del Sistema IPRESTO.	91
24	Pantalla Sistema IPRESTO, donde se define el rango de fecha para generar el recurso de solicitud de información.	92
25	Pantalla Sistema IPRESTO, donde se captura el recurso con la información generada por la aplicación SAI.	92
26	Pantalla Sistema IPRESTO, donde se configura la ubicación de los recursos.	93
27	Pantalla Sistema SAI, Menú de opciones.	93
28	Pantalla Sistema SAI, donde se configura la ubicación de los recursos.	94

LISTA DE TABLAS Y CUADROS

CUADRO		pp.
1	Sistemas de información en la UNEXPO (Vicerrectorado Barquisimeto).	64
2	Comparativo de las Arquitecturas ROA y SOA	71
3	Distribución de las funcionalidades de cada aplicación.	85
4	URIs y enlace de cada Recurso.	86
5	Recursos, métodos HTTP y representaciones.	87
TABLA		
1	Esquema de Actividad InDoCaS.	36
2	Esquema de Artefacto InDoCaS.	37
3	Resumen de Actividades y Artefactos de InDoCaS.	42
4	Actividades del modelo de procesos InDoCaS aplicados al dominio.	74
5	Lista de requisitos funcionales del sistema administrativo SAI.	77
6	Lista de requisitos no funcionales del sistema administrativo SAI.	77
7	Lista de requisitos funcionales del sistema administrativo IPRESTO.	78
8	Lista de requisitos no funcionales del sistema administrativo IPRESTO.	79
9	Conjunto de características del sistema administrativo SAI.	80
10	Conjunto de características del sistema administrativo IPRESTO.	80
11	Conjunto de puntos de variación del sistema administrativo SAI.	81
12	Conjunto de puntos de variación del sistema administrativo IPRESTO.	81
13	Conjunto minimal de requisitos funcionales y no funcionales del sistema administrativo SAI.	82
14	Conjunto minimal de requisitos funcionales y no funcionales del sistema administrativo IPRESTO.	83

**UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGÍA
COORDINACIÓN DE POSTGRADO
Maestría en Ciencias de la Computación**

**PROPUESTA DE INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN
BASADOS EN ROA. CASO: UNEXPO**

AUTOR: JOSÉ VALENTÍN LÓPEZ M.
TUTOR: EDGAR GONZALEZ

RESUMEN

Propósito- Este trabajo de grado tiene por finalidad el desarrollo de una propuesta de integración de los sistemas de información para la UNEXPO basada en ROA (Resource Oriented Architecture - Arquitectura Orientada a Recursos). **Diseño / metodología / enfoque-** El tipo de investigación se presenta dentro de una modalidad de proyecto factible, enmarcado dentro de una investigación de carácter descriptivo y sustentado en la revisión bibliográfica y documental, respondiendo al paradigma cuantitativo. Para desarrollar dicha investigación se basó en tres fases: Fase I (Caracterización de los Sistemas de Información), Fase II (Estudio de Factibilidad) y Fase III (Diseño del Proyecto). Para concebir la propuesta de integración de sistemas de información enfocado en una ROA, se sigue el procedimiento de RESTful, además de apoyarse con actividades de la ingeniería del dominio, para lo cual se utilizó el método InDoCaS y lograr de esta manera la obtención del análisis del dominio, permitiendo la combinación de estos métodos abordar con mayor exactitud los objetivos deseados para el desarrollo de la presente propuesta. **Conclusiones-** Fue importante la incorporación de actividades de la ingeniería del dominio, para lo que se utilizó el método InDoCaS, ya que permitió obtener el análisis del dominio y así dar paso a la aplicación del procedimiento RESTful. Al final se concluye que los beneficios de utilizar el procedimientos RESTful, que combina criterios de REST, las ventajas de HTTP y el uso de las URIs, impactarán positivamente a la institución, dado que los sistemas están relacionados con la parte operativa de la organización y a la medida que los sistemas estén trabajando de forma integrada, mayores serán los beneficios que se obtienen.

Descriptores: integración de aplicaciones, RESTful, ROA, InDoCaS.

INTRODUCCIÓN

La evolución que experimentan los sistemas de información en las organizaciones, le da cabida a la apertura de los sistemas transaccionales, para que posteriormente se introduzcan los sistemas de apoyo a las decisiones, y por último, emerjan los sistemas estratégicos que dan forma a la estructura competitiva de una empresa o institución.

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la escuela de negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática. La función de la informática en las organizaciones evoluciona a través de ciertas etapas de crecimiento, entre las cuales se mencionan algunas: el inicio con la adquisición de la primera computadora que normalmente se justifica por el ahorro de mano de obra, exceso de papeles y las aplicaciones típicas o sistemas transaccionales tales como nóminas o contabilidad, entre otros.

Pero al paso de los años cada vez es más crucial, para experimentar la evolución tecnológica, mantener tanto los sistemas de información como los procesos de una forma integrada, por lo que prácticamente las organizaciones se ven obligadas a la búsqueda de soluciones o nuevas estrategias que le permitan mantener toda su información consolidada, contenida en todos sus sistemas de información, y poder dar respuesta de una manera mucho más rápida y eficiente.

De acuerdo a lo comentado por Doug Kennedy, vicepresidente sénior de Alianzas y Canales Internacionales de Oracle, la Integración de Aplicaciones es un requisito fundamental para la soluciones de negocios de los clientes, por lo que no es descabellado pensar que para lograr la máxima productividad de soluciones incluso heterogéneas, así como la incorporación de nuevos desarrollos, la Integración de Aplicaciones, juega un papel muy importante dentro de una organización. Todo esto lleva a que los mecanismos de integración deban sostenerse bajo una arquitectura que permita a los procesos adaptarse fácilmente para responder de forma más eficiente a las necesidades de la institución.

Hoy en día instituciones como la UNEXPO (Universidad Nacional Experimental Politécnica Antonio José de Sucre) tienen la necesidad de adaptarse a constantes cambios exigidos por entes gubernamentales, como mecanismos de control, y por tanto, deben ser capaces de adaptar sus procesos administrativos a unas condiciones dinámicas que permitan mantener, mejorar y solapar todas las solicitudes en tiempos de respuestas oportunos.

Por lo tanto, se pretende con el desarrollo de la presente investigación, proponer un mecanismo de integración de sistemas de información, que sea eficiente y mantenible, para la UNEXPO ubicada en la ciudad de Barquisimeto, Estado Lara, que le permita enfrentar con éxito los desafíos que representa la integración de sus sistemas de información a un mediano plazo.

El trabajo desarrollado se dividió en cinco capítulos a saber:

Capítulo I: Aquí se hace hincapié en el asunto a investigar, planteamiento del mismo, objetivo general y objetivos específicos del proyecto, justificación y alcance.

Capítulo II: Se desarrolló todo lo relacionado con la reseña histórica, estructura organizativa de la institución, antecedentes de la investigación, bases teóricas y legales del tema tratado.

Capítulo III: Se hace referencia a la metodología empleada, el tipo de investigación y la metodología utilizada para el diseño del proyecto propuesto. Se desarrolla la fase I y II del proyecto factible, describiéndose la unidad de estudio, población y muestra, además se presenta el diseño de la investigación, así como el estudio de las factibilidades.

Capítulo IV: Se desarrolla la fase III del proyecto factible, en donde se presenta de forma detallada el diseño de la propuesta que se plantea.

Capítulo V: En este capítulo se abordan las conclusiones y recomendaciones.

CAPÍTULO I

EL PROBLEMA

Planteamiento del Problema

Hoy en día el vertiginoso dinamismo donde se desenvuelve o hacen vida las organizaciones, impulsa a la necesidad de ajustarse a los distintos cambios, como consecuencia de las exigencias a las cuales son sometidas, donde mantener sincronizadas he integradas tanto sus actividades ordinarias, tecnologías de información y comunicaciones, son indicadores para el desarrollo exitoso de cualquier actividad operativa de una organización.

Vázquez, O. (2005), hace referencia a la integración de aplicaciones como el proceso de lograr que aplicaciones independientes trabajen juntas para producir un conjunto unificado de funcionalidades. Si las necesidades de integración fueran siempre iguales, habría solamente un estilo de integración. Como todo proyecto tecnológico complejo, la integración de aplicaciones implica una serie de consideraciones y de consecuencias que se deben tener en cuenta para cualquier oportunidad de integración.

El primer criterio es la integración de aplicaciones en sí mismo. Si es posible desarrollar una aplicación que sea independiente, que no necesite colaborar con ningún otro sistema, entonces no es necesario realizar una integración de aplicaciones. Sin embargo, en la vida real, hasta una pequeña empresa tiene múltiples aplicaciones que necesitan trabajar juntas para proporcionar mayores beneficios a los directivos, empleados y a los clientes de la empresa. El término de integración de aplicaciones se ha vuelto popular con el crecimiento de la importancia de la integración, y con proyectos de integración más extensos.

La integración de aplicaciones no es una fórmula, ni una solución que se pueda adquirir, e inmediatamente resolver los problemas de integración existentes en la empresa, en cambio, es el proceso continuo de crear una infraestructura flexible y estandarizada que permita que los nuevos sistemas y procesos del negocio puedan ser integrados fácilmente y comunicarse en tiempo real con el resto de las aplicaciones de la empresa.

En un contexto más técnico, la integración de aplicaciones es el proceso en el que aplicaciones heterogéneas, funciones y datos son integrados, para poder permitir el uso compartido de datos y la integración de procesos de negocios a lo largo de todas las aplicaciones. El propósito es lograr este nivel de integración sin grandes cambios a las aplicaciones existentes y bases de datos, usando métodos eficientes que sean efectivos en cuanto a costo y tiempo se refiere.

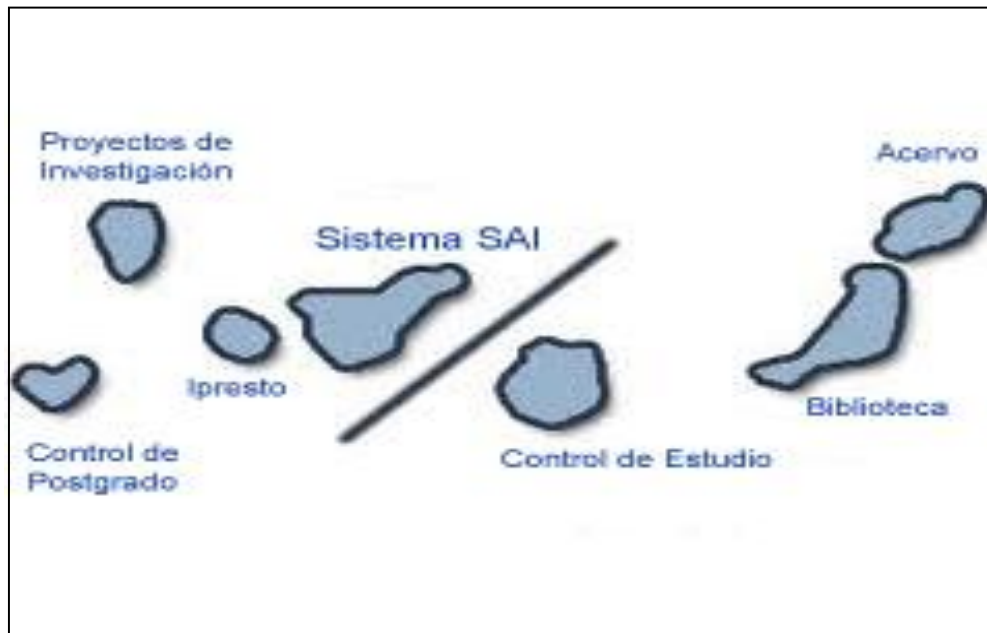
En consecuencia, las organizaciones educativas no escapan de requerir de la integración de sus sistemas de información para ampliar la efectividad en sus procesos, y de esta manera lograr que sus aplicaciones trabajen de forma cohesionadas para obtener mayores beneficios, además se sumarle un valor agregado a la toma de decisiones. Bajo este enfoque se presenta la Universidad Nacional Experimental Politécnica “Antonio José de Sucre” (UNEXPO), que aunque cuenta con sistemas de información con buenas plataformas, se ha observado ausencia de cooperación entre sus aplicaciones que pueden evitar una inversión tanto de tiempo como trabajo adicional.

Es importante resaltar que en la actualidad, la implantación de sistema de información que sirva de soporte para la realización de una administración eficiente, ha adquirido un auge significativo en el mercado empresarial, ya que las instituciones u organizaciones buscan mantener sus sistemas de información trabajando de forma eficiente e integrados, sacando el mayor provecho posible de ellos, en pro de maximizar sus beneficios a través de la minimización de sus costos.

A la realidad antes mencionada no escapa la UNEXPO muy específicamente el caso de estudio el Vicerrectorado Barquisimeto que según información recopilada en la Oficina Regional de Tecnología y Servicio de Información (ORTSI) (s/f),

alberga un conjunto de aplicaciones de las que se han ido haciendo acopio a lo largo de los años y de acuerdo al desarrollo de las necesidades informáticas, que trabajan de forma separada, tal como se muestra en la siguiente figura.

Figura 1. Sistemas de Información dentro de la Unexpo.



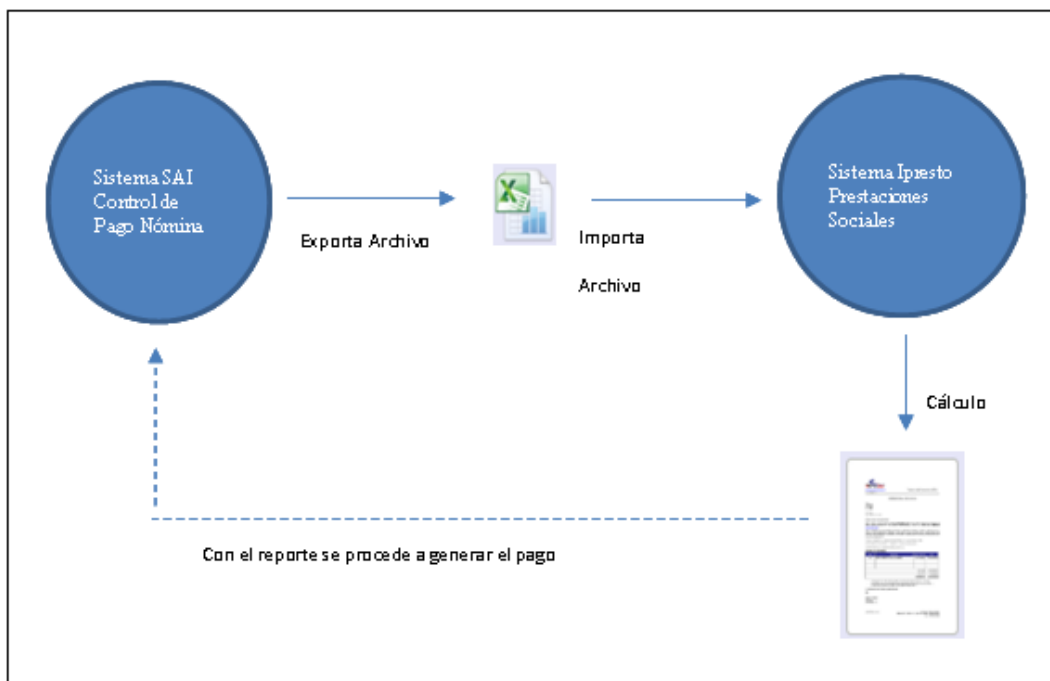
Fuente: Autor de la Investigación.

Todo ello, ha derivado a que se originen islas de información, con los problemas asociados a las mismas, y se adopten mecanismos manuales, pero necesarios, como se presenta en el caso de estudio de los sistemas: sistema administrativo IPRESTO en donde se realiza el control y cálculo de los intereses de las prestaciones sociales, necesitando este insumo de datos (Anticipos de Prestaciones Sociales, Sueldo Integral, entre otros) del sistema administrativo SAI para poder llevar a cabo su finalidad principal como lo es el cálculo de los intereses sobre las prestaciones sociales, así como otras exigencia como por ejemplo, dar respuesta a las solicitudes de la Oficina de Planificación del Sector Universitario (OPSU). Por otro lado el sistema administrativo SAI con el cual se controla el pago de nómina, requiere el resultado de los cálculos generados por el sistema IPRESTO para poder

realizar las respectivas cancelaciones a los trabajadores.

En la actualidad se utiliza una técnica asincrónica como es la transferencia de la información a través de archivos en Excel, procedimiento por demás delicado, que mantiene cierto margen de error e induce a procesos manuales que generan trabajos adicionales y una inversión de tiempo considerable para llevar adelante las tareas, llevando esto a que estas aplicaciones no trabajen de forma más eficiente. La siguiente figura muestra el proceso necesario actual de transferencia de información de los sistemas del caso de estudio.

Figura 2. Proceso actual de la Unidad en Estudio.



Fuente: Autor de la Investigación.

El hecho de no existir una forma que permita integrar las aplicaciones que hacen vida dentro de la institución, no le garantiza a la misma acceder o plantearse una reutilización de la información de forma confiable y segura, así como un trabajo más eficiente entre sus sistemas de información. De esta manera dentro de la organización se hace inminente la necesidad de un enfoque de integración de

aplicaciones, que permita y garantice una comunión entre sus sistemas de información que aporte de manera significativa a la mejora de sus procesos administrativos.

En ese sentido, las debilidades de no tener los sistemas de información integrado (unificados), no poseer un plan de integración así como la inexistencia de un mecanismo de integración que sea eficiente y mantenible, plantean en lo sucesivo situaciones que de no preverse conllevarían a una ampliación de los tiempos de respuesta a la medida que se incorporen otras aplicaciones complementarias.

En este orden de ideas el paradigma REST (Representational State Transfer - Transferencia de Estado Representacional), propuesta por Fielding, R.(2000), y enfocado a una ROA (Resource Oriented Architecture - Arquitectura Orientada a Recursos), se ha hecho muy popular porque es un enfoque ligero que no introduce la sobrecarga de información de los mensajes SOAP, y gracias al protocolo HTTP, que permite acceder a los recursos a través de Internet, brinda un gran apoyo a una posible integración de sistemas de información en organizaciones que no cuenten con una infraestructura física adecuada o incluso deterioradas por el tiempo, tal es el caso de UNEXPO, que además de poseer instalaciones físicas muy antiguas presenta el reto de la ausencia de presupuesto no solo para invertir en mejoras de las mismas, si no también en recursos tecnológicos.

Por lo que se pretende con esta investigación dar respuesta a la problemática con una propuesta de integración de los sistemas de información basada en ROA, una estructura que permita incorporar técnicas efectivas, que guíen a impulsar a los sistemas de información a trabajar en forma mucho más eficiente,.

Por consiguiente, el desarrollo de la investigación pretende dar respuestas a las siguientes interrogantes:

- 1.- ¿Cuales son las características de los sistemas de información ubicados en el Vicerrectorado Barquisimeto de la UNEXPO?
- 2.- ¿Están dadas las condiciones para lograr una interrelación entre los sistemas o subsistemas de información existente en el Vicerrectorado Barquisimeto de la UNEXPO?

3.- ¿De qué manera se pudiera establecer una estrategia de integración entre los sistemas o subsistemas de información para lograr la consistencia y mayor aprovechamiento de la información que generan los mismos?

Objetivos de la Investigación

Objetivo General

Proponer una integración de Sistemas de Información basados en ROA. Caso de estudio: Sistemas de información SAI e IPRESTO. UNEXPO Vicerrectorado Barquisimeto.

Objetivos Específicos

Caracterizar los sistemas de información de la UNEXPO Vicerrectorado Barquisimeto.

Determinar las factibilidades operativa, institucional, tecnológica y económica, que indicarán la disponibilidad de los recursos necesarios para llevar adelante la propuesta de integración.

Diseñar un modelo de Integración de Sistemas de Información basados en ROA. Caso de estudio: Sistemas de información SAI e IPRESTO. UNEXPO Vicerrectorado Barquisimeto.

Justificación e Importancia

La tecnología de información (TI) ha venido reformando los procesos administrativos en las empresas de hoy día, situación que permite que los procesos se integren más fácilmente y a la vez sean más eficientes. Sumado a ello, la integración de aplicaciones en la actualidad es de vital importancia en las empresas para reducir los costos de sus actividades, a medida que mejoran la productividad de sus empleados y se integran, de manera más estrecha, con sus clientes, proveedores y socios. Debido al constante avance de la tecnología y su aplicación en los procesos de las organizaciones conllevará a la incorporación de nuevas aplicaciones y por lo tanto, será necesario llevar a cabo soluciones de integración de aplicaciones para garantizar el intercambio entre las nuevas y viejas.

En consideración a lo antes descrito, la UNEXPO como organización educativa requiere dada su envergadura administrativa que sus aplicaciones trabajen de forma más eficiente e integradas, en donde la información contenida en cada una de ellas pueda ser compartidas. Al respecto, Vázquez, O. (2005) expone que la creciente necesidad de los ejecutivos, trabajadores y clientes de acceder en tiempo real a información de la empresa, ha llevado a que sistemas, desarrollados con diferentes tecnologías y que funcionan sobre disímiles plataformas se enlacen con el objetivo de aumentar la productividad, la eficiencia y la satisfacción de los clientes.

En la actualidad la necesidad de tener en cuenta la integración de aplicaciones tiene su origen en la multitud de dispositivos y la amplia variedad de aplicaciones y servicios necesarios para dar cobertura a las necesidades de los usuarios. Por lo que se requiere que el conjunto de aplicaciones con que se cuente interactúen de manera más automáticas y flexibles, esto le da a las empresas la ventaja de compartir la información y disparar los procesos asociados a un evento de una manera más rápida y eficiente.

No obstante, en los actuales momentos, en la UNEXPO la forma en que se emplea la transferencia de información entre las aplicaciones es muy deficiente (manual sin ningún mecanismo de integración). Pero es de hacer notar que la UNEXPO posee recursos tecnológicos que es importante aprovechar y a la que se le

debe dar valor agregado, una integración de sus aplicaciones incrementará la retroalimentación entre sus procesos administrativo y personas que tienen contacto directo con los mismos.

Por lo tanto, el objetivo del presente estudio se basa en una Propuesta de Integración de los Sistemas de Información para la UNEXPO basada en ROA, caso de estudio: Sistemas de información IPRESTO y SAI Vicerrectorado Barquisimeto, el mismo busca maximizar la calidad y efectividad de los sistemas de información bajo los parámetros mundiales que se manejan en la actualidad relacionados con la variable integración.

Estos beneficios también contribuirán con el trabajo a los usuarios, ya que si las aplicaciones trabajan de forma integrada, mejorará significativamente los procesos de la organización, debido a que las actividades y tareas se harán en el tiempo previsto, ayudando a lograr los objetivos planteados y mejorar los servicios prestados.

Alcances y Limitaciones

Habitualmente la reacción ante las novedades suele ser lenta, y en ocasiones tardía e inútil, por lo que las organizaciones e instituciones necesitan disponer de medios para gestionar y dirigir los cambios de una manera sistemática y sistémica. Para lograrlo es de capital importancia que dichas organizaciones dispongan de sistemas para prevenir las situaciones de crisis, que contemplen la creación de planes para el futuro, programas que les ayuden a orientarse en la dirección deseada o adaptarse al cambio no previsto, pero real.

Es decir, es imprescindible que los gerentes de las organizaciones e instituciones entre las cuales se encuentran las de educación superior, analicen y reflexionen acerca de las necesidades de transformar la universidad mediante la orientación de un enfoque gerencial anclado a un adecuado sistema de información, donde la capacidad para conseguir que los procesos sean eficaces depende cada vez más de las mejoras en la funcionalidad de las tecnologías de información, como el groupware (Lotus Notes), la videoconferencia y las intranets que facilitan la

coordinación del equipo y el intercambio de información y conocimiento.

En este orden de ideas, se planteó en la presente investigación la realización de una Propuesta de Integración de los Sistemas de Información basada en ROA: Sistemas de información SAI e IPRESTO, caso de estudio: UNEXPO Vicerrectorado Barquisimeto.

Por lo tanto, el desarrollo de la propuesta responde a las características organizacionales de la UNEXPO vicerrectorado Barquisimeto, no obstante, puede aplicarse en otra organización si la misma responde a características iguales o semejantes, tomando en consideración un diagnóstico previo de dicha organización.

CAPITULO II

MARCO TEORICO

La temática planteada se sustenta en diversos componentes teóricos, en los cuales se hace énfasis en una integración de los Sistemas de Información basada en ROA. En síntesis, se establecen los fundamentos teóricos y los antecedentes de la investigación que tienen como finalidad respaldar la misma y el modelo de propuesta que se generó.

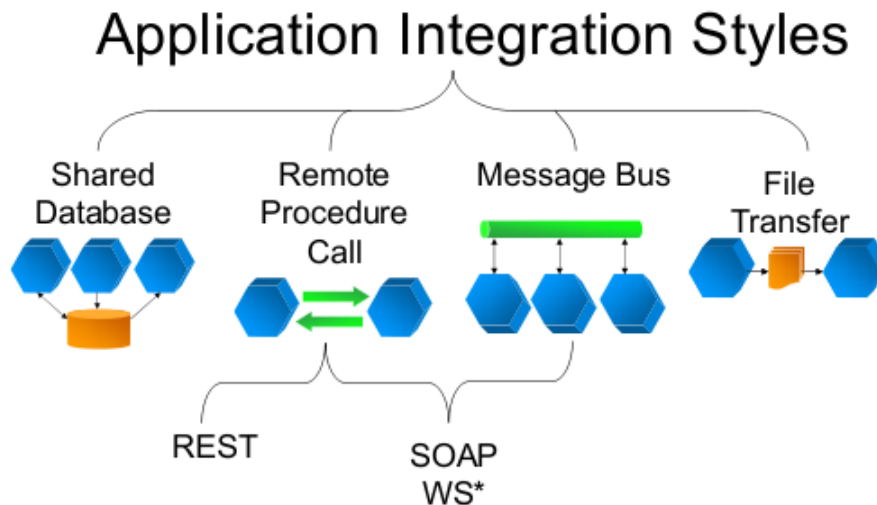
Antecedentes de la Investigación

Lucchi, R. Y Millot, M. (2008) llevan adelante un trabajo muy interesante sobre la captura de los principios de la Arquitectura orientada a Recursos y evalúan la viabilidad, así como las ventajas de la utilización de este enfoque en comparación con Arquitecturas Orientadas a Servicios (SOA). El estudio lo realizaron sobre INSPIRE (European Parliament and European Council, Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 Establishing an Infrastructure for Spatial Information in the European Community. INSPIRE 2007) y de los servicios OGC (Open Geospatial Consortium Inc. [cited 2006 May 17]; Available from: <http://www.opengeospatial.org>.) infraestructuras que actualmente se basan en una arquitectura SOA como de dominio para el análisis comparativo. Además se evalúa la razón de costos y beneficios de implementar ROA a partir de una ya disponible SOA.

El aporte obtenido se puntualiza en la evaluación de los aspectos tecnológicos que podrían ser relevantes para evaluar de forma idónea una solución con ROA, dicha experiencia incorpora puntos importantes para el presente trabajo, sobre la conceptualización de una metodología que guíe o permita concebir a través de ella una ROA.

Pautasso C., Zimmermann O. and Leymann F. (2008), realizan una investigación referida a una comparación de los Servicios RESTful vs los grandes Servicios Web para tomar la decisión correcta de la arquitectura, en donde enfatizan que muchos estilos diferentes se puede utilizar para integrar las aplicaciones empresariales, tan como se muestra en la figura 3.

Figura 3: Putting the WS-* vs. REST decision in the context of application integration styles.



Fuente: Pautasso C., Zimmermann O. and Leymann F. (2008).

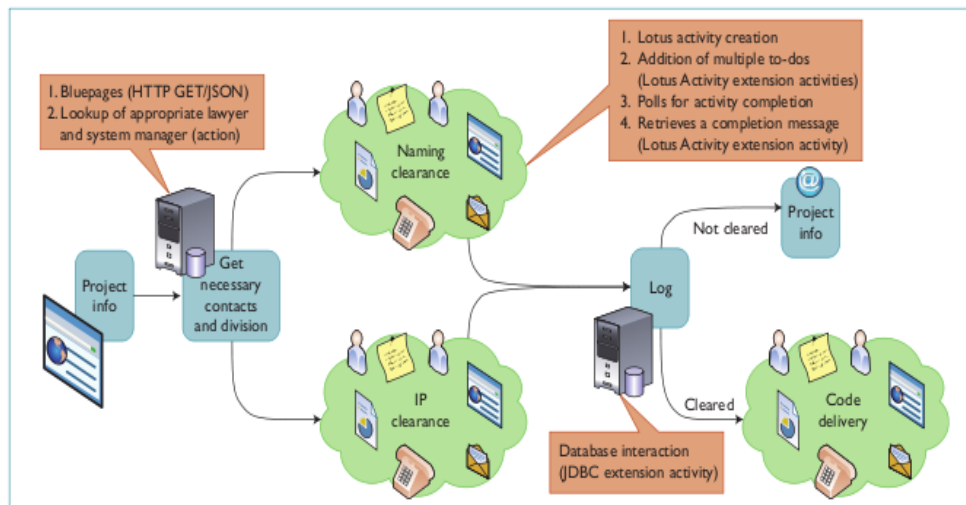
La elección entre confiar en una base de datos compartida, la transferencia de archivos por lotes, la llamada a procedimientos remotos, o el intercambio de mensajes asíncronos en un bus de mensajes, es una decisión importante para la arquitectura, lo que influye en los requerimientos para la plataforma y las propiedades del sistema integrado. En este trabajo investigativo se usaron los principios arquitectónicos y se ilustra las diferencias conceptuales y tecnológicas entre los servicios RESTful y WSDL / SOAP basados en los grandes servicios Web.

El aporte que agrega a la presente investigación es la facilidad del uso del enfoque RESTful y el estudio comparativo de varios estilos diferentes de integrar

aplicaciones empresariales, en donde el tema de costos juega un papel importante.

De igual manera, Rosenberg F., Curbera F., Duftler M., and Khalaf R. (2008), abordan una investigación cuyo objetivo general fue proponer un enfoque ligero construyendo servicios RESTful y Workflows colaborativos. Tradicionalmente, un flujo de trabajo proporciona un comportamiento estructurado, pero los seres humanos en gran medida el uso de aplicaciones Web y, a diferencia de las máquinas, a menudo realizan una tarea en una manera no estructurada. Un ejemplo de un flujo colaborativo que un equipo debe llevar a cabo para liberar una pieza de software para el público se aprecia en la figura 4.

Figura 4: A collaborative Bite flow. Bite composes a flow for releasing software to the public by combining unstructured collaboration activities, back-end services, and user interactions with Web forms.



Fuente: Rosenberg F., Curbera F., Duftler M., and Khalaf R. (2008).

En este estudio los autores indican que la variedad de aplicaciones basadas en Web y su alcance, está trayendo tanto nuevas demandas del usuario como requisitos de los desarrolladores, por lo que las aplicaciones Web necesitan integrar servicios RESTful con sistemas tradicionales de back-end.

El aporte se centra, en la importancia de la utilización del enfoque RESTful en

las entidades principales para la creación de flujos de trabajo de la escala Web. En donde el objetivo fue proporcionar un lenguaje ligero y extensible ceñido a los principios de RESTful para la construcción de los servicios en flujos de trabajo a escala Web.

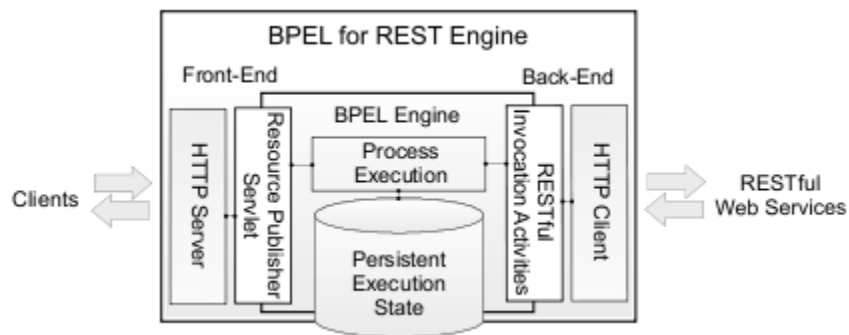
Hamad H., Saad. and Abed R. (2009), proponen en su trabajo una evaluación del rendimiento de los servicio Web RESTful para dispositivos móviles, ya que los dispositivos móviles inteligentes y servicios Web se están convirtiendo en unos temas muy populares. Los dispositivos móviles son físicamente dispositivos limitados, en cuanto a la velocidad de procesador, memoria, batería y la conexión inalámbrica, lo que implica factores que se deben tomar en cuenta en la aplicación de servicios web para dispositivos móviles.

Los resultados de las evaluaciones de rendimiento muestran las ventajas de utilizar servicios web RESTful sobre los servicios web convencionales, las ventajas incluyen menos tamaño de los mensajes y el tiempo de respuesta. Por lo tanto, RESTful ofrece una solución perfectamente válida para la mayoría de las implementaciones, con mayor flexibilidad y menores gastos.

En este estudio, el aporte obtenido se encuentra enmarcado en la factibilidad de implementar RESTful en condiciones limitadas, enfatizando que este enfoque brinda rendimientos a bajos costos.

Por su parte, Pautasso C. (2009), basó su investigación en la construcción de servicios web RESTful con BPEL para REST. En este trabajo se presenta el objetivo de BPEL para las extensiones de REST. En primer lugar, permitir la construcción de servicios Web RESTful y servicios tradicionales de internet desde el mismo proceso de lenguaje orientado a la composición de servicios. En segundo lugar, se muestra cómo publicar un proceso BPEL como un servicio web RESTful, mediante la exposición de las partes seleccionadas de su estado de ejecución mediante las interacciones primitivas de REST, como se muestra en la figura 5.

Figura 5: Reference architecture for the BPEL for REST extensions.



Fuente: Pautasso C. (2009).

El aporte para esta investigación radica en el hecho de ver como REST gana tracción y el número de publicación con servicios web RESTful crece, también es importante observar como el enfoque RESTful es incorporado en combinación y acople a otros enfoques como alternativa ligera en la búsqueda de una mayor eficiencia y rendimiento de los procesos existentes.

Zhao H. and Doshi P. (2009), proponen una búsqueda de automatización con la construcción de servicios Web RESTful como una opción que emerge para las empresas líderes de internet para exponer los datos y recursos internos, por lo que los servicios Web RESTful está trayendo una especial atención creciente en la industria. En este estudio se da una introducción de una descripción formal del problema en la creación de servicios web RESTful, se analizan las diferencias y los desafíos involucrados en este problema, además se propone un modelo formal para la clasificación y descripción de los servicios web RESTful.

Debido a la naturaleza declarativa y otras características de los servicios web RESTful, como ser de peso ligero, fácilmente accesible y escalable, se argumentó que los servicios web RESTful tienen algunas ventajas únicas sobre la construcción de servicios web tradicionales. Los autores sostienen que aunque los servicios web RESTful han sido ampliamente utilizados en la construcción de aplicaciones mashup, los mismos creen que los servicios web RESTful jugará un papel cada vez más importante en el contexto de SOA, donde WSDL / SOAP WSS son dominantes.

El aporte obtenido para el presente trabajo se fundamenta en la experiencia para la clasificación e identificación de los recursos en un enfoque RESTful.

Bases Teóricas

Es importante que en toda investigación exista un fundamento teórico y el contenido técnico, ya que estas bases permiten una descripción más amplia del problema, al integrar la teoría con toda una variedad de elementos involucrados en el estudio.

Al respecto, Acevedo (2001) expresa que es el conjunto de proposiciones teóricas interrelacionadas, que fundamentan y explican aspectos significativos del tema o problema en estudio, y lo sitúan dentro de un área específica del conocimiento.

En virtud a lo expuesto, se puede agregar que las bases teóricas ayudan en la detección y organización de los elementos que contienen la descripción del problema, la discusión de resultados y la propuesta formulada, de forma tal que puedan ser manejados y finalmente convertidos en acciones concretas que permitan obtener la totalidad de los resultados esperados. Como fundamento teórico para la presente investigación es ineludible abordar la teoría general de los sistemas, como conocimiento sólido y central.

Teoría General de los Sistemas

Se puede definir, según Bertalanffy (1986), como “una ciencia general de la totalidad, concepto tenido hasta hace poco por vago, nebulosos y semimetafísico. En forma elaborada sería una disciplina lógico-matemática, puramente formal en si misma pero aplicable a las varias ciencias empíricas. Para ciencias que se ocupan de todos organizados, tendría significado análoga a la que disfrutó la teoría de la probabilidad para ciencias que las ven con acontecimientos aleatorios”.

Además refiere a la teoría general de los sistema como un instrumento útil al dar, por una parte, modelos utilizables y transferibles entre diferentes campos, y

evitar, por otra parte, vagas analogías que a menudo han perjudicado el proceso en dichos campos. Bertalanffy (1986), indica que las metas principales de la teoría general de los sistemas son:

1. Una tendencia general hacia la integración en las varias ciencias, naturales y sociales.
2. La integración parece girar en torno a una teoría general de los sistemas.
3. La teoría general de los sistemas pudiera ser un recurso importante para buscar una teoría exacta en los campos no físicos de la ciencia.
4. La teoría general de los sistemas nos acerca a la meta de la unidad de la ciencia, al elaborar principios unificadores que corren verticalmente por el universo de las ciencias.

Un elemento que juega un papel muy importante incluso dentro de una teoría, es la información, la palabra información es tan masivamente utilizada en destinas disciplinas, ámbitos y momentos que su significado se vuelve algo difuso.

Información

En la actualidad la necesidad de producir información está creciendo cada vez más. Las personas se involucran y dependen día a día más de la información, sobre todo los que tienen a su cargo responsabilidades de gerenciar organizaciones. En una definición algo general, se podría decir que la información es una parte fundamental y necesaria en todo proceso comunicativo en cuanto que es significada por quien la recibe, y como concepto existe en la naturaleza y en la cultura y es trasformada por esta misma cultura que la produce socialmente o la toma de la naturaleza misma. Según el Diccionario de la lengua española - Vigésima segunda edición, define la información como: “Comunicación o adquisición de conocimientos que permiten ampliar o precisar los que se poseen sobre una materia determinada.”

Los datos se procesan mediante modelos para crear información, el receptor recibe la información, luego toma la decisión y actúa, esto genera otras acciones o eventos, los que a su vez crean diversos datos en forma dispersa que se capturan y sirven como entrada, repitiéndose nuevamente el ciclo. Es muy importante acotar que

la información cuando experimenta un crecimiento en volumen, es fundamental y de extrema necesidad tratarla con procedimientos tecnológicos, como lo son los sistemas de información, de manera tal que se pueda lograr un rápido y eficiente proceso de la misma.

Sistemas de Información

No es fácil la definición de sistemas e información, esto debido a la mala interpretación que se le da a los términos “datos” e “información”; de acuerdo a Montilva (1999), define un sistema como “un conjunto de dos o más elementos (conceptos, eventos, objetos, etc.), interrelacionados que conforman un todo”. (p.24). De un modo general se puede decir que un sistema es la unificación de una serie de elementos que trabajan en conjunto para alcanzar un fin que es beneficioso para todos los componentes de dichos sistemas.

Es importante señalar que un sistema de información es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio, sin olvidar que existen distintos tipos de ellos.

Tipos de Sistemas de Información

Sobre la base de su naturaleza y objetivo, los sistemas de información según Senn (1999), se clasifican de la siguiente manera:

Sistemas de información Formal: Basados en un conjunto de normas, estándares y procedimientos que permiten que la información se genere y llegue a quien la necesita en el momento deseado. La información formal puede ser producida por el computador.

Sistemas de Información Informal: Están basados en la comunicación no formalizada, ni predefinida entre las personas de la organización. Este tipo de sistema no tiene estructura y no sigue normas o procedimientos establecidos porque su información puede ser bastante imprecisa, irregular e incierta, imposibilitándose así, el procedimiento automático.

Según Senn (1992), a través de los numerosos estudios realizados a lo largo de

su carrera investigativa los define de la siguiente manera:

Un sistema Hombre – Máquina que posee datos.

Proporciona información que facilite la ejecución de tareas, operaciones y funciones en una organización.

Operación central del sistema de información está constituida por las transacciones y entidades. (p.19)

Los sistemas de información entre sus roles deben cumplir con ciertas funciones que lo caracterizan.

Funciones de un Sistema de Información

Según Senn (1999), los elementos que componen un sistema de información están determinados por el propósito u objetivo del sistema. A pesar de las diferencias que pueden existir entre los distintos sistemas, en todo ello se puede encontrar un conjunto de funciones, de acuerdo al análisis y diseño de sistemas son las que a continuación se describen:

Procesamiento de transacciones: La cual consiste en capturar o recolectar, clasificar, ordenar, resumir y almacenar los datos originados por las transacciones que tiene lugar durante la realización de actividades en la organización.

Definición de archivos: Consiste en almacenar los datos capturados por el procesamiento de transacciones. Los archivos son las unidades básicas de almacenamiento que permiten a la computadora distinguir entre los diversos conjuntos de información. Aunque no siempre es el caso, un archivo se suele encontrar en un formato legible por los usuarios.

Mantenimiento de archivos: Los archivos o bases de datos deben mantenerse actualizados. Las operaciones de mantenimiento son la inserción, modificación y eliminación de datos en los medios de almacenamientos.

Generación de reportes: Se encarga de producir la información requerida y transmitirlas a los puntos o centros de información que la soliciten, es decir, es la producción de la información requerida por la organización como: factura, lista de inventario, entre otros.

Procesamiento de consultas: Las consultas constituyen un medio directo de comunicación hombre/máquina. Esta función es ejecutada generalmente por los subsistemas de administración de datos, los cuales son transformados en información.

Existen también aplicaciones que permiten estandarizar la información a nivel mundial, tal es el caso de UML.

UML

El UML (Unified Modeling Language – Lenguaje unificado de modelado), según Kimmel, P. (2007), es una definición oficial de un lenguaje pictórico con símbolos y relaciones comunes que tienen un significado común. Posiblemente es uno de los lenguajes más recientes inventados por la humanidad, alrededor de 1997.

El UML comprende símbolos y una gramática que define la manera en que se pueden usar estos símbolos, se considera un estándar, y lo que es o no es este lenguaje lo define un consorcio de empresas que constituyen el OMG (Object Management Group - Grupo de Administración de Objetos).

En muchas oportunidades al referirse al UML se emplea el término de software, igualmente que para darle una connotación a las aplicaciones en un sentido general.

Software

Según Pressman, R. (2002). El software de computadora se ha convertido en el alma mater. Es la máquina que conduce a la toma de decisiones comerciales. Sirve de base para la investigación científica moderna y de resolución de problemas de ingeniería. Es el factor clave que diferencia los productos y servicios modernos. Está inmerso en sistemas de todo tipo: de transportes, médicos, de telecomunicaciones, militares, procesos industriales, entretenimientos, productos de oficina, la lista es casi interminable. El software es casi ineludible en un mundo moderno. A medida que nos adentremos en el siglo XXI, será el que nos conduzca a nuevos avances en todo, desde la educación elemental a la ingeniería genética.

Es impresionante la evolución que han ido experimentado los software, a tal

punto de crear necesidades de interacción entre los mismo, lo que ha dando paso a la incorporación de técnicas que permiten dar respuestas a esas necesidades, en este sentido, se puede puntualizar en una técnica que al paso de los días emerge con mucho auge en la actualidad como lo es la Arquitectura Orientada a los Recursos.

Proceso para la Ingeniería del Dominio Basado en Calidad de Software (InDoCaS)

El proceso para la Ingeniería del Dominio basado en Calidad de Software denominado InDoCaS (Canelón, 2010), tiene como objetivo obtener una arquitectura base para una familia, aplicando las actividades correspondiente a las disciplinas de análisis y diseño del dominio. Este proceso desarrollado para la ingeniería de dominio puede ser utilizado en el enfoque de desarrollo de línea de producción de software, el cual puede ser instanciado para un dominio específico y los activos de software producidos pueden ser reutilizados para generar un producto de una familia particular del dominio.

Según (Canelón, 2010), La estructura del proceso está basada en lo siguiente:

- RECLAMO (Requirement Clasification Model): Modelo de clasificación de requisitos propuesto por Chirinos y otros. (Chirinos et al., 2004).
- ISO/IEC 25010: El Estándar Internacional que describe un modelo bipartito para la calidad del producto de software. (ISO/IEC, 2009).
- Un proceso para el análisis del dominio para construir el modelo de calidad propuesto por Losavio y otros. (Losavio et al., 2008).
- FODA (Feature-Oriented Domain Analisis): análisis del dominio orientado a rasgos, desarrollado por el SEI (Software Engineering Institute) para el modelo de variabilidad. (Kang et al., 1990).
- Un proceso general para el diseño arquitectónico del dominio propuesto por Hofmeister y otros. (Hofmeister et al., 2007).
- ADD (Attribute-Driven Design Method): Método de diseño dirigido por atributos,

usado para la formulación de escenarios de calidad. (Bass et al., 2003).

•ATAM (Architecture Tradeoff Analysis Method): Método para elegir una arquitectura para un sistema, utilizado en InDoCaS para la evaluación arquitectural.

Para la presentación de los procesos, se utiliza la notación SPEM 2, en la figura 6 se muestran las disciplinas del proceso InDoCaS, mencionadas anteriormente.

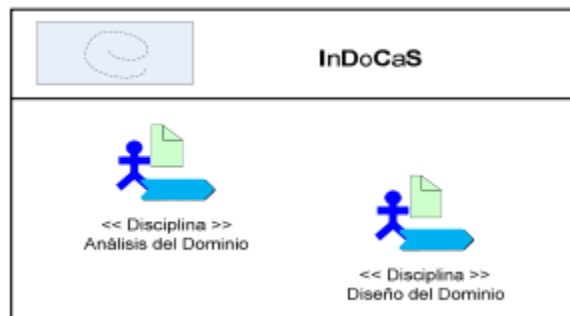


Figura 6. Disciplinas del proceso InDoCaS (Canelón, 2010).

Adicionalmente, InDoCaS define su propia nomenclatura para describir las actividades y artefactos que forman parte de las disciplinas del proceso (Análisis y Diseño), que se muestran a continuación en las tablas 1 y 2:

Tabla 1. Esquema de Actividad InDoCaS.

InDoCaS	
ACTIVIDAD	DESCRIPCIÓN
Nombre de la Actividad:	
Responsable:	
Objetivo:	
Nro. De Identificación	
Tipo documento generado	
Técnica(s) utilizada(s)	
Artefactos de Entrada	
Artefactos de Salida	

Fuente: Canelón, 2010.

Como se observa en la tabla 2, en el caso de las actividades se utiliza: el Nombre de la Actividad, Responsable, Objetivo, un número ordinal para el Nro. De

Identificación, que sirve para futuras referencias, tipo de documento generado (lista, esquema, modelo, diagrama, tabla entre otros), la Técnica(s) Utilizada(s) para la construcción de la actividad, los Artefactos de Entrada que se necesitan para la actividad y los Artefactos de salida producidos en la actividad.

Tabla 2. Esquema de Artefacto InDoCaS.

InDoCaS		
ARTEFACTO		DESCRIPCIÓN
Nombre:		
Constructor:		
Objetivo:		
Nro. De Identificación	ACTIVIDAD	
Nro. De Identificación	ARTEFACTO	
Formato Asociado		

Fuente: Canelón, 2010.

Por su parte en la tabla 2, se muestran los elementos para definir un artefacto InDoCaS a saber: Nombre del artefacto, Constructor responsable, un número ordinal para el Nro. De Identificación ACTIVIDAD para asociarlo a la actividad en referencia, un número ordinal para el Nro. De Identificación ARTEFACTO que hace referencia al artefacto y el formato asociado al tipo de documento generado (lista, esquema, modelo, diagrama entre otros).

A continuación se presentará las descripciones de las disciplinas Análisis del Dominio y Diseño del Dominio de la ingeniería de dominio.

Análisis del Dominio (InDoCaS)

En la fase del análisis del dominio del proceso InDoCaS se definen los aspectos comunes a todas las familias del dominio y los particulares de cada una de ellas. Este subproceso permite la caracterización del dominio, identifica las propiedades de calidad que deben ser garantizadas y define el modelo de calidad asociado al dominio que brinda soporte al resto de las fases. La disciplina de análisis

del dominio está conformada por seis actividades: identificación de requisitos, obtener modelo de similitudes y variabilidad, identificación de propiedades de calidad asociadas al conjunto minimal de requisitos funcionales y no funcionales, obtener el modelo de calidad asociado al dominio, creación de escenarios de calidad del dominio e identificar los estilos arquitecturales para el dominio (Ver figura 7).

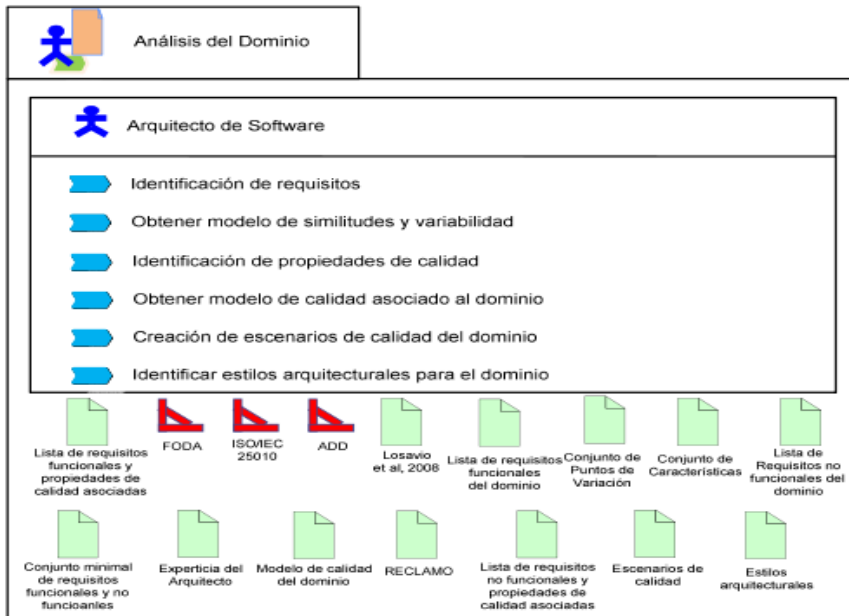


Figura N° 7. Análisis del dominio InDoCaS (Canelón, 2010)

En la figura 8, se muestra el diagrama de actividades de la disciplina análisis del dominio, en el cual se muestran las entradas y las salidas de cada una de las actividades y la secuencia de ejecución, del mismo modo se indica qué técnicas intervienen en cada una de las actividades.

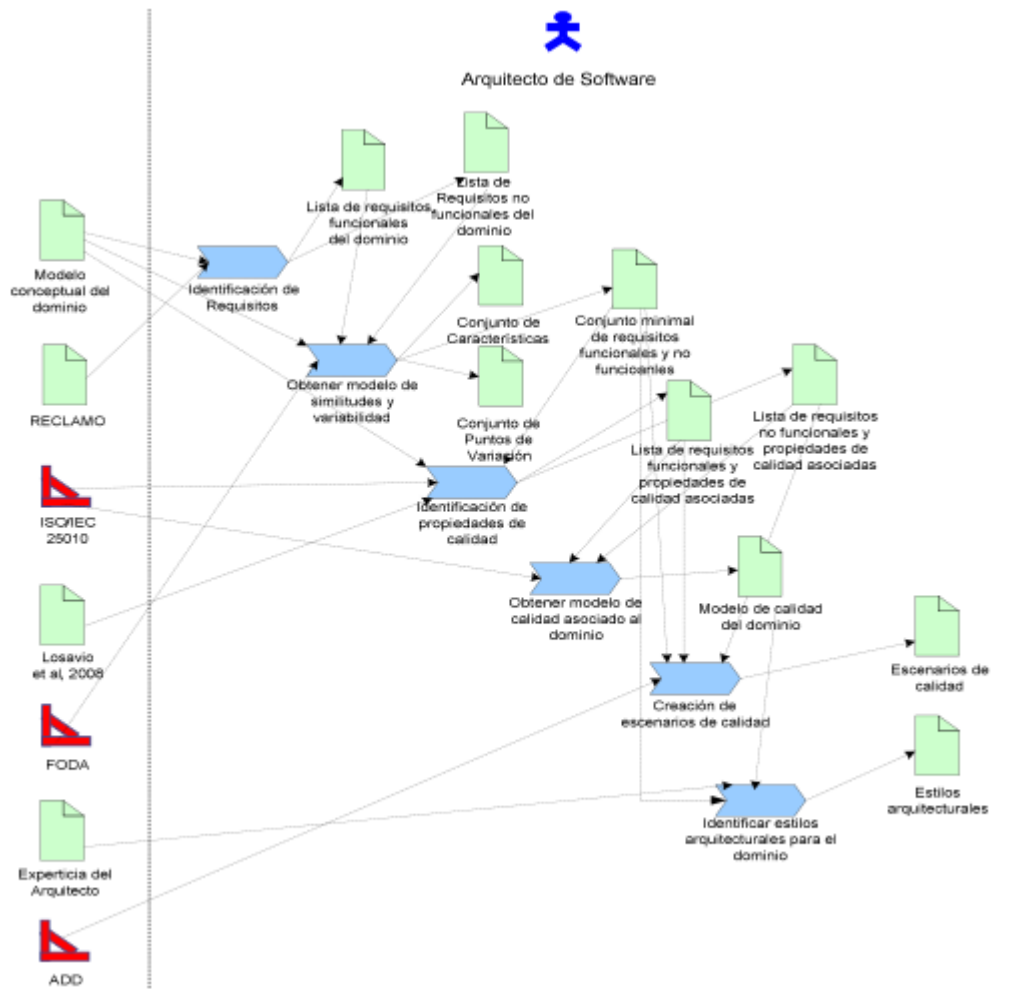


Figura 8. InDoCaS: Diagrama de actividades de la Disciplina de análisis del dominio. **Fuente:** (Canelón, 2010).

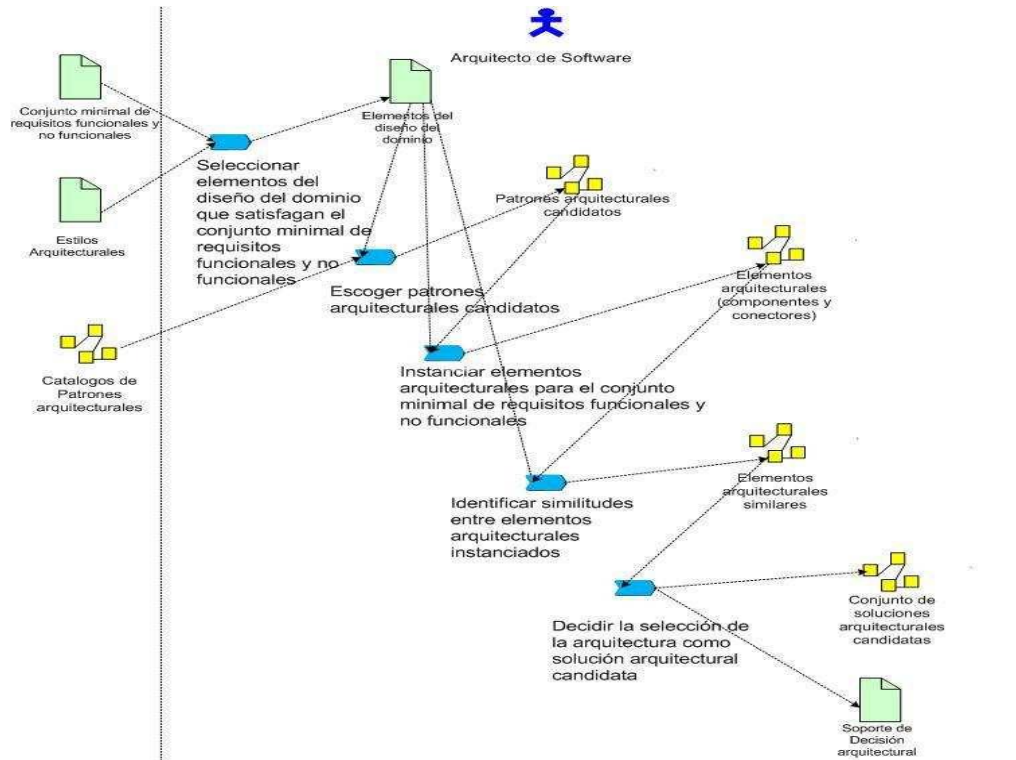
Diseño del Dominio (InDoCaS)

En la fase del diseño del dominio, se especializa las actividades para la síntesis arquitectural y evaluación arquitectural del dominio, proponiendo y adaptando un conjunto de técnicas específicas para la construcción de dichas actividades y artefactos.

La síntesis arquitectural es el núcleo del diseño de la arquitectura, este subproceso propone soluciones consistentes a un conjunto de requisitos arquitecturales, por lo que se traslada del problema a un espacio de solución. Como se

muestra en la figura 9, las entradas a esta fase son los artefactos: Conjunto minimal de requisitos funcionales y no funcionales, Estilos arquitecturales y Catálogos de patrones arquitecturales obtenidos en la disciplina del Análisis del Dominio.

Figura N° 9. InDoCaS: Diagrama de actividades subproceso para Síntesis arquitectural (Canelón, 2010)

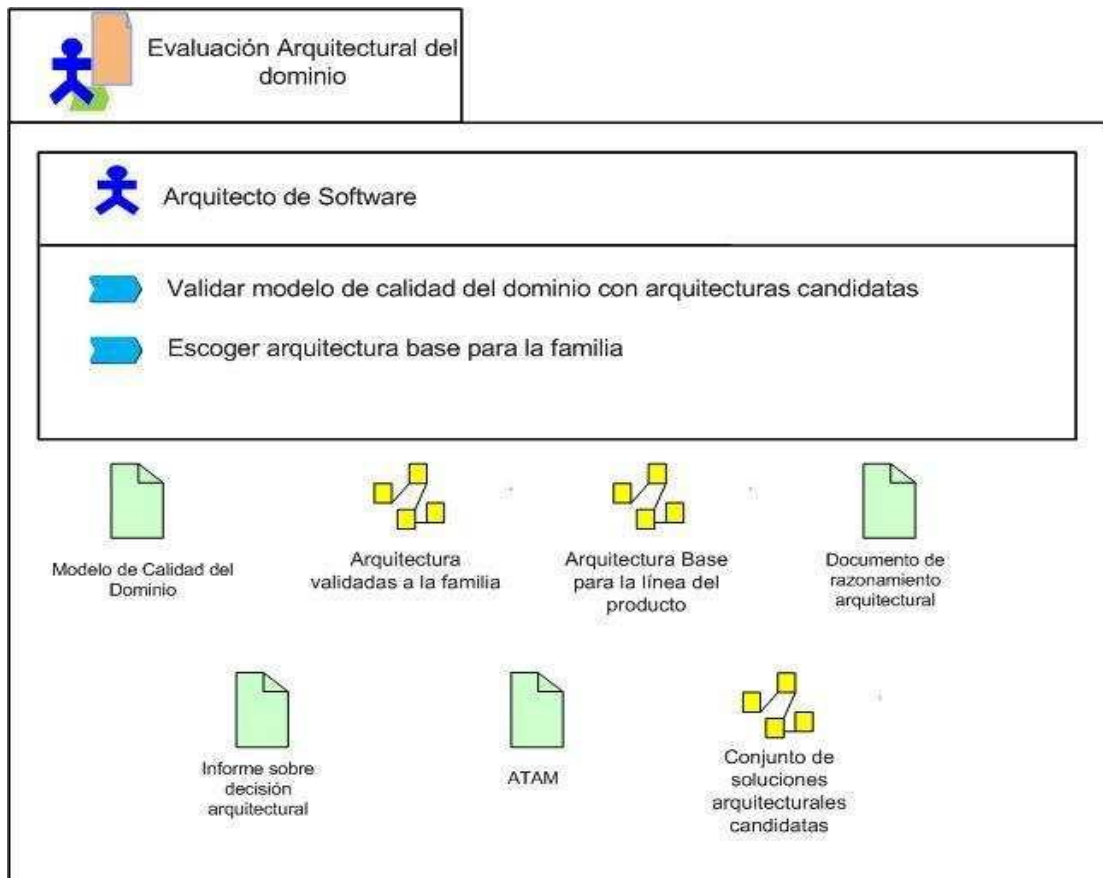


La salida de la síntesis arquitectural identificada como conjunto de soluciones candidatas, son arquitecturas candidatas para la solución al conjunto de requisitos arquitecturales pudiendo ser soluciones alternativas, incluso parciales (partes de arquitectura). En ellas se reflejan las decisiones de diseño acerca de la estructura de software, reflejadas inicialmente en el artefacto estilo arquitectural, obtenido en la disciplina del Análisis del Dominio y en este subproceso se incorpora la información del proceso de decisión.

En las actividades de la evaluación arquitectural del dominio (Ver figura 10),

se busca analizar e identificar riesgos potenciales en su estructura y sus propiedades, que puedan afectar al sistema de software resultante, verificar que los requisitos no funcionales estén presentes en la arquitectura, así como determinar en que grado se satisfacen los atributos de calidad.

Figura N° 10. InDoCaS: la disciplina de Diseño del dominio, subproceso para Evaluación arquitectural del Dominio (Canelón, 2010)



En la figura 11, se presentan las dependencias encontradas en la evaluación arquitectural del dominio.

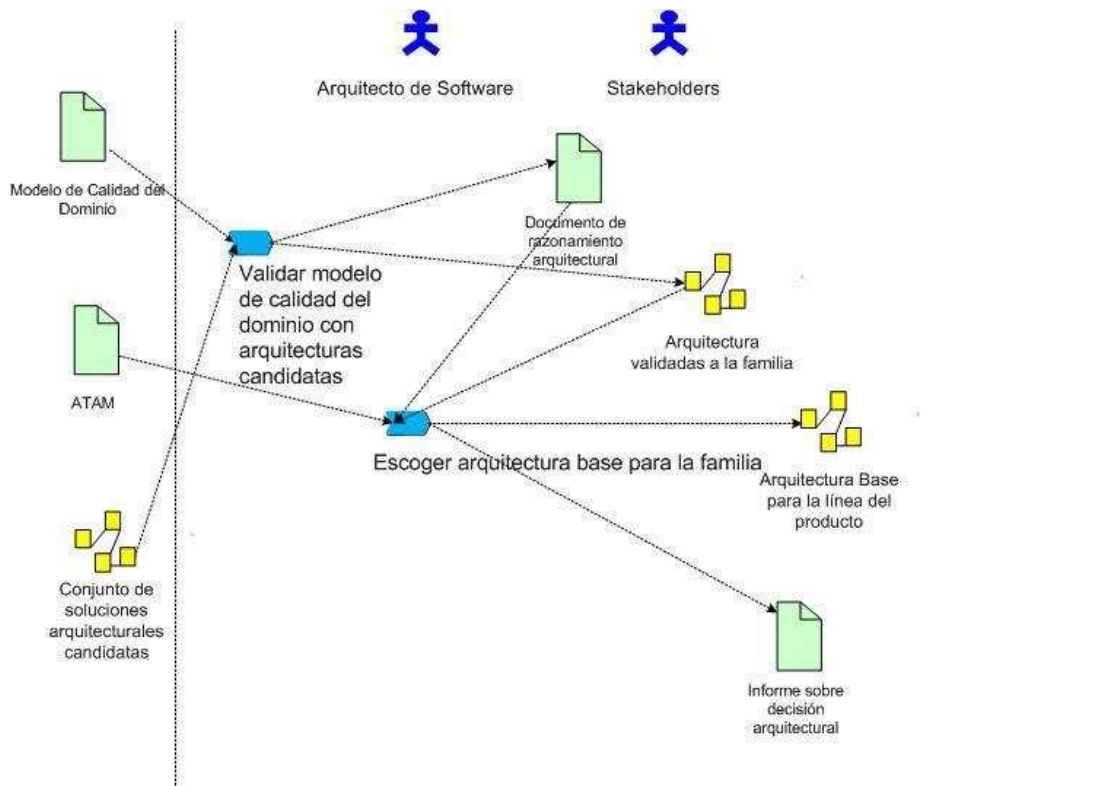


Figura N° 11. InDoCaS: Diagrama de actividades subproceso para Evaluación arquitectural del Dominio (Canelón, 2010)

En la tabla 3, se resumen las actividades de las disciplina análisis del dominio y diseño del dominio, asociando los artefactos de entrada a cada actividad y los artefactos que produce la misma.

Tabla 3. Resumen de Actividades y Artefactos de InDoCaS.

Disciplina	Actividades		Artefactos	
	Nro.	Nombre de la actividad	Artefactos de Entradas	Artefactos de Salida
Análisis del Dominio	A_01	Identificación de Requisitos	- Modelo conceptual del dominio. - Reclamo.	-Lista de requisitos funcionales. -Lista de requisitos no funcionales.
	A_02	Obtener modelo de similitudes y variabilidad	- Modelo conceptual del dominio. -Lista de requisitos funcionales. -Lista de requisitos no funcionales.	-Conjunto de características. -Conjunto minimal de requisitos funcionales y no funcionales.

				- Conjunto de puntos de variación.
	A_03	Identificación de propiedades de calidad	-Modelo conceptual del dominio. -Conjunto minimal de requisitos funcionales y no funcionales. - Losavio et al., 2008.	- Lista de requisitos Funcionales y propiedades de calidad asociadas. - Lista de requisitos no funcionales y propiedades de calidad asociadas.
	A_04	Obtener modelo de calidad asociado al dominio	- Lista de requisitos funcionales y propiedades de calidad asociadas. - Lista de requisitos no funcionales y propiedades de calidad asociadas.	- Modelo de calidad del dominio.
	A_05	Creación de escenarios de calidad del dominio	- Modelo de calidad del dominio como. - Lista de requisitos funcionales y propiedades de calidad asociadas. - Lista de requisitos no funcionales y propiedades de calidad asociadas. - Conjunto minimal de requisitos funcionales y no funcionales.	-Escenarios de calidad.
	A_06	Identificar los estilos arquitecturales para el dominio	- Conjunto minimal de requisitos funcionales y no funcionales. - Modelo de calidad del dominio como.	-Estilos arquitecturales.
Diseño del Dominio Síntesis Arquitectural	A_07	Seleccionar los elementos del diseño del dominio que satisfagan el conjunto minimal de requisitos funcionales y no funcionales	- Conjunto minimal de requisitos funcionales y no funcionales. -Estilos arquitecturales.	-Elementos del diseño del dominio
	A_08	Escoger Patrones arquitecturales candidatos	-Catalogo de Patrones arquitecturales. -Elementos del diseño del dominio.	-Patrones arquitecturales candidatos.
	A_09	Instanciar elementos arquitecturales para los elementos del diseño del dominio .	-Elementos del diseño del dominio. -Patrones Arquitecturales candidatos.	-Elementos Arquitecturales (Componentes y conectores).

	A_10	Identificar similitudes entre los elementos arquitecturales instanciados	-Elementos del diseño del dominio. -Elementos Arquitecturales (Componentes y conectores).	-Elementos Arquitecturales Similares.
	A_11	Decidir la selección de la arquitectura como solución arquitecturales candidata	-Elementos Arquitecturales Similares.	-Conjunto de soluciones arquitecturales candidatas. -Soporte de decisión arquitectural.
Evaluación Arquitectural del Dominio	A_12	Validar modelo de calidad del dominio con arquitectura candidatas	- Modelo de calidad del dominio. -Conjunto de soluciones arquitecturales candidata	-Documento de razonamiento arquitectural. -Arquitectura Validadas a la familia.
	A_13	Escoger arquitectura base para la familia	-Documento de razonamiento arquitectural. -Arquitectura Validadas a la familia.	-Arquitectura base para la línea del producto. -Informe sobre decisión arquitectural.

Fuente: Canelón, 2010.

REST

Rest (Representational State Transfer - Transferencia de Estado Representacional), es un estilo de arquitectura de software, de acuerdo a Navarro, R. (2006), el término es frecuentemente utilizado en el sentido de describir a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional, se refiere estrictamente a una colección de principios que resumen como los recursos son definidos y direccionados para el diseño de arquitecturas en red.

Por otra parte Fielding, R. (2000), señala que el estado de transferencia de representación (REST), es el estilo de arquitectura para sistemas hipermedia distribuidos, describiendo los principios de ingeniería de software guiados por REST. Este estilo arquitectural es una abstracción de los elementos arquitectónicos dentro de un sistema distribuido hipermedia. REST ignora los detalles de la aplicación del componente y la sintaxis del protocolo con el fin de centrarse en las funciones de los

componentes, las limitaciones de su interacción con otros componentes, y su interpretación de los elementos significativos de datos. Abarca las limitaciones fundamentales sobre los componentes, los conectores, y los datos que definen la base de la arquitectura de la Web, y por lo tanto la esencia de su comportamiento como una red basada en la aplicación.

Principios de REST

Según Navarro, R. (2006), los objetivos de este estilo de arquitectura se listan a continuación:

- Escalabilidad de la interacción con los componentes. La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ello es la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles, etc.
- Generalidad de interfaces. Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial. Esto no es del todo cierto para otras alternativas, como SOAP para los Servicios Web.
- Puesta en funcionamiento independiente. Este hecho es una realidad que debe tratarse cuando se trabaja en Internet. Los clientes y servidores pueden ser puestos en funcionamiento durante años. Por tanto, los servidores antiguos deben ser capaces de entenderse con clientes actuales y viceversa. Diseñar un protocolo que permita este tipo de características resulta muy complicado. HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- Compatibilidad con componentes intermedios. Los más populares intermediarios son varios tipos de proxys para Web. Algunos de ellos, las caches, se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls. Y por último, otro tipo importante de intermediarios, gateway, permiten encapsular sistemas no propiamente Web. Por tanto, la compatibilidad con intermediarios nos permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

ROA

Nadgouda, A. (2006), señala que la Arquitectura Orientada a Recursos (ROA), está completamente basada en REST y aprovecha sus ventajas – simplicidad, conocimientos técnicos mínimos y URI para cada recurso. El uso de elementos básicos de la WWW original hace que sea fácil que dos recursos se comuniquen.

La Arquitectura Orientada a Recursos (ROA), según Richardson L. & Ruby S. (2007), se define como la combinación de varias tecnologías de la Web, como lo son: el conjunto de criterios de REST, las ventajas de HTTP (Hypertext Transfer Protocol - Protocolo de Transferencia de HiperTexto) y el uso de las URIs (Uniform Resource Identifier - Identificador Uniforme de Recurso), en lenguajes de programación específicos, a lo que estos autores denominaron RESTful. En otras palabras la Arquitectura Orientada a Recursos también es RESTful.

De acuerdo con Richardson L. & Ruby S. (2007), la Arquitectura Orientada a Recursos en solo cuatro conceptos:

1. Recursos.
2. Sus nombres (URI).
3. Sus representaciones.
4. Los vínculos entre ellos.

Y cuatro propiedades:

1. Direccionamiento.
2. Carencia de estado.
3. Conexión.
4. Una interfaz uniforme.

Recurso.

Un recurso es cualquier cosa que sea lo suficientemente importante como para hacer referencia a una cosa en sí misma. Por lo general, un recurso es algo que puede ser almacenado en un ordenador y representado como una secuencia de bits: un documento, una fila de una base de datos, o el resultado de ejecutar un algoritmo. Un

recurso puede ser un objeto físico, como una manzana, o un concepto abstracto como el valor. A continuación algunos posibles recursos:

Versión 1.0.3 de la versión de software.

La última versión de la versión de software.

La entrada de primer weblog de 24 de octubre 2006.

El siguiente número primo después de 1024.

Los próximos cinco números primos después de 1024.

Según Méndez, E. (1999), un recurso es cualquier objeto web identificable unívocamente por un URI, es decir, un identificador uniforme de recursos como un URL. Un recurso puede ser un documento HTML; una parte de una página web como por ejemplo un elemento HTML o XML dentro de un documento fuente, una colección de páginas, un sitio web completo; y en síntesis, cualquier recurso entendido como objeto de información.

URI (Uniform Resource Identifier - Identificador Uniforme de Recurso).

Una URI es la tecnología fundamental de la Web, es lo que hace que un recurso sea realmente un recurso, es el nombre y la dirección de un recurso. Si una pieza de información no tiene un URI, entonces no es un recurso.

Como lo define Berners-Lee (2005), un URI (por sus siglas en inglés, Identificador Uniforme de Recurso) es un identificador que consta de una secuencia de caracteres que identifica sin duda alguna a un recurso, a través de un conjunto ampliable definido por separado de los sistemas de nombres. URI tienen un alcance global y se interpreta siempre con independencia del contexto, aunque el resultado de la interpretación puede estar en relación con el contexto del usuario final. Por ejemplo, "http://localhost/" tiene la misma interpretación para todos los usuarios de esa referencia, a pesar de que la interfaz de red que corresponde a "localhost" pueden ser diferentes para cada usuario final, denotando que la interpretación es independiente del acceso, lo que implica que las medidas en referencia a algo único en el mundo, debe utilizar un URI que distingue a ese recurso de todos los demás.

Las características de una URI se describen a continuación:

1.- Uniforme: Permite que diferentes tipos de identificadores de recursos sean utilizados en el mismo contexto, aun cuando los mecanismos empleados para acceder a esos recursos pudieran ser diferentes.

2.- Recurso: En cuanto al mismo, no limita el alcance de lo que podría ser un recurso, sino que el término "recurso" se utiliza en un sentido general de lo que pudiera ser identificado por un URI. Ejemplos conocidos incluyen un documento electrónico, una imagen, una fuente de información con un propósito coherente (por ejemplo, "el informe del tiempo de hoy en Maracay"), un servicio (por ejemplo, una puerta de enlace HTTP-a-SMS), y una colección de otros recursos.

3.- Identificador: Representa la información necesaria para distinguir lo que es identificable a partir de todas las otras cosas dentro de su ámbito de identificación, se refieren al distinguir un recurso de todos los demás recursos.

Representaciones de las URIs.

Las URIs deben ser descriptivas y este es uno de los puntos donde el ROA se basa en las recomendaciones de REST y las recomendaciones del Consorcio W3C. Un recurso puede tener una o varias URIs, por ejemplo: las cifras de ventas pueden estar disponibles en <http://www.example.com/sales/2004/Q4>, pero también puede estar disponible en <http://www.example.com/sales/Q42004>.

A continuación se presentan algunos ejemplos de representaciones de los recursos con su URI:

- <http://www.example.com/software/releases/1.0.3.tar.gz>
- <http://www.example.com/software/releases/latest.tar.gz>
- <http://www.example.com/weblog/2006/10/24/0>
- http://www.example.com/map/roads/USA/AR/Little_Rock
- <http://www.example.com/wiki/Jellyfish>
- <http://www.example.com/search/Jellyfish>
- <http://www.example.com/nextprime/1024>
- <http://www.example.com/next-5-primes/1024>

- <http://www.example.com/sales/2004/Q4>
- <http://www.example.com/relationships/Alice;Bob>
- <http://www.example.com/bugs/by-state/open>

Relación entre la URI y Recursos.

Por definición, no pueden existir dos recursos con el mismo nombre, de ser así, sólo se tendría un solo recurso. Sin embargo, en algún momento dentro de dos recursos diferentes pueden apuntar a los mismos datos. Cada URI designa exactamente un recurso. Si se designa más de una, no sería un identificador universal de recursos.

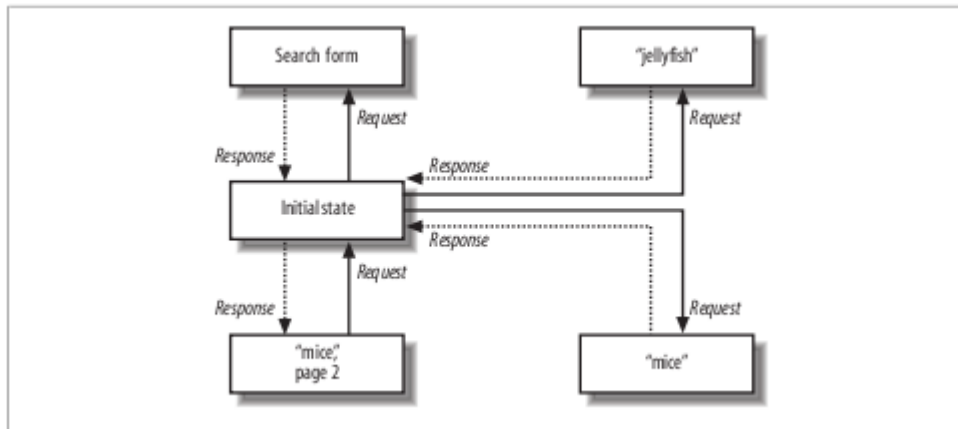
Direccionamiento.

Una aplicación es direccionable si expone los aspectos interesantes de su conjunto de datos como recursos. Dado que los recursos están expuestos a través de URIs, una aplicación direccionable expone una URI para cada pieza de información que posiblemente podría servir. Esto es por lo general un número infinito de URIs. Un ejemplo de un servicio direccionable es el servicio Amazon Simple Storage Service (Amazon S3, almacenamiento para Internet), porque cada objeto tiene su propio URI.

Carencia de estado.

Es otra característica principal de la ROA y significa que cada solicitud HTTP se ejecuta de forma separada. Cuando un cliente realiza una solicitud HTTP, que incluye toda la información necesaria para que un servidor pueda cumplir con una petición, el servidor nunca se basará en la información de las solicitudes anteriores. Si la información era importante, el cliente tendría que enviar de nuevo la solicitud.

Figura 12: Un motor de búsqueda sin estado.

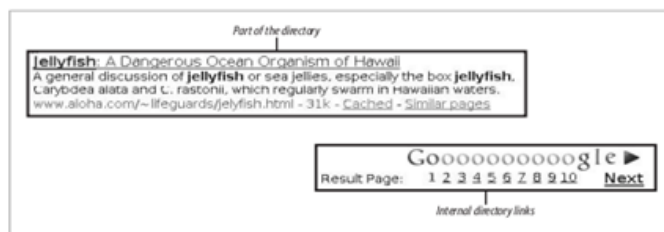


Fuente: Richardson L. & Ruby S. (2007).

Conexión o Conectividad.

En la mayoría de los casos, las representaciones no son más que estructuras de datos en serie, que están destinados a que sus datos sean absorbidos y luego se desechan, pero en la mayoría de los servicios RESTful las representaciones son documentos que contienen no sólo datos, sino enlaces o conectividad a otros recursos. Por ejemplo, si se utiliza la búsqueda de google para ubicar documentos sobre las medusas (<http://www.google.com/search?q=jellyfish>), podrá ver algunos resultados obtenidos de la búsqueda, y un conjunto de enlaces internos a otras páginas de guía. La siguiente figura muestra un ejemplo.

Figura 13: Primer plano de una página de resultados de búsqueda de Google.



Fuente: Richardson L. & Ruby S. (2007).

Una interfaz uniforme.

Se refiere a las acciones HTTP básicas que se le pueden aplicar a un recurso

en la WEB. Toda la interacción entre los clientes y los recursos es mediada a través de algunos métodos básicos de HTTP para las cuatro operaciones más comunes, las cuales se describen a continuación:

Recuperar una representación de un recurso: HTTP GET.

Crear un nuevo recurso: HTTP PUT para un nuevo URI o HTTP POST para uno existente.

Modificar un recurso existente: HTTP PUT para un URI existente.

Eliminar un recurso existente: HTTP DELETE.

Procedimiento básico para ROA

A continuación, según Richardson L. & Ruby S. (2007) una vez que se llegue al final del este procedimiento, se queda dispuesto a poner en práctica los recursos en el lenguaje y el framework que se desee. Los pasos a seguir son:

- 1) Calcular el conjunto de datos.
- 2) Dividir el conjunto de datos en los recursos.
Para cada tipo de recurso:
 - 3) Colocar el nombre de los recursos con URIs.
 - 4) Exponer un subconjunto de la interfaz uniforme.
 - 5) Diseñar la representación o representaciones aceptadas por el cliente.
 - 6) Diseñar la representación o representaciones, servidas al cliente.
 - 7) Integrar un recurso con los recursos existentes.
 - 8) Considerar el curso típico de eventos: lo que se supone que suceda.
 - 9) Considerar las condiciones de error: lo que podría ir mal.

Calcular el conjunto de datos.

Un servicio web comienza con un conjunto de datos, o por lo menos con una idea de lo que podría formar el conjunto de datos, que será recibido u ofrecido a los usuarios.

Este es un paso estándar y primordial en cualquier análisis. En algunos casos

se tiene que elegir el conjunto de datos, y en otros casos se tratará de exponer los datos que ya se tienen. Se puede retornar a este paso, tantas veces sea necesario, hasta lograr obtener la mejor manera de exponer el conjunto de datos como recursos.

Dividir el conjunto de datos en los recursos.

Una vez que exista un conjunto de datos en mente, el siguiente paso es decidir como exponer los datos como recursos HTTP. Se debe recordar que un recurso es cualquier cosa lo suficientemente interesante como para ser objeto de un enlace de hipertexto. Cualquier cosa que pueda ser referenciado por el nombre, debe tener un nombre. Los tres tipos de recursos que comúnmente se exponen son:

- Recursos únicos en su tipo, en un URI conocida, predefinidos por los aspectos especialmente importantes de la aplicación. Esto incluye directorios de nivel superior de otros recursos disponibles.
- Un recurso para cada objeto expuesto a través del servicio. Un servicio puede exponer a muchos tipos de objetos, cada uno con sus propios recursos.
- Recursos que representan los resultados de los algoritmos aplicados al conjunto de datos.

Colocar el nombre de los recursos con URIs.

Los recursos son nombrados con los URI, y existen tres reglas básicas para el diseño de URI, obtenido de la experiencia colectiva:

- 1) Utilizar variables de ruta para codificar la jerarquía: / padre / hijo
- 2) Colocar caracteres de puntuación en el path de las variables para evitar la implicación de la jerarquía que no existe: / parent/child1; child2
- 3) Utilice las variables de consulta que implica entradas en un algoritmo, por ejemplo: / q = buscar medusas & start = 20

Exponer un subconjunto de la interfaz uniforme.

En este paso se establecen los métodos HTTP a utilizar, este paso es bastante

simple ya que la interfaz uniforme es siempre la misma. Los métodos utilizados pueden ser GET, HEAD, OPTIONS, PUT, POST y DELETE, dependiendo del caso necesitado.

Diseño de Representaciones.

El objetivo principal de cualquier representación es el de transmitir el estado del recurso.

Diseñar la representación o representaciones aceptadas por el cliente.

Se refiere a los documentos que el servidor envía al cliente, en otras palabras es la forma que el servidor transmite el estado de un recurso. En este paso se decide el tipo de formato y qué datos enviar, cuando un cliente solicita un recurso.

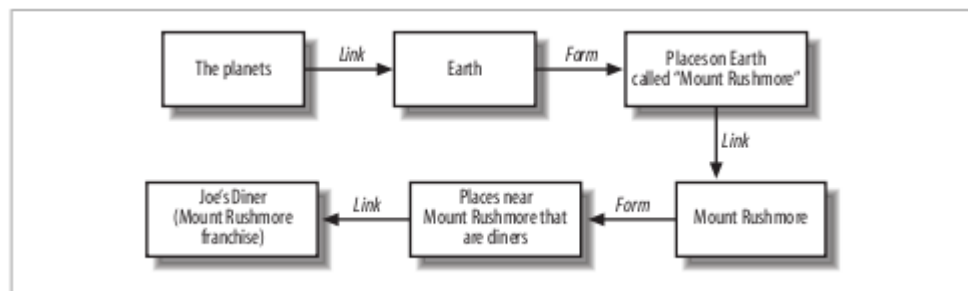
Diseñar la representación o representaciones, servidas al cliente.

En este paso se definen los requerimientos que deben ser exigidos al cliente, para que este pueda realizar la solicitud de un recurso, por ejemplo si se debe solicitar cuenta de usuario, contraseña, etc.

Integrar un recurso con los recursos existentes.

En este punto se establecen los vínculos o enlaces que se necesiten entre los recursos, como se puede apreciar en el ejemplo de la figura siguiente figura.

Figura 14: El camino desde la raíz de servicio a un restaurante cerca del Monte Rushmore.



Fuente: Richardson L. & Ruby S. (2007).

Considerar el curso típico de eventos: lo que se supone que suceda.

En este paso se consideran los códigos HTTP de respuestas que pueden ocurrir cuando un cliente envía una solicitud. Como es habitual en HTTP, el servidor lee la solicitud, toma una medida detrás de las escenas, y sirve una respuesta, el servidor podría devolver un código de respuesta feliz como 200 ("OK"), 201 ("Created") o 205 ("Reset Content").

Considerar las condiciones de error: lo que podría ir mal.

Una solicitud para crear, modificar o eliminar un recurso tiene más condiciones de fallo que una solicitud que sólo recupera una representación. A continuación se presentan algunas de las condiciones de error que se deben considerar:

- El problema más obvio es que la representación del cliente pueda no ser entendida por el servidor. El código de estado, en este caso es de 415 ("Unsupported Media Type").
- Por otra parte, el cliente no podría haber proporcionado una representación. El código de estado, en este caso es de 400 ("Bad Request").
- Tal vez la representación tiene sentido, pero se le indica al servidor colocar un recurso en un estado incoherente o imposible. Por ejemplo si la representación es "password =", y no se permite las cuentas con contraseñas vacías, entonces para el caso de la contraseña vacía, probablemente el código de estado sería 400 ("Bad Request") y para un caso de incoherencia podría ser 409 ("Conflict").
- Para un caso donde por ejemplo un cliente envíe las credenciales incorrectas, por ejemplo, envía las credenciales de autorización para una cuenta de usuario totalmente diferente, el código de respuesta en este caso es 401 ("Unauthorized").
- En el caso de existir un problema no especificado en el servidor, el código de respuesta sería 500 ("Internal Server Error") o 503 ("Service

Unavailable").

SOA

Según Erl, T. (2008) SOA (Service Oriented Architecture - Arquitectura Orientada a Servicios) establece un modelo arquitectónico que tiene por objeto mejorar la eficiencia, agilidad y productividad de una empresa de servicio de posicionamiento como el principal medio a través del cual la solución lógica está representada en apoyo de la realización de los objetivos estratégicos relacionados con la informática orientada al servicio.

Es importante mencionar la aclaratoria que hace Erl, T. (2008) en cuanto a la confusión que ha generado el término SOA, ya que el indica que el término "arquitectura orientada a servicios" y sus siglas se han utilizado de manera tan amplia por los medios de comunicación y proveedores dentro de la literatura de marketing que se ha convertido prácticamente en sinónimo de la informática orientada al servicio en sí. Por lo tanto, es muy importante hacer una distinción clara entre lo que realmente es SOA y cómo se relaciona con otros servicios orientados a elementos de cálculo. Por lo que enfatiza que una implementación de SOA puede consistir en una combinación de tecnologías, productos, extensiones de la infraestructura de apoyo, y otras partes. El verdadero rostro de un desplegado de arquitectura orientada a servicios es único dentro de cada empresa, sin embargo, se caracteriza por la introducción de nuevas tecnologías y plataformas que apoyan de manera específica a la creación, ejecución y evolución de las soluciones orientadas al servicio.

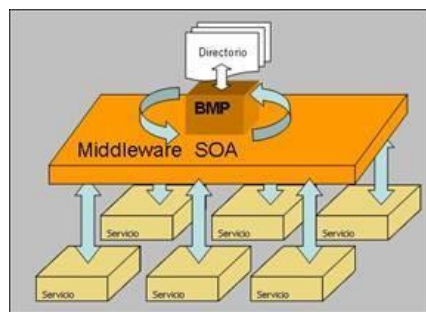
Principales características de las soluciones SOA

Al hablar de las características de SOA, es importante mencionar los componentes básicos que la conforman, así como las relaciones que se establecen entre ellos, como se puede apreciar en la figura 15. Pudiéndose decir que, en esta arquitectura, los sistemas de información están formados por servicios, cuyas definiciones se mantienen en directorios y que intercambian información a través de interfaces estándares.

De acuerdo con Monteagudo, B., Sánchez, T., Duran, A.(2008), las principales características funcionales de las soluciones SOA son las siguientes:

- Infraestructura tecnológica formada por servicios independientes: Distintos servicios se unen en forma de red para componer los procesos de negocio de la organización y pueden ser utilizados por el resto de aplicaciones.
- Posibilidad de reutilizar los servicios: Muchos de los servicios se pueden reutilizar en distintos procesos de negocio porque realizan funciones estándares.
- Bajo nivel de acoplamiento de los servicios: Cada servicio debe ser independiente del resto de servicios conectados a la misma red. De este modo, se facilitan las modificaciones tanto del código de un servicio como de la configuración de todos ellos para dar lugar a los procesos de negocio.
- Independencia entre los servicios y las características técnicas de su implementación: Favorece la independencia entre los distintos servicios, las funciones que éstos realizan deben ser independientes de las características técnicas de su implementación.
- Uso de interfaces estándares: La utilización de interfaces estándares es necesaria para permitir que servicios de distintas tecnologías puedan intercambiar información.

Figura 15: Esquema de componentes de SOA.



Fuente: Monteagudo, B., Sánchez, T., Duran, A.(2008).

Capas de una SOA

Las capas en las que se estructura la arquitectura orientada a servicios, las refiere Arsanjani, A., Borges, B. y Holley, K. (2004) como:

Capa 1: Sistemas operacionales. Contiene sistemas o aplicaciones existentes, incluyendo aplicaciones ERP (Enterprise Resource Planning - Planificación de recursos empresariales) o CRM (Customer Relationship Management - La administración basada en la relación con los clientes) existentes, sistemas heredados e implementaciones de sistemas orientados a objetos, así como aplicaciones de inteligencia de negocio. SOA puede reutilizar sistemas existentes e integrarlos utilizando técnicas de integración orientadas a servicios.

Capa 2: Componentes empresariales. Utiliza tecnología y diseños basados en contenedores y en desarrollos basados en componentes. Es la capa encargada de realizar la funcionalidad y mantenimiento de la calidad del servicio de los servicios expuestos.

Capa 3: Servicios. Los servicios de negocio residen en esta capa. Pueden ser descubiertos o pueden ser enlazados estáticamente y después invocados o coreografiados en servicios compuestos. Esta capa de exposición de servicios también permite extraer componentes empresariales de la capa anterior, componentes de unidades de negocio, y en algunos casos componentes específicos del proyecto y externalizar un subconjunto de sus interfaces en forma de descripción de servicios. Los servicios existen aislados o en servicios compuestos.

Capa 4: Composición de procesos de negocio. Define la composición de procesos de negocio a partir de los servicios de la capa anterior. Los servicios se introducen en un flujo a través de orquestación y coreografía, para ejecutar un proceso de negocio. Se pueden utilizar herramientas visuales de composición de flujos para el diseño.

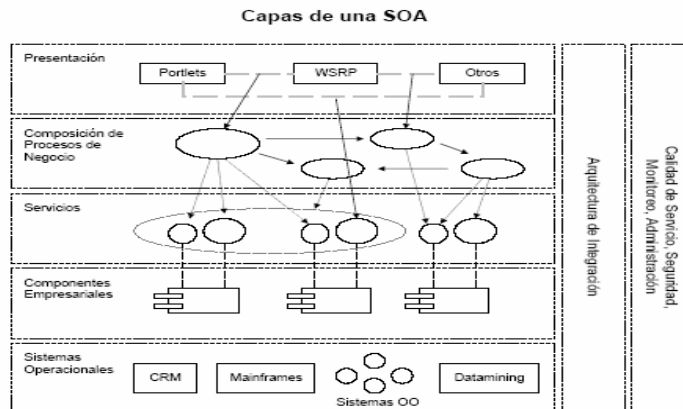
Capa 5: Presentación o acceso. Es importante resaltar que SOA desacopla la interface de usuario de los componentes.

Capa 6: Integración de servicios. Posibilita la integración de servicios a través de la introducción de un conjunto fiable de capacidades como: enrutamiento

inteligente y otros mecanismos de transformación.

Capa 7: Calidad del servicio. Esta capa proporciona las capacidades necesarias para monitorizar, gestionar y mantener propiedades de calidad del servicio, como seguridad, ejecución y disponibilidad.

Figura 16. Capas y componentes de una SOA.

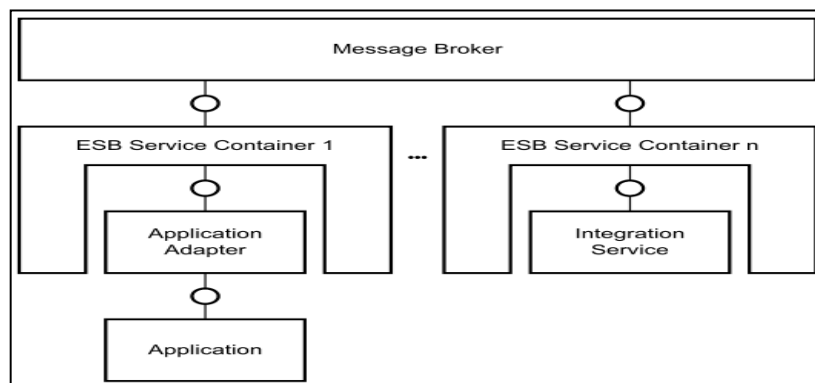


Fuente: Arsanjani, A., Borges, B. y Holley, K. (2004).

ESB (Enterprise Service Bus - Bus de Servicios Empresariales)

De acuerdo con Menge, F (2007), un ESB son estándares abiertos, basados en un sistema de mensajes, con una infraestructura distribuida de integración que proporciona enrutamiento, la invocación y los servicios de mediación para facilitar las interacciones de las diferentes aplicaciones distribuidas y servicios de una manera segura y confiable.

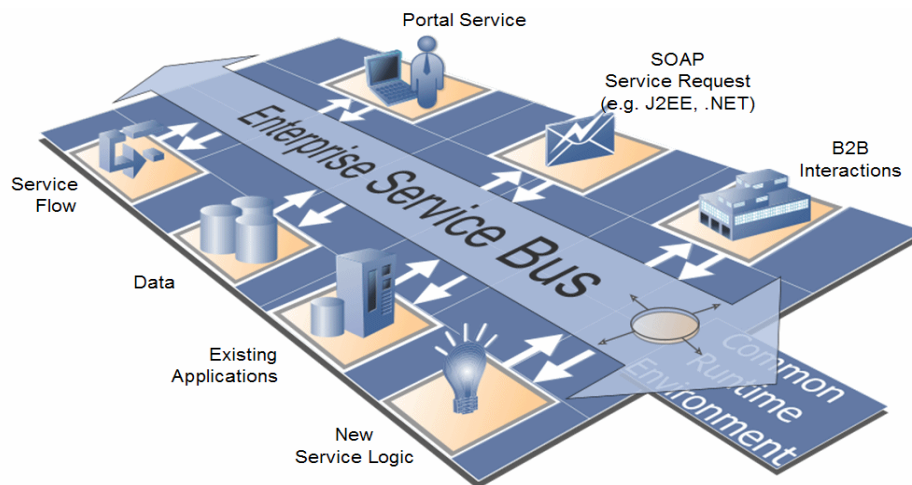
Figura 17. Un Simple escenario de ESB.



Fuente: Menge, F (2007)

El ESB (Enterprise Service Bus) representa un gran cambio en el modo de integrar aplicaciones para crear procesos de negocio, se trata de una nueva categoría de middleware de integración que remite a un concepto de troncal orientada a servicios y basada en estándares capaz de conectar y coordinar cientos de aplicaciones mediante la combinación de Servicios Web, XML y transformación de datos. El modelo de ESB se plasma en una red integrada por nodos de servicios colaboradores, desplegados en “contenedores de servicios”.

Figura 18. Vista de Alto Nivel de un ESB.



Fuente: IBM (2004)

El papel de la ESB en SOA

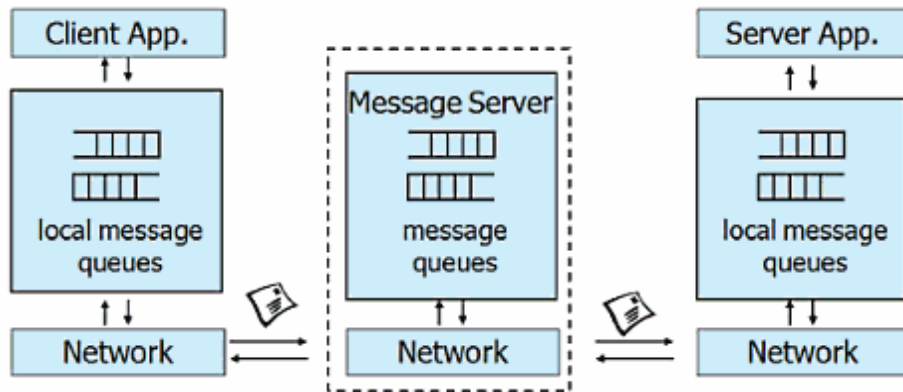
Con la finalidad de implementar una SOA, IBM (2004) apunta que tanto las aplicaciones como sus infraestructuras deben ser compatibles con los principios de SOA, permitiendo que las aplicaciones envueltas en la creación de interfaces de servicios a las funciones existentes o nuevas sean mediante el uso de adaptadores. La habilitación de la infraestructura en el nivel más básico consiste en la provisión de capacidad a las solicitudes de servicio de ruta y el transporte al proveedor de servicio correcto, por lo que el papel de la ESB es, en parte, simplemente permitir o guiar la

infraestructura de esta manera.

Middleware Orientado a Mensaje (MOM)

Según Guerra, J. (2010), este tipo de Middleware, está diseñado con el fin de manejar un intercambio de mensajes entre cliente y servidor de manera asíncrona, de tal forma que estos procesos no van a tener conexión directa entre ellos, sino a través de colas de mensajes en donde el cliente colocará su solicitud de servicio y en los que el servidor colocará la respuesta. Dado que el cliente y el servidor tienen la capacidad de poder ejecutarse en tiempos distintos, podrán existir casos en los cuales a una solicitud de servicio no necesariamente se recibirá una respuesta.

Figura 19. Organización de MOM.



Fuente: Guerra, J. (2010)

De acuerdo con Guerra, J. (2010), el Middleware MOM tiene como implementación fundamental para su funcionamiento el uso de las colas de mensaje, que tienen ventajas como:

- Tanto el cliente como el servidor controlan en que momento envía el mensaje o lo procesan.
- No existe el concepto de bloqueo, por ello no están pendiente de la llegada de mensajes, ni tienen que estar disponibles después de haber enviado el mensaje

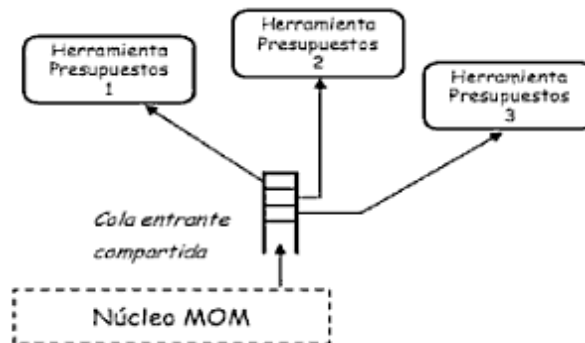
correspondiente.

- Las colas de mensajes pueden ser compartidas, así como se puede alterar la secuencia de envíos de mensajes en la cola, asignando prioridades definidas por el usuario.

Y entre las propiedades que posee MOM Guerra, J. (2010), indica:

- Al ser de comunicación asíncrona el cliente y el servidor están en bajo acoplamiento, por lo que es una buena alternativa para integración de aplicaciones.
- Soporta presentación de servicios confiables, debido al manejo antes descrito de las colas de mensaje, lo que hace ideal para implementaciones como almacenamiento persistente.
- Realiza procesamiento de mensajes, a través de un servidor de mensajes intermediario, el que posee servicios como filtrados, transformación, control de acceso, pudiendo inclusive implementarse redes de servidores de mensajes.

Figura 20. Colas de mensajes MOM.



Fuente: Guerra, J. (2010)

CAPITULO III

MARCO METODOLOGICO

El Marco Metodológico se refiere al conjunto de aspectos relevantes para el desarrollo de la investigación. El mismo está constituido por el tipo y diseño de la investigación, por la unidad de investigación y la población; incluyéndose finalmente, las fases empleadas para el desarrollo de la investigación.

Naturaleza de la Investigación

El estudio que se desarrolló responde a un paradigma cuantitativo, enmarcado en las características de la modalidad de Proyecto Factible y sustentado por una investigación descriptiva, bibliográfica y documental, cuyo objetivo general es integración de los sistemas de información para la UNEXPO basado en ROA. Se sostiene que es un proyecto factible debido a que es definido según el Manual para la elaboración de Tesis Doctorales, Trabajos de Grado y Trabajos Especiales de la Universidad Pedagógica Experimental Libertador (UPEL 2007): como la elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organización o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos, apoyada en un estudio de campo descriptivo documental.

De igual manera la investigación se apoya por un estudio de campo-descriptiva y documental. En relación a la primera de ellas, la misma es definida por Méndez (2001) como aquella en la cual los datos son recogidos en forma directa por el investigador, de la realidad donde ocurre el fenómeno, con el fin de describirla y explicar sus causas y efectos hasta entender su propia naturaleza.

En este tipo de investigación se recoge la información directa de la realidad, lo que significa que la investigación permitirá informar sobre una situación, su naturaleza y el tipo de condiciones existentes en un momento determinado.

En concordancia a la investigación descriptiva Sabino, C. (1980), afirma que es "aquella que se prepara para conocer un grupo homogéneo o fenómeno utilizando criterios sistemáticos que permiten poner de manifiesto su estructura o comportamiento."(P. 89).

En lo que se refiere a la investigación documental, UPEL (Ob cit), la define como:

“El estudio de problemas con el propósito de ampliar y profundizar el conocimiento de su naturaleza, con apoyo, principalmente, en trabajos previos, información y datos divulgados por medios impresos, audiovisuales o electrónicos. La originalidad del estudio se refleja en el enfoque, criterios, conceptualizaciones, reflexiones, conclusiones, recomendaciones y, en general, en el pensamiento del autor” (p. 15).

El estudio se desarrolló en tres fases como lo son: Fase I Caracterización, Fase II Estudio de Factibilidad, Fase III Diseño del Proyecto.

Fases del Estudio

Fase I: Caracterización de los Sistemas de Información

Unidad de Investigación

La unidad de investigación en este estudio es la Oficina Regional de Tecnología de Servicios e Información (ORTSI) de la UNEXPO Vicerrectorado Barquisimeto, la cual está exenta de toda finalidad de lucro, y tiene como objetivos, coordinar, supervisar e implantar proyectos que brinden soluciones integradas, para servicios de información.

Población y Muestra.

Población

Según Tamayo y Tamayo M. (1999), la población es la totalidad de un fenómeno a estudiar, y a partir de ésta se obtienen datos para la investigación. Para el caso de estudio de esta investigación, la población estará representada por los sistemas administrativos que soportan las actividades operativas de la UNEXPO Vicerrectorado Barquisimeto, los cuales se detallan en el cuadro 1, siendo la Oficina Regional de Tecnología y Servicio de Información (ORTSI) la encargada de gestionar el funcionamiento de estos sistemas de información.

Cuadro 1.

Sistemas de información en la UNEXPO (Vicerrectorado Barquisimeto).

Id	Acrónimo	Nombre	Ubicación	Lenguaje	Base Datos
1	Sab	Sistema Automatizado de Biblioteca	Postgrado	Clipper	Tablas DBF
2	Scp	Sistema de control de proyectos de investigación	Postgrado	Visual basic	
3	Sced	Sistema de control de postgrado	Postgrado	Access	Access
4	Sdp	Sistema de desarrollo profesional docente	Postgrado	Visual basic	Access
5		Micro contabilidad ver 3.0			
6	Contab	Sistema administrativo independiente			
7	Sibcu	Sistema de información de biblioteca central UNEXPO	Biblioteca Central	Visual FoxPro	Visual FoxPro
8	Acervo	Sistema de información computarizado para administración y control del proceso de clasificación y catalogación del Acervo bibliográfico	Biblioteca Central	Visual basic	Access
9	Personal	Personal	Producción local	Gupta	Gupta
10	Control estudios	Sistema de control de estudios	Producción local	Sql Windows	Sqltalk
11	Inscripción viejo	Sistema de inscripción, inclusión y retiro	Producción local	Sql Windows	Sqltalk
12	Inscripción nuevo	Sistema de inscripción, inclusión y retiro	Producción local	Php	Sqltalk
13	Consulta académica	Sistema de consulta académica	Producción local	Phpl	Sqltalk
14	Nómina	Nómina	Producción local	Gupta	Gupta
15	Fideicomiso	Personal	Producción local	Gupta	Gupta
16	Recurso	Prestaciones Sociales	Producción local	Gupta	Gupta
17		Sistema de Contabilidad	Postgrado	Access	Access
18	Mixnet	Sistema de Contabilidad	Acrosft Sistemas		
19	Sai	Sistema administrativo integrado	Producción local	Visual basic	Oracle
20	IPRESTO	Sistema para el cálculo de los intereses sobre las prestaciones sociales	Producción local	Php	MySql

Fuente: UNEXPO

Muestra

Con respecto a la muestra, es importante señalar que cuando se seleccionan algunos segmentos de la población con el objeto de explorar algo relacionado en una investigación se le denomina muestra.

Tamayo y Tamayo M. (1999), establece que la muestra “es un conjunto de operaciones que se realizan para estudiar la distribución de determinadas características en la totalidad de una población, universo o colectivo, partiendo de la observación de una fracción considerada” (p 96).

Es por esto, que la muestra constituye el subconjunto de una población, una parte representativa que se extrae para el estudio del problema de la investigación. Para este estudio la muestra está constituida por dos sistemas administrativos: uno de ellos es el sistema administrativo SAI, con el cual se gestiona el control de las nóminas de personal y todo lo relacionado a estas, y el otro sistema administrativo llamado IPRESTO con el cual se controla el cálculo de los intereses sobre las prestaciones sociales.

Diseño de la Investigación

En cuanto al diseño de la propuesta de la investigación, se realizó una revisión bibliográfica, descriptiva y documental para el logro de los objetivos.

En primer lugar, se caracterizó el dominio de los sistemas de información tomados como muestra, aplicando la fase de Análisis del Dominio del proceso InDoCaS, levantando los requisitos funcionales y no funcionales del área en cuestión, obteniendo así un modelo conceptual del dominio.

En segundo lugar, se estudió las factibilidades, así como las actividades, activos de software, y procesos fundamentales inherentes al dominio.

Finalmente, se aplicó el proceso RESTful para obtener una ROA y lograr así la propuesta de las aplicaciones IPRESTO y SAI de la UNEXPO Vicerrectorado Barquisimeto, tomado como caso de estudio.

Técnicas e Instrumentos de Recolección de Datos

Para la presente investigación, se utilizó el método de revisión bibliográfica, descriptiva y documental, el mismo permite conocer conceptos, características y funcionamiento de las mejoras de procesos aplicadas a las organizaciones. Según Lee (1992), expone que “muchos investigadores dedicados a hacer estudios tanto cuantitativos como cualitativos, sugieren combinar más de un método en una misma investigación”. También se requirió de consultas en Internet dado que es una fuente actualizada en el área de: un enfoque de integración de aplicaciones basadas en ROA para mejoras de procesos a unos costos muy atractivos. De la misma manera se aplicaron técnicas de resumen, subrayado, presentación de cuadros, ilustraciones, diagramas, que permiten de forma teórica apoyar la presente investigación.

Fase II: Estudio de Factibilidad

Factibilidad Técnica

Para una integración de aplicaciones, desde el punto de vista tecnológico, se requiere de recursos de software, personal calificado y hardware.

Con respecto al software requerido para el desarrollo de la investigación, se consta principalmente de lenguajes de programación y herramientas que permiten documentar los componentes de software. Para los lenguajes de programación se utilizan frameworks manejados perfectamente por el personal técnico de la UNEXPO, los cuales se describen a continuación:

- Para el lenguaje de Programación PHP están: Symfony Php, Cakephp y Kumbiaphp.
- La herramienta de programación de Visual Basic 6.0.

Adicionalmente se agregan, herramientas de modelado para UML con licencia gpl como Dia, ArgoUML, Eclipse Uml, entre otros, herramienta ACME para modelado de la notación SPEM y herramientas para realizar documentos de oficinas como libreoffice, para documentar el proceso de desarrollo.

De lo antes mencionado, en cuanto al software se refiere, es muy factible el

uso de "licencias libres", decretadas por el Gobierno Nacional Venezolano en el año 2001, ya que no representan grandes inversiones para el presupuesto de la organización.

En cuanto al personal calificado, es necesario personas graduadas en carreras como Ingeniero o Licenciado en el área de Sistemas o Informática, con una experiencia mínima de tres años en el área de desarrollo de sistemas, como por ejemplo Sistemas Administrativos. Es importante personas que dominen y tengan conocimientos sobre arquitectura de software y metodologías de mejoras de procesos.

Por lo antes planteado es factible en la UNEXPO Vicerrectorado Barquisimeto, ya que existen al menos cuatro (4) personas que poseen experiencia en el área de desarrollo de sistemas administrativos y metodologías de mejoras de procesos.

Por último, se cuentan con equipos de buen desempeño, con las siguientes características de hardware: A nivel de estaciones de trabajo, Procesador Pentium IV, 4 GB de RAM, 120 GB de disco duro, monitor a color, ratón, teclado, unidad de CD/RW/DVD/RW y a nivel de servidores se cuenta con equipos de dos (2) Procesador Intel XEON 3,5 GB de RAM, 203,49 de disco duro

Factibilidad Institucional

Con relación al aspecto institucional la propuesta de una integración de aplicaciones, fortalecerá la interrelación entre sus aplicaciones, esto garantizaría a la institución mejorar la efectividad de sus procesos administrativos, que se lleva a cabo dentro de sus departamentos. En este orden de ideas es importante resaltar que la UNEXPO cuenta con el personal idóneo que está en condiciones de aprovechar al máximo una propuesta como la que se plantea en el presente trabajo de investigación.

Factibilidad Operativa

En relación al aspecto de la factibilidad operativa desde el punto de vista de

los usuarios, se tiene que esta queda garantizada ya que el manejo de las aplicaciones dentro de la UNEXPO, es conocido por los miembros del personal relacionados a las mismas.

Por otra parte, desde el punto de vista de servidores, se requiere de un administrador. Este profesional debe contar con los conocimientos y habilidades que le permitan manejar los aspectos inherentes a mantenerlos en funcionamiento, incluyéndose tareas de: instalación, administración, seguridad, y otros. La operatividad en este sentido está garantizada por la estructura organizativa asociada a la UNEXPO.

Factibilidad Económica

Haciendo referencia a la factibilidad económica, se dispone de cuatro (4) computadoras Pentium IV con 120gb, 4gb de RAM, monitor a color, ratón, teclado, unidad de CD/RW/DVD/RW y tres (3) servidores de dos (2) Procesador Intel XEON 3,5 GB de RAM, 203,49 de disco duro, con casi todas las herramientas necesarias para llevar adelante la investigación. Cada computadora cuesta aproximadamente Bs.F 8.000,00, al multiplicarlos por las cuatro (4) genera como resultado un monto de Bs.F 32.000,00 y cada servidor esta alrededor de Bs.F 12.000,00, que al multiplicarlos por los tres (3) servidores genera como resultado un monto de Bs.F 36.000,00, generando un total general de Bs.F 68.000,00. No obstante, la UNEXPO posee las computadoras que cumplen tanto con la cantidad como las especificaciones señaladas.

En cuanto al personal, el costo por cada Ingeniero o Licenciado en el área de sistema es de Bs. 4.100,00 mensual en el ámbito universitario y en la Oficina Regional de Tecnología y Servicios de Información (ORTSI) UNEXPO Vicerrectorado Barquisimeto, trabajan 4 personas, el costo total mensual es de Bs. 16.400,00. Es importante destacar que este costo del personal es cancelado en su totalidad por la UNEXPO.

Para finalizar, el costo de las aplicaciones de software a utilizar es de Bs 0,00,

debido a estos se encuentran en la red y no poseen costo en cuanto a la adquisición y licencia.

Es importante destacar que la UNEXPO no cuenta en la actualidad, ni tampoco para un futuro cercano, con un presupuesto de inversión en el área tecnológica, por lo que la presente investigación se muestra como una buena alternativa para la institución, debido a que existen tanto las herramientas, como los equipos básicos y necesarios para desarrollar la propuesta.

CAPITULO IV

Fase III: Diseño del Proyecto Propuesto

Justificación

Un sistema de información debe mantener una constante actualización, que vaya a la par del crecimiento progresivo de cualquier organización. Según Burch, J. (2000) “Muchos expertos calculan que un ciclo de vida normal para un sistema es de dos a diez años, antes de que sea necesario un rediseño importante o una reparación total” (p. 32). Pero es importante destacar que dichos cálculos acerca del Ciclo de Vida de un sistema están sujetos a muchas condiciones y es necesario realizarlo por separado para cada sistema o subsistema de información.

En toda entidad dinámica, la utilidad de la salida del sistema de información, la eficiencia de sus operaciones y la confiabilidad en la operación global del sistema pueden variar de manera considerable con el tiempo.

Los sistemas de información están expuestos a deterioraciones, obsolescencia y por último, a la sustitución. No obstante, rara vez acontece que en un momento dado, se le hagan mejoras al sistema de información completo. En consecuencia, es evidente que parte del sistema de la organización, necesitará continuamente reparación o reconstrucción.

Pero uno de los retos más importantes que deben enfrentar las organizaciones es la integración de una diversa cantidad de sistemas de información que han sido implantados durante años. Según Vázquez, O. (2005), La integración de aplicaciones es un tema muy amplio y difícil, en él no existen respuestas sencillas y siempre hay personas con más habilidades que otras diseñando soluciones de integración.

El conocimiento que poseen pueden haberlo adquirido con la práctica mediante prueba y error, o a través de experimentados arquitectos de integración. Basándose en sus experiencias pueden resolver nuevos problemas comparándolos con problemas que han resuelto previamente, ya que detectan las características comunes de los problemas y conocen la solución asociada. Estas soluciones no son componentes o fragmentos de código que se pueden copiar y pegar, sino descripciones que permiten resolver ciertos tipos de problemas que se presentan frecuentemente y ayudan a entender cómo alcanzar una integración de alto nivel tomando como punto de partida las implementaciones de los sistemas actuales.

Dentro de este marco, la propuesta de una integración de aplicaciones planteado en el presente trabajo de investigación, esta orientada hacia el enfoque ROA para los sistemas de información de la UNEXPO Vicerrectorado Barquisimeto, que le permitirá mejorar la gestión de los procesos administrativos que se realizan. Estos beneficios también impactarán positivamente a la institución, dado que los sistemas están relacionados con la parte operativa de la organización y mientras los sistemas estén trabajando de forma integrada, mayores serán los beneficios obtenidos. A continuación en el siguiente cuadro 2 se presenta una pequeña comparación entre los beneficios del enfoque ROA con respecto a otro enfoque de integración.

Cuadro 2.

Comparativo de las Arquitecturas ROA y SOA.

ROA	SOA
1.-En ROA la importancia se da al recurso.	1.- En SOA, la importancia se da al servicio.
2.- ROA es la facilidad de aplicación, la agilidad del diseño, y el enfoque ligero a las cosas.	2.- Arquitectura Orientada a Servicios describe el diseño de un sistema de modelado de procesos de negocio como servicios o acciones.
3.- Arquitectura orientada a recursos es un conjunto de directrices de una	3.-Arquitectura orientada a servicios es un conjunto flexible de los principios de

implementación de una arquitectura REST.	diseño utilizados durante las fases de desarrollo e integración de sistemas en computación.
4.- ROA se centra en la exposición de muchos recursos.	4.- SOA se centra en la exposición de muchas acciones.
5.- ROA es ideal para aplicaciones que están basadas en recursos naturales, tales como un lector de feeds RSS donde están los recursos que el usuario está interesado en obtener información y luego actualizar el estado de ese recurso.	5.- SOA se utiliza para encapsular los datos con la lógica de negocio que opera en los datos, con el único acceso a través de una interfaz de servicios publicados.
6.- ROA es un componente de acceso directo distribuido que se maneja a través de una interfaz estándar, común hacer manejo de los recursos posibles.	6.- SOA proporciona métodos para el desarrollo e integración de sistemas donde la funcionalidad del paquete como los sistemas de servicios de interoperabilidad.
7.- La funcionalidad de informes que es ante todo la funcionalidad basada en recursos.	7.- El comercio que es la principal funcionalidad basada en acción.
8.- Los recursos no viven de manera aislada, sino que se puede establecer enlaces entre ellos.	8.- Cada servicio es una unidad distinta de la funcionalidad y no interactúa con los demás servicios.

Fuente: Autor de la Investigación..

Para obtener la integración de sistemas de información para la UNEXPO Vicerrectorado Barquisimeto, se deben realizar una combinación de los procesos de la ingeniería del dominio y el método RESTful basado en ROA. Para ejecutar el primer proceso, se utiliza como base el proceso para la Ingeniería del Dominio basado en Calidad de Software (InDoCaS), que abarca las disciplinas del análisis y diseño de la Ingeniería del dominio. Con respecto al segundo proceso se siguen los pasos que permiten concebir una Arquitectura Orientada a Recursos, en donde la combinación de ambos proceso permite así obtener una propuesta de integración de sistemas de información.

De esta manera, representa un buen aporte en la línea de investigación de Ciencias de la Computación mención Ingeniería de Software, puesto que podría incentivar a otras investigaciones en el área, facilitando el aprendizaje acerca de un enfoque más sobre integración de sistemas de información, incluso se podría utilizar este modelo como un punto de partida para incorporarle adaptaciones necesarias que permitan el logro de otros objetivos en el área.

Objetivos

Objetivo General

Proponer una integración de Sistemas de Información basados en ROA. Caso de estudio: Sistemas de información SAI e IPRESTO. UNEXPO Vicerrectorado Barquisimeto.

Objetivos Específicos

Estudiar el proceso de la Ingeniería del Dominio basado en Calidad de Software (InDoCaS) para la obtención del análisis del dominio.

Ejecutar la integración de Sistemas de Información basados en ROA. Caso de estudio: Sistemas de información SAI e IPRESTO. UNEXPO Vicerrectorado Barquisimeto.

Descripción de la Propuesta

Para la presente investigación la solución de integración está fundada en una integración de los datos de las aplicaciones. El nivel de datos lo forman los procesos, técnicas y tecnologías aplicados en la extracción de información de una base de datos, transformarla si es necesario y almacenarla en otra base de datos. El costo es la principal ventaja de este tipo de integraciones ya que al no alterar la aplicación

origen, no se incurre en grandes costos de desarrollar, probar y distribuir una nueva aplicación. Además la tecnología que permite mover datos entre bases de datos y reformatearlos es relativamente económica comparada con otras tecnologías de integración de aplicaciones.

En la presente investigación, se propone una integración de sistemas de información basado en ROA, para lo que se utiliza una combinación de dos estrategias, como lo son: en primera instancia el proceso InDoCaS (proceso para la Ingeniería del Dominio basado en Calidad de Software) que nos ayuda a ejecutar el análisis del dominio y con respecto a la segunda estrategia se incorpora el método RESTful propuesto por Richardson L. & Ruby S. (2007) que permite guiar hacia una Arquitectura Orientada a Recursos.

Estructura de la Propuesta

Para la presente propuesta de integración de sistemas de información basada en ROA y lograr así su objetivo, se debe abarcar una disciplina importante de la Ingeniería del dominio como lo es el análisis del dominio. Para tal propósito, se utilizó como primer paso el proceso realizado por Canelón (2010), llamado InDoCaS, que abarca las disciplinas del análisis y diseño del dominio.

A continuación se describen solo las actividades que se toman del proceso InDoCaS, que permiten el análisis del dominio.

Tabla 4. Actividades del modelo de procesos InDoCaS aplicados al dominio.

Disciplina	Actividades		Artefactos	
	Nro.	Nombre de la actividad	Artefactos de Entradas	Artefactos de Salida
Análisis del Dominio	A_01	Identificación de Requisitos	- Modelo conceptual del dominio. - Reclamo.	-Lista de requisitos funcionales. -Lista de requisitos no funcionales.
	A_02	Obtener modelo de similitudes y variabilidad	- Modelo conceptual del dominio. -Lista de requisitos	-Conjunto de características. -Conjunto minimal

			funcionales. -Lista de requisitos no funcionales.	de requisitos funcionales y no funcionales. - Conjunto de puntos de variación.
--	--	--	---	--

Fuente: Canelón, 2010.

Una vez que se obtiene el análisis del dominio, se procede entonces con la incorporación de un proceso que permite lograr una ROA, en este sentido se utilizó el procedimiento de Richardson L. & Ruby S. (2007).

A continuación se describen solo los pasos que se toman del procedimiento de Richardson L. & Ruby S. (2007) que permiten guiar hacia una Arquitectura Orientada a Recursos.

- 1) Calcular el conjunto de datos.
- 2) Dividir el conjunto de datos en los recursos.
- 3) Colocar el nombre de los recursos con URIs.
- 4) Exponer un subconjunto de la interfaz uniforme.
- 5) Diseñar las representaciones.
- 6) Considerar el curso típico de eventos: lo que se supone que suceda.
- 7) Considerar las condiciones de error: lo que podría ir mal.

Actividades del Análisis del Dominio

Actividad 1: Identificación de Requisitos.

En la siguiente actividad se desarrollan los artefactos listas de requisitos funcionales y no funcionales del dominio comprendido por los sistemas SAI e IPRESTO de la UNEXPO Vicerrectorado Barquisimeto. Con respecto al sistema de información SAI, se presentan a continuación los requisitos levantados:

- Manejo de nóminas por vicerrectorado y núcleo.
 - Generar reportes de listado por vicerrectorado y/o núcleo.
 - Generar reportes nóminas por vicerrectorado y/o núcleo.
- Gestión de cargos de la institución con sus dedicaciones y remuneraciones

ordinarias y prestaciones sociales asociadas.

- Generar reportes de remuneración ordinaria por cargo.
- Generar listado de conceptos de las remuneraciones.
- Gestionar Cargos y el tipo de dedicación.
- Gestionar la información relacionada a los empleados como datos personales, cargos y escalas dentro de la institución, cuenta de banco para el pago de la nómina, prestaciones sociales generadas y el historial de cargos.
 - Generar reportes sobre el empleado y sus datos básicos.
 - Listado de funcionarios asignados a fondos de pensionados y jubilados.
 - Consulta de Recibos de pagos.
 - Gestionar Cargos y el tipo de dedicación de los funcionarios.
 - Listado de funcionarios asignados por bancos.

En cuanto a los requisitos levantados del sistema de información IPRESTO, se describen los siguientes:

- Gestión de maestros principales.
 - Antigüedad.
 - Categoría de Personal.
 - Tipos de Personal.
 - Dedicación.
 - Dependencia.
 - Estatus.
 - Parámetros.
 - Tasas.
 - Tipo de Anticipo.
 - Trabajadores.
- Registro de movimientos.
 - Trabajos anteriores en Administración de Pública.

- Anticipos concedidos a empleados.
 - Permisos concedidos al personal.
 - Jubilados que han ejercidos cargos.
 - Sueldos de los empleados.
- Gestión del cálculo de los intereses sobre las prestaciones sociales, tanto individual como grupal.

A continuación se presentan en la tabla 5 y 6, los requisitos funcionales y no funcionales del sistema administrativo SAI.

Tabla 5: Lista de requisitos funcionales del sistema administrativo SAI.

InDoCaS	
ARTEFACTO	
DESCRIPCIÓN	
Nombre:	Lista de requisitos funcionales del sistema administrativo SAI
Constructor:	Analista de Requisitos
Nro. Identificación ACTIVIDAD:	A_01
Nro. Identificación ARTEFACTO:	1
Nro. Identificación	Descripción del requisito funcional
1	Manejo de nóminas por vicerrectorado y núcleo.
2	Generar reportes de listado por vicerrectorado y/o núcleo.
3	Generar reportes nóminas por vicerrectorado y/o núcleo.
4	Gestionar los cargos de la institución con sus dedicaciones, remuneraciones ordinarias y prestaciones sociales asociadas.
5	Generar reportes de remuneración ordinaria por cargo.
6	Generar listado de conceptos de las remuneraciones.
7	Gestionar Cargos y el tipo de dedicación.
8	Gestionar la información relacionados a los empleados como datos personales, cargo y escalas dentro la institución, cuenta de banco para el pago de la nómina, prestaciones sociales generadas y el historial de cargo.
9	Generar reportes sobre el empleado y sus datos básicos.
10	Listado de funcionarios asignados a fondos de pensionados y jubilados.
11	Consulta de Recibos de pagos.
12	Asignar Cargos y el tipo de dedicación de los funcionarios.
13	Listado de funcionarios asignados por bancos.

Fuente: Autor de la Investigación.

Tabla 6: Lista de requisitos no funcionales del sistema administrativo SAI.

InDoCaS	
ARTEFACTO	
DESCRIPCIÓN	
Nombre:	Lista de requisitos no funcionales del sistema administrativo SAI
Constructor:	Analista de Requisitos

Nro. Identificación ACTIVIDAD:		A_01
Nro. Identificación ARTEFACTO:		2
Nro. Identificación	Reglas del negocio asociadas al dominio	Requisitos no funcionales derivados de las reglas del negocio
Políticas		
1	Los servicios funcionan en una Intranet.	-Debe funcionar en máquinas con sistemas operativo Canaima. -Cumplimientos de estándares y normativas de la institución con el fin de garantizar que el sistema se ejecute.
Procesamiento		
2	Integridad de la información	-La transacción debe ser guardada completa y correctamente en la Base de datos y cumpliendo con las reglas de negocio de la institución. -El sistema debe garantizar que una vez la transacción se confirma, su efecto no se pierde en la base de datos, a pesar de fallos posteriores.
3	El servicio debe ser garantizado en la intranet	-Hacer posible el acceso a los servicios, lo cual implica, ancho de banda apropiado y garantizar las conexiones, teniendo que solventar problema de redes. -Tiempo de transmisión apropiado, tiempo de repuesta adecuados dentro de rango establecido.
4	Solución centralizada con respecto a los datos	-Acceso centralizado a los datos mediante interfaces conectadas a la base de datos.
Implementación		
5	Fiabilidad	-El sistema en caso de fallas, debe tener mecanismos de recuperación y de respaldo de datos.
6	Restricción de miembros del grupo	-Se permite el acceso únicamente a personal autorizado por la Institución.

Fuente: Autor de la Investigación.

Seguidamente se presentan en la tabla 7 y 8, los requisitos funcionales y no funcionales del sistema administrativo IPRESTO.

Tabla 7: Lista de requisitos funcionales del sistema administrativo IPRESTO.

InDoCaS	
ARTEFACTO	DESCRIPCIÓN
Nombre:	Lista de requisitos funcionales del sistema administrativo IPRESTO
Constructor:	Analista de Requisitos
Nro. Identificación ACTIVIDAD:	A_01
Nro. Identificación ARTEFACTO:	3
Nro. Identificación	Descripción del requisito funcional
1	Gestión de los maestros: antigüedad, categoría de personal, tipos de personal, dedicación, dependencia, estatus, parámetros, tasas, tipo de anticipo, trabajadores.
2	Registrar los movimientos de los trabajos anteriores en Administración de Pública.

3	Registrar los movimientos de anticipos concedidos a empleados.
4	Registrar los movimientos de permisos concedidos al personal.
5	Registrar los movimientos de jubilados que han ejercidos cargos.
6	Registrar los movimientos de sueldos de los empleados.
7	Gestión del cálculo de los intereses sobre las prestaciones sociales, tanto individual como grupal.

Fuente: Autor de la Investigación.

Tabla 8: Lista de requisitos no funcionales del sistema administrativo IPRESTO.

InDoCaS		
ARTEFACTO		DESCRIPCIÓN
Nombre:		Lista de requisitos no funcionales del sistema administrativo IPRESTO
Constructor:		Analista de Requisitos
Nro. Identificación ACTIVIDAD:		A_01
Nro. Identificación ARTEFACTO:		4
Nro. Identificación	Reglas del negocio asociadas al dominio	Requisitos no funcionales derivados de las reglas del negocio
	Políticas	
1	Los servicios funcionan en una Intranet.	-Debe funcionar en máquinas con sistemas operativo Canaima. -Cumplimientos de estándares y normativas de la institución con el fin de garantizar que el sistema se ejecute.
	Procesamiento	
2	Integridad de la información	-La transacción debe ser guardada completa y correctamente en la Base de datos y cumpliendo con las reglas de negocio de la institución. -El sistema debe garantizar que una vez la transacción se confirma, su efecto no se pierde en la base de datos, a pesar de fallos posteriores.
3	El servicio debe ser garantizado en la intranet	-Hacer posible el acceso a los servicios, lo cual implica, ancho de banda apropiado y garantizar las conexiones, teniendo que solventar problema de redes. -Tiempo de transmisión apropiado, tiempo de repuesta adecuados dentro de rango establecido.
4	Solución centralizada con respecto a los datos	-Acceso centralizado a los datos mediante interfaces conectadas a la base de datos.
	Implementación	
5	Fiabilidad	-El sistema en caso de fallas, debe tener mecanismos de recuperación y de respaldo de datos.
6	Restricción de miembros del grupo	-Se permite el acceso únicamente a personal autorizado por la Institución.

Fuente: Autor de la Investigación.

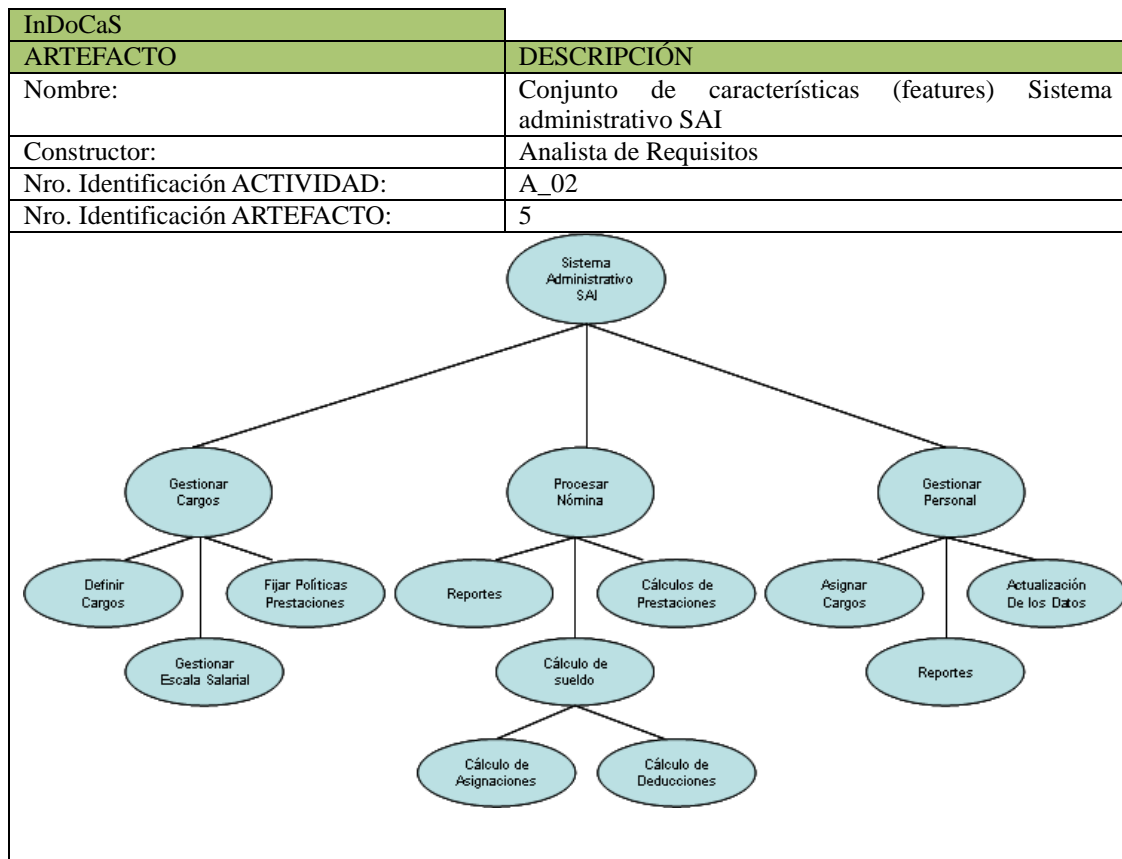
Actividad 2: Obtener modelo de similitudes y variabilidad.

Se procede entonces a definir el conjunto de requisitos mínimos y obligatorios

en los sistemas de información tanto IPRESTO como SAI de la UNEXPO Vicerrectorado Barquisimeto, usando FODA, obteniendo de esta manera los artefactos correspondientes a: conjunto de características, conjunto de puntos de variación y el conjunto minimal de requisitos funcionales y no funcionales.

A continuación, en las siguientes tablas 9 y 10, se presentan el conjunto de características:

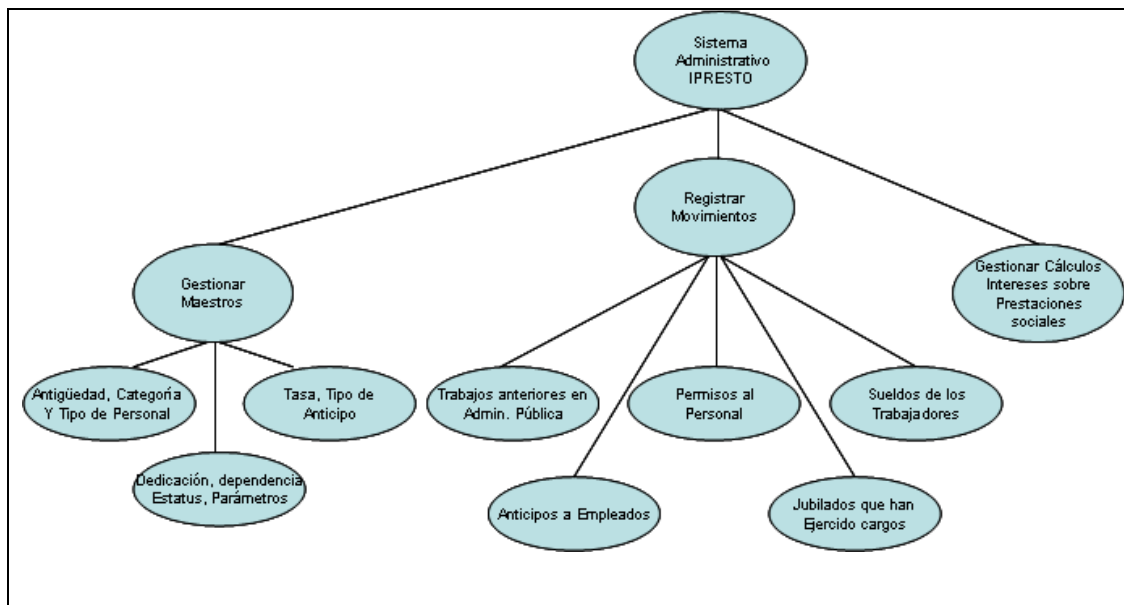
Tabla 9: Conjunto de características del sistema administrativo SAI.



Fuente: Autor de la Investigación.

Tabla 10: Conjunto de características del sistema administrativo IPRESTO.

InDoCaS	
ARTEFACTO	DESCRIPCIÓN
Nombre:	Conjunto de características (features) Sistema administrativo IPRESTO
Constructor:	Analista de Requisitos
Nro. Identificación ACTIVIDAD:	A_02
Nro. Identificación ARTEFACTO:	6



Fuente: Autor de la Investigación.

A continuación se obtienen los puntos de variación, los cuales describen dónde existen diferencias en las aplicaciones, expresan la variabilidad en las características y se presentan en las tablas 11 y 12.

Tabla 11: Conjunto de puntos de variación del sistema administrativo SAI.

InDoCaS	
ARTEFACTO	DESCRIPCIÓN
Nombre:	Conjunto de puntos de variación del Sistema administrativo SAI
Constructor:	Analista de Requisitos
Nro. Identificación ACTIVIDAD:	A_02
Nro. Identificación ARTEFACTO:	7
Característica	Punto de Variación
Gestionar Escala Salarial	Esta característica depende en gran medida al tabulador universitario que exige la Oficina de Planificación del Sector Universitario (OPSU).
Fijar Política de Prestaciones	Esta característica depende en gran medida a la formula que exige la Oficina de Planificación del Sector Universitario OPSU.

Fuente: Autor de la Investigación.

Tabla 12: Conjunto de puntos de variación del sistema administrativo IPRESTO.

InDoCaS

ARTEFACTO		DESCRIPCIÓN
Nombre:	Conjunto de puntos de variación del Sistema administrativo IPRESTO	
Constructor:	Analista de Requisitos	
Nro. Identificación ACTIVIDAD:	A_02	
Nro. Identificación ARTEFACTO:	8	
Característica		Punto de Variación
Fijar Política de tasa de para el cálculo de los intereses sobre las Prestaciones Sociales		Esta característica depende en gran medida al tabulador que fija el Banco Central de Venezuela.

Fuente: Autor de la Investigación.

Seguidamente, se presentan el conjunto de requisitos mínimos y obligatorios de los sistemas administrativos IPRESTO y SAI de la UNEXPO Vicerrectorado Barquisimeto, descritos por los artefacto “conjunto minimal de requisitos funcionales y no funcionales” que se presentan en las tablas 13 y 14.

Tabla 13: Conjunto minimal de requisitos funcionales y no funcionales del sistema administrativo SAI.

InDoCaS		
ARTEFACTO		DESCRIPCIÓN
Nombre:		Lista de requisitos funcionales del sistema administrativo SAI
Constructor:		Analista de Requisitos
Nro. Identificación ACTIVIDAD:		A_02
Nro. Identificación ARTEFACTO:		9
Nro. Identificación	Descripción del requisito funcional	
1	Manejo de nóminas por vicerrectorado y núcleo.	
2	Generar reportes de listado por vicerrectorado y/o núcleo.	
3	Generar reportes nóminas por vicerrectorado y/o núcleo.	
4	Gestionar los cargos de la institución con sus dedicaciones, remuneraciones ordinarias y prestaciones sociales asociadas.	
5	Generar reportes de remuneración ordinaria por cargo.	
6	Generar listado de conceptos de las remuneraciones.	
7	Gestionar Cargos y el tipo de dedicación.	
8	Gestionar la información relacionados a los empleados como datos personales, cargo y escalas dentro la institución, cuenta de banco para el pago de la nómina, prestaciones sociales generadas y el historial de cargo.	
9	Generar reportes sobre el empleado y sus datos básicos.	
10	Listado de funcionarios asignados a fondos de pensionados y jubilados.	
11	Consulta de Recibos de pagos.	
12	Asignar Cargos y el tipo de dedicación de los funcionarios.	
13	Listado de funcionarios asignados por bancos.	
Nro. Identificación	Descripción del requisito no funcional	

1	Cumplimientos de estándares y normativas de la institución con el fin de garantizar que el sistema se ejecute.
2	La transacción se debe guardar completa y correctamente en la base de datos y cumpliendo con las reglas de negocio de la institución.
3	El sistema debe garantizar una vez que la transacción se confirma, no perder su efecto en la base de datos, a pesar de presentarse fallos posteriores.
4	Hacer posible el acceso a los servicios, lo que implica, ancho de banda apropiado y garantizar las conexiones.
5	Tiempo de transmisión apropiado, tiempo de repuesta adecuado dentro de rango establecido.
6	Acceso centralizado a los datos mediante de interfaces conectadas a la base de datos.
7	Deben existir mecanismos de recuperación y de respaldo de datos, en caso de fallas del sistema.
8	Acceso permitido únicamente a personal autorizado por la Institución.

Fuente: Autor de la Investigación.

Tabla 14: Conjunto minimal de requisitos funcionales y no funcionales del sistema administrativo IPRESTO.

InDoCaS	
ARTEFACTO	DESCRIPCIÓN
Nombre:	Lista de requisitos funcionales del sistema administrativo IPRESTO
Constructor:	Analista de Requisitos
Nro. Identificación ACTIVIDAD:	A_02
Nro. Identificación ARTEFACTO:	10
Nro. Identificación	Descripción del requisito funcional
1	Gestión de los maestros: antigüedad, categoría de personal, tipos de personal, dedicación, dependencia, estatus, parámetros, tasas, tipo de anticipo, trabajadores.
2	Registrar los movimientos de los trabajos anteriores en Administración de Pública.
3	Registrar los movimientos de anticipos concedidos a empleados.
4	Registrar los movimientos de permisos concedidos al personal.
5	Registrar los movimientos de jubilados que han ejercidos cargos.
6	Registrar los movimientos de sueldos de los empleados.
7	Gestión del cálculo de los intereses sobre las prestaciones sociales, tanto individual como grupal.
Nro. Identificación	Descripción del requisito no funcional
1	Cumplimientos de estándares y normativas de la institución con el fin de garantizar que el sistema se ejecute.
2	La transacción se debe guardar completa y correctamente en la base de datos y cumpliendo con las reglas de negocio de la institución.
3	El sistema debe garantizar una vez que la transacción se confirma, no perder su efecto en la base de datos, a pesar de presentarse fallos posteriores.
4	Hacer posible el acceso a los servicios, lo que implica, ancho de banda apropiado y garantizar las conexiones.
5	Tiempo de transmisión apropiado, tiempo de repuesta adecuado dentro de rango

	establecido.
6	Acceso centralizado a los datos mediante de interfaces conectadas a la base de datos.
7	Deben existir mecanismos de recuperación y de respaldo de datos, en caso de fallas del sistema.
8	Acceso permitido únicamente a personal autorizado por la Institución.

Siguiendo los pasos del procedimiento que permite llevar a cabo una ROA, tal y como se definió en la estructura de la propuesta, se proceden de esta manera a desarrollar cada uno de ellos.

Paso 1: Calcular el conjunto de datos.

Se inicia entonces a definir las funcionalidades de las aplicaciones involucradas en la propuesta de integración, que permite identificar cuáles son los recursos que se proveerá; así que provechando los artefactos 9 y 10 obtenidos de la aplicación del proceso InDoCaS aplicados al dominio, los cuales permitieron hilvanar información importante y crucial para el ensamblaje y la creación del conjunto de datos que se describen a continuación:

- Generar y controlar el envío de los anticipos sobre las prestaciones sociales, asociadas a cada trabajador.
- Controlar la recepción de los anticipos sobre las prestaciones sociales, asociadas a cada trabajador.
- Generar y controlar el envío del sueldo integral por fecha asociado a cada trabajador.
- Controlar la recepción del sueldo integral por fecha asociado a cada trabajador.
- Controlar el envío del resumen de los cálculos de los intereses sobre las prestaciones sociales, asociadas a cada trabajador.
- Controlar la recepción del resumen de los cálculos de los intereses sobre las prestaciones sociales, asociadas a cada trabajador.
- Generar estatus de los pagos efectivos realizados de los intereses sobre las prestaciones sociales.

- Controlar el estatus de los pagos realizados de los intereses sobre las prestaciones sociales.

En el cuadro 3 se muestran estas funcionalidades clasificadas de acuerdo a cuál de las aplicación corresponden.

Cuadro 3.

Distribución de las funcionalidades de cada aplicación.

Funcionalidad	IPRESTO	SAI
Administración de envío de los anticipos sobre las prestaciones sociales, asociadas a cada trabajador.		X
Administración de recepción de los anticipos sobre las prestaciones sociales, asociadas a cada trabajador.	X	
Administración de envío del sueldo integral.		X
Administración de recepción del sueldo integral.	X	
Administración de envío de resumen de cálculos de intereses sobre prestaciones sociales.	X	
Administración de recepción de resumen de cálculos de intereses sobre prestaciones sociales.		X
Administración de pagos de intereses sobre las prestaciones sociales.		X
Administración de estatus de pagos realizados de los intereses sobre las prestaciones sociales.	X	

Fuente: Autor de la Investigación..

Paso 2: Dividir el conjunto de datos en los recursos.

Seguidamente se procede entonces a distinguir los siguientes recursos:

- Proceso de solicitud de cálculo de intereses de prestaciones sociales, éste corresponde al recurso principal, desde el cual se puede acceder a los demás.
- El conjunto de datos asociados al resumen de nómina.
- El conjunto de datos asociados al sueldo integral.
- El conjunto de datos asociados al resumen de intereses sobre prestaciones

sociales.

- El conjunto de parámetros para el control de estatus en pagos de intereses sobre prestaciones sociales.

Paso 3: Colocar el nombre de los recursos con URIs.

Continuando con la ROA, se procede en la cuadro 4 a presentar los recursos con su URI, así como sus respectivos enlaces.

Cuadro 4.

URIs y enlace de cada Recurso.

Recurso	URI	Enlaces
Solicitud de Cálculo	/	- Conjunto de datos - Conjunto de parámetros
Conjunto de datos	/datos	- Cada Conjunto de dato individual
Conjunto de dato individual	/datos/{id_dato}	- Listado resumen de nómina
Listado resumen de anticipos	/datos/{id_dato}/nominas	- Datos asociados a cada trabajador
Listado sueldo integral	/datos/{id_dato}/sueldos	- Datos asociados a cada trabajador
Listado resumen de intereses prestaciones sociales	/datos/{id_dato}/prestaciones	- Datos asociados a cada trabajador
Conjunto de parámetros	/parametros	- Conjunto de tipos de respuesta
Conjunto de tipos de respuesta	/tipos_respuesta	- Cada tipo de respuesta individual

Fuente: Autor de la Investigación.

Paso 4: Exponer un subconjunto de la interfaz uniforme.

Tal y como lo indica Richardson L. & Ruby S. (2007) este paso es bastante simple, ya que la interfaz uniforme es siempre la misma para el diseño de los recursos

de solo lectura, que es el caso de la presente propuesta. Los métodos utilizados son el POST y GET, es importante resaltar que para el uso del POST se realiza en muy poca escala, todo esto para evitar que el recurso se convierta en un estilo RPC (Remote Procedure Call - Llamada a procedimiento remoto) y no recargar tanto al servidor, tal como lo recomienda Richardson L. & Ruby S. (2007).

Paso 5: Diseñar las representaciones.

En este paso se procede a indicarle cada uno de los recursos identificados anteriormente, los métodos HTTP que éstos aceptan y la representación de entrada y salida en cada caso, mostrándose a continuación en siguiente cuadro.

Cuadro 5.

Recursos, métodos HTTP y representaciones.

Recurso	Método	Representación de entrada	Representación de salida
Solicitud de Cálculo	GET	N/A	Nombre y enlaces a: conjunto de datos, de Listados y de parámetros.
Conjunto de datos	GET	N/A	Enlaces a cada uno de los conjuntos de datos existentes.
Dato	GET	N/A	Período, cantidad de registros comprometidos y efectivos, monto total comprometido y efectivo, fecha de cargo y enlace a listados. Además, se agrega una de las siguientes: - En estado “Por iniciar”: interfaz para subir XML. -En estado “Iniciado”: enlace para bajar archivo XML. - En estado “Finalizado”: enlace para bajar archivo

			XML.
	PUT	Archivo XML	Idéntico a estado “Iniciado”.
Listado resumen de nómina	GET	N/A	- Fecha - Cantidad de Registro - Cédula - Nombre - Monto
Listado sueldo integral	GET	N/A	- Fecha - Cantidad de Registro - Cédula - Nombre - Monto
Listado resumen de intereses prestaciones sociales	GET	N/A	- Fecha - Cantidad de Registro - Cédula - Nombre - Porcentaje de Interes - Monto
Conjunto de parámetros	GET	N/A	Enlace a conjunto de datos y conjunto de tipos de respuesta.
Conjunto de tipos de respuesta	GET	N/A	Listado de tipos de respuesta con enlace a cada uno.
	POST	Código y valor de éxito.	Idéntico a GET de tipo de respuesta.
	DELETE	N/A	Enlace a conjunto de datos.

Fuente: Autor de la Investigación.

Paso 6: Considerar el curso típico de eventos: lo que se supone que suceda.

En este paso se consideran los códigos HTTP de respuestas para las diferentes interacciones. En el caso de esta propuesta se ha optado por los que provee HTTP. Para los casos normales de ejecución, en donde la solicitud se llevó de manera correcta en las operaciones GET y POST se obtendrá código “200 OK”.

Paso 7: Considerar las condiciones de error: lo que podría ir mal.

Por último se definen los códigos de respuestas HTTP, para los casos de error, en donde la representación de entrada tenga problemas de formato se utiliza el código “400 Bad Request”. El código “404 Not Found” se utiliza para los accesos a recursos no existente y finalmente, al intentar acceder a operaciones con métodos no implementados por los recursos, se utilizará el código “405 Method Not Allowed”.

Una vez cubierto todos los pasos y haber llegado al final del procedimiento, se queda dispuesto a poner en práctica los recursos en el lenguaje de programación y el framework que se desee, así como lo sugiere Richardson L. & Ruby S. (2007).

Para el caso de la presente propuesta se reutilizan los recursos tecnológicos ya existentes en la UNEXPO como lo son el lenguaje de programación PHP y el framework Kumbiaphp, herramientas con las cuales fue diseñado el sistema de información IPRESTO, permitiendo la activación de una librería que forma parte del núcleo de PHP como lo es la LibCurl: la cual una biblioteca creada por Daniel Stenberg, que permite conectar y comunicar a muchos tipos diferentes de servidores con diferentes tipos de protocolos, actualmente soporta los protocolos HTTP, HTTPS, FTP, Gopher, telnet, dict, archivo, y los protocolos LDAP, además también soporta certificados HTTPS, HTTP POST, HTTP PUT.

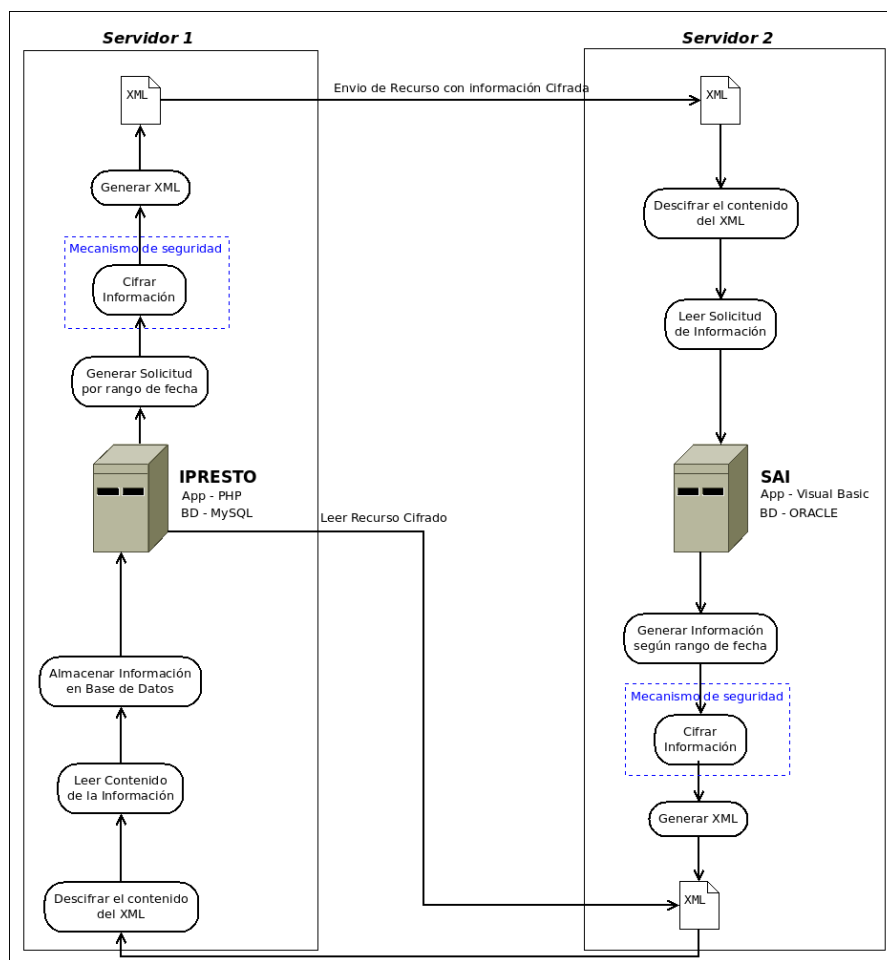
El tipo de formato que se utiliza para el envío de la información o recurso, es el formato XML, que representan un estándar y además responde a una estructura correcta y verificable, que garantiza un intercambio de información de una manera accesible y que se puede utilizar sin ningún tipo de inconvenientes para ambiente totalmente heterogéneos.

Para garantizar la seguridad en la presente propuesta de integración, en lo que respecta al traslado de los recursos en formato XML, el investigador se apoya en los estándares de seguridad recomendados por el Consorcio W3C (2011), entre los cuales se encuentra el XML Encryption, el cual contiene como requerimiento trabajar con algoritmos de cifrado arbitrarios, incluidos los sistemas de claves simétricas y asimétricas, así como la negociación dinámica del material clave.

Para el caso de la presente propuesta y basándose en el estándar antes mencionado, se utilizó la librería Mcrypt, la cual a además de formar parte y estar incorporada en el lenguaje PHP, proporciona una amplia cantidad de algoritmos de cifrado, incluso algunos recomendados por el Consorcio W3C. Para el caso de estudio se implemento el algoritmo de cifrado MCRYPT_DES.

A continuación en la siguiente figura se muestra de manera gráfica la arquitectura propuesta.

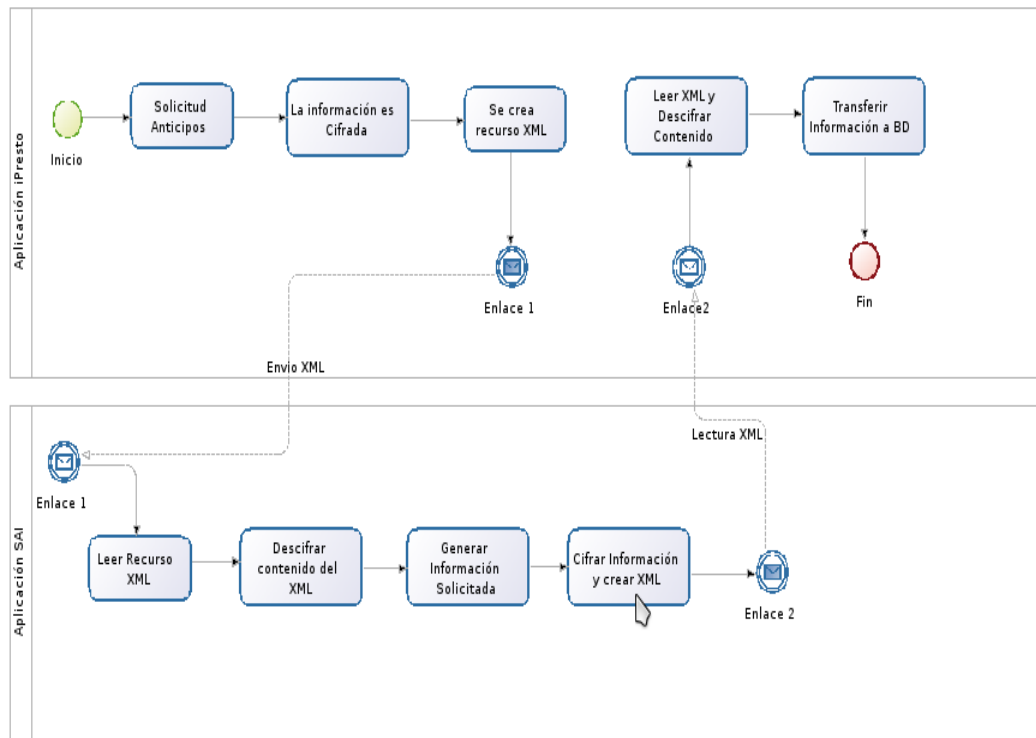
Figura 21. Arquitectura de la Propuesta.



Fuente: Autor de la Investigación.

En la figura número 22 se describe el proceso de la presente propuesta de integración, utilizando la Notación para el Modelado de Procesos de Negocio (BPMN).

Figura 22. Proceso Propuesta de Integración.



Fuente: Autor de la Investigación.

A continuación se presentan las pantallas principales de los sistemas administrativos IPRESTO y SAI. En la siguiente figura se muestra las opciones del módulo incorporado en la aplicación IPRESTO, en donde se podrá realizar la solicitud, la transferencia de la información, así como la configuración de la ubicación de los recursos.

Figura 23. Opciones del módulo Integrador del Sistema IPRESTO.



Fuente: Autor de la Investigación.

En la figura 24, se muestra la pantalla donde el usuario podrá solicitar el rango de fecha de la información a solicitar, la fecha de inicio la genera el sistema de forma automática, partiendo del último rango de fecha solicitado, manteniendo el control y evitando la duplicidad de información.

Figura 24. Pantalla Sistema IPRESTO, donde se define el rango de fecha para generar el recurso de solicitud de información.



Fuente: Autor de la Investigación.

En la figura 25, se muestra la pantalla donde el usuario podrá realizar la petición del recurso generado por la aplicación SAI con la información solicitada.

Figura 25. Pantalla Sistema IPRESTO, donde se captura el recurso con la información generada por la aplicación SAI.

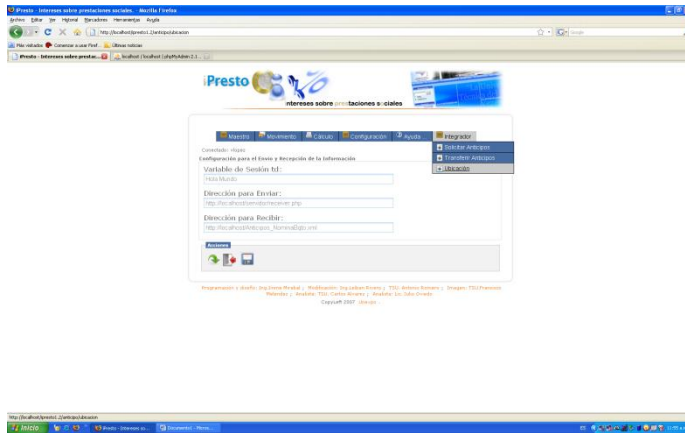


Fuente: Autor de la Investigación.

En la siguiente figura se muestra la pantalla del sistema IPRESTO donde se

podrá configurar la dirección de ubicación del recurso, tanto el que se genera con la solicitud, como el que se captura del sistema SAI.

Figura 26. Pantalla Sistema IPRESTO, donde se configura la ubicación de los recursos.



Fuente: Autor de la Investigación.

A continuación se muestran las pantallas pertenecientes al sistema SAI, donde se puede configurar la ubicación de los recursos, tanto el que recibe como el del recurso que genera.

Figura 27. Pantalla Sistema SAI, Menú de opciones.



Fuente: Autor de la Investigación.

Figura 28. Pantalla Sistema SAI, donde se configura la ubicación de los recursos.



Fuente: Autor de la Investigación.

Se ha presentado una propuesta de integración de sistema de información para la UNEXPO Vicerrectorado Barquisimeto, caso de estudio los sistemas IPRESTO y SAI, completándose las disciplinas de análisis del dominio del proceso de Ingeniería de dominio llamado InDoCaS, en combinación con el procedimiento RESTful que conlleva a una ROA. Esta propuesta puede ser utilizada como base para el diseño de integraciones de sistemas de información en las universidades públicas venezolanas que posean infraestructura estructuras muy antiguas y no cuente con presupuesto para inversiones tecnológicas.

CAPÍTULO V

Conclusiones y Recomendaciones

Conclusiones

Después de realizar la implementación de una propuesta de integración de sistemas de información y realizar un caso de estudio de los sistemas de información IPRESTO y SAI de la UNEXPO, se llegó a las siguientes conclusiones:

- 1.- El presente estudio se centró en el estudio e implementación del procedimiento sugerido por Richardson L. & Ruby S. (2007), el cual guía hacia una ROA, permitiendo llevar a cabo una integración de aplicaciones utilizando como caso de estudio los sistemas de información IPRESTO y SAI de la UNEXPO Vicerrectorado Barquisimeto.
- 2.- Se hizo necesario la incorporación de algunas actividades de la ingeniería del dominio, para lo cual se utilizó el método InDoCaS y lograr de esta manera la obtención del análisis del dominio.
- 3.- Se utilizó el método InDoCaS con la finalidad de dar respuesta al objetivo uno de la investigación, el cual hace referencia a la caracterización de los sistemas de información de la UNEXPO Vicerrectorado Barquisimeto.
- 4.- Como hallazgo se obtiene un método para integración de los sistemas de información para la UNEXPO, puesto que se realiza una fusión del método InDoCaS con el procedimiento sugerido por Richardson L. & Ruby S. (2007) que induce a una ROA.
- 5.- Los beneficios asociados a la integración de aplicaciones con la utilización de procedimientos como RESTful, que combina criterios de REST, las ventajas de http y el uso de las URIs, conlleva a una reducción en tiempo de respuesta, crecimiento controlado de las aplicaciones y mejoras en calidad de servicios prestados,

trasladándose todo esto a las actividades de los trabajadores que guardan estrecha relación con las aplicaciones integradas.

Recomendaciones

En base a la investigación realizada y toda la información recopilada, se presentan las siguientes recomendaciones:

- 1.- Continuar con el proceso de integración de aplicaciones basado en ROA, con su respectivo ajuste al dominio, y permearlo para todos los vicerrectorados en las distintas ciudades donde se encuentra la UNEXPO.
- 2.- Este enfoque puede ser generalizado para proponer integración de sistemas de información tanto en el área académica como en el mercado comercial.

BIBLIOGRAFÍA

Acevedo (2001). Introducción a la Metodología de Investigación en Ciencias Administrativas y del Comportamiento. Editorial Trillas. México.

Artículo SOA Agenda. Extraído el 23 de junio de 2009 desde <http://soaagenda.com/journal/articulos/que-es-bpm-que-es-bpms/>

Arsanjani, A., Borges, B. y Holley, K. (2004). "Service-oriented architecture", Web Service Journal, Vol. 4. No. 9, pp. 34-38.

Berners-Lee (2005). URI Generic Syntax. URL:<http://tools.ietf.org/html/rfc3986#page-6> (Consulta: Enero 25,2011).

Bertalanffy (1986) Teoría General de los Sistemas. Fondo de Cultura Económica México.

Burch, J. (2000). Sistemas de Información Gerencial. Mc Graw Hill. Cuarta edición.

Canelón R. (2010). Un Proceso para la Ingeniería del Dominio Basado en Calidad de Software. Una aplicación al dominio del aprendizaje móvil sensible al contexto. Tesis Doctoral. Universidad Central de Venezuela. Caracas.

Consortio W3C (2011). Consorcio World Wide Web. Desarrolladores de estándares de alta calidad. URL: <http://www.w3.org/standards/xml/security> (Consulta: Enero 25,2011).

Doug Kennedy, vicepresidente sénior de Alianzas y Canales Internacionales de Oracle.<http://www.infochannel.com.mx/10-h14357/www.dupapier.com.mx> Consulta: Mayo 21, 2009).

DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición. URL: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=Informaci%C3%B3n (Consulta: Mayo 12, 2009).

Erl, T. (2008). What is SOA. URL: <http://www.whatissoa.com/p10.asp> (Consulta: Abril 30, 2009)

Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures. URL: <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm> (Consulta: Agosto 25, 2009).

Gaceta oficial N° 38.095 de fecha 28/12/2004. Decreto N° 3.390. URL: <http://www.gobiernoenlinea.ve/docMGr/sharedfiles/Decreto3390.pdf> (Consulta: Mayo

12, 2009).

Guerra, J. (2010). Sistemas Distribuidos. Editorial Jaed Editores. Perú. URL: http://books.google.co.ve/books?id=AdkLDdgiFIUC&pg=PA170&lpg=PA170&dq=Middleware+orientado+a+mensajes&source=bl&ots=IZKBzKRSNC&sig=9hYaouadyvd4u-cZQODJ26SrCtM&hl=es&ei=1GsKTfHjKMGqlAer3-XVAQ&sa=X&oi=book_result&ct=result&resnum=6&ved=0CD0Q6AEwBTgK#v=onepage&q=Middleware%20orientado%20a%20mensajes&f=false (Consulta: Diciembre 16, 2010).

Hamad H., Saad. and Abed R. (2009). Performance Evaluation of RESTful Web Services for Mobile Devices. Islamic University of Gaza, Palestine.

IBM (2004). Patterns: Implementing an SOA Using an Enterprise Service Bus. URL: <http://ck20.com/MQ/WBIMB/sg246346%20Implementing%20SOA%20using%20ESB.pdf> (Consulta: Diciembre 13, 2010).

IBM (2005). Service-oriented agility: Methods for successful Service-Oriented Architecture (SOA) development. URL: <http://www.ibm.com/developerworks/webservices/library/ws-agile1/> (Consulta: Mayo 21, 2009).

Kimmel, P. (2007). Manual de UML. Editorial McGraw-Hill Interamericana Editores, S.A. México.

Lucchi, R. Y Millot, M. (2008). Resource Oriented Architecture and REST. European Commission Joint Research Centre Institute for Environment and Sustainability. ITALY

Méndez, E. (1999). RDF: Un modelo de metadatos flexible para las bibliotecas digitales del próximo milenio. URL: <http://eprints.rclis.org/handle/10760/12694> (Consulta: Agosto 23,2010).

Méndez (2001). Metodología. Diseño y desarrollo del proceso de investigación. . Editorial McGraw-Hill Internacional. Colombia.

Menge, F (2007). Enterprise Service Bus. URL: http://programm.froscon.de/2007/attachments/15-falko_menge_-_enterprise_service_bus.pdf (Consulta: Dic 11, 2010)

Monteagudo, B., Sanchez, T., Duran, A.(2008). CARÁCTERÍSTICAS DE LAS EMPRESAS QUE UTILIZAN ARQUITECTURA ORIENTADA PARA SERVICIOS Y DE SU CONTEXTO DE APLICACIÓN. URL: <http://www.jistem.fea.usp.br/index.php/jistem/article/view/138/115> (Consulta: Junio 18, 2009).

Montilva (1999). Desarrollo de Sistemas de Información. Universidad de los Andes. Mérida – Venezuela.

Nadgouda, A. (2006). The Different Web Programming Paradigms. URL: <http://iface.wordpress.com/2006/08/16/oa-the-different-web-programming-paradigms/> (Consulta: Septiembre 27, 2009).

Navarro, R. (2006). REST vs Web Services. URL: <http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf> (Consultado: 27/Julio/2009)

Pautasso C. (2009). RESTful Web service composition with BPEL for REST. Faculty of Informatics, University of Lugano, via Buffi 13, 6900 Lugano, Switzerland.

Pautasso C., Zimmermann O. and Leymann F. (2008). Restful web services vs. "big" web services: making the right architectural decision. Association for Computing Machinery. New York, NY, USA.

Pressman, R. (2002). Ingeniería del Software. Editorial McGraw-Hill Interamericana de España, S.A.U. Madrid – España.

Richardson L. & Ruby S. (2007). RESTful Web Services. O'Reilly. First Edition. Northern California. United States of America.

Richardson, L. y Ruby, S. (2009). RESTful Web Services. URL: <http://www.crummy.com/writing/RESTful-Web-Services/> (Consulta: Agosto 21,2009).

Rosenberg F., Curbera F., Duftler M., and Khalaf R. (2008). Composing RESTful Services and Collaborative Workflows: A Lightweight Approach. Technical University Vienna. Austria.

Sabino, C. (1980). "El proceso de investigación". El CID Editor, Primera Edición, Caracas, Venezuela.

Senn, J. (1992). Análisis y Diseño de Sistemas. Editorial McGraw-Hill Internacional de México, S.A. México.

Senn, J. (1999). Análisis y Diseño de Sistemas. Editorial McGraw-Hill Internacional de México, S.A. México.

Tamayo y Tamayo M. (1999). Proceso de la investigación científica, 2da Edición. Editorial Limusa. México.

UPEL (2007). Universidad Pedagógica Experimental Libertador. Manual de Trabajos de Grados de Especialización, Maestría y Tesis Doctorales. Caracas.

Vázquez, O. (2005). INTEGRACIÓN DE APLICACIONES EN LAS EMPRESAS. URL: <http://www.revistaii.cujae.edu.cu/index.php/revistaind/article/view/102/81> (Consulta: Diciembre 19, 2010).

Zhao H. and Doshi P. (2009). Towards Automated RESTful Web Service Composition. University of Georgia, Athens, GA, USA.


ANEXOS

ANEXO A. CURRÍCULO VITAE DEL AUTOR

José Valentín López Montilla


Cursante del Postgrado en Ciencias de la Computación Mención Ingeniería de software de la Universidad Centroccidental “Lisandro Alvarado”. Nació en Acarigua, Estado Portuguesa, el 30 de Noviembre de 1967. Realizó estudios de Educación primaria en el Grupo Escolar Goajira II, Acarigua-Venezuela. Seguidamente cursó sus estudios de Educación Secundaria y Diversificada en el Liceo “Eduardo Chollet Boada”, Acarigua-Venezuela. Posteriormente inicia su carrera universitaria en el Instituto Universitario Nuevas Profesiones, allí obtiene el título de “Técnico Superior en Computación” en el año 1990. Luego continuó sus estudios en el Instituto Universitario Politécnico Santiago Mariño allí, obtiene el título de “Ingeniero en Sistemas” en el año 2000. Posee experiencia laboral como Analista Programador de Sistema en la Empresa Corporación Grupo Químico (Sherwin Williams), Valencia-Venezuela, desde 1991 a 1992, como Analista Programador de Sistema en la Empresa Productores Asociados Chispas, Acarigua-Venezuela, desde 1992 a 1993, como Analista Programador de Sistema en la Empresa Alimentos Ceres, C.A., Acarigua-Venezuela, desde 1993 a 1995, como Subgerente de Sistemas en las Empresas del Grupo Continental Grain co: Alimentos del Centro (ALCECA) y Alimentos Primarios Animales (APACA), Cagua-Venezuela, desde 1995 a 1999, como Gerente de Desarrollo y Soporte en la Empresa Multicomp, C.A., Maracay-Venezuela, desde 2000 a 2000, como Asesor de Sistemas en las Empresas: Reproductora del Centro, C.A., Nirgua-Venezuela y Llano Petrol, C.A., Araure-Venezuela, desde 2000 a 2000, como Gerente de Informática en la Empresa Agrobueyca, S.A., Santa Cruz de Aragua-Venezuela, desde 2001 a 2005, Actualmente Analista de Tecnología de Información y Comunicación en la Universidad Nacional Experimental Politécnica “Antonio José de Sucre” (UNEXPO), desde 2006 a la actualidad.

ANEXO B. ESTRUCTURA ORGANIZATIVA

	Estructura Organizativa	Figura 1/1
---	-------------------------	---------------



ANEXO C. ESTRUCTURA ORGANIZATIVA

	Estructura Organizativa	Figura 1/1
---	-------------------------	---------------

