

UNIVERSIDAD CENTROCCIDENTAL  
“LISANDRO ALVARADO”

**DISEÑO DE UNA ARQUITECTURA DE INTEGRACIÓN DE  
APLICACIONES SOBRE LA BASE DE INTERCAMBIO DE DOCUMENTOS,  
CASO DE ESTUDIO ALCALDIA DEL MUNICIPIO IRIBARREN, ESTADO  
LARA**

Ambar A. Mendoza C.

Barquisimeto, 2015

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”  
DECANATO DE CIENCIAS Y TECNOLOGÍA  
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

**DISEÑO DE UNA ARQUITECTURA DE INTEGRACIÓN DE  
APLICACIONES SOBRE LA BASE DE INTERCAMBIO DE DOCUMENTOS,  
CASO DE ESTUDIO ALCALDIA DEL MUNICIPIO IRIBARREN, ESTADO  
LARA**

Trabajo presentado para optar al grado de  
Magister Scientiarum

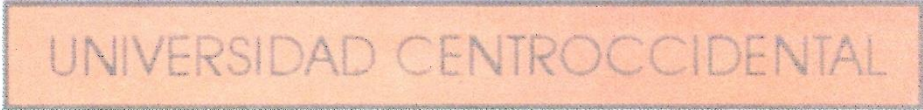
Por: Ambar A. Mendoza C.

Barquisimeto, 2015



Universidad Centroccidental "Lisandro Alvarado"  
 Decanato de Ciencias y Tecnología  
 Coordinación de Postgrado

**ACTA VEREDICTO TRABAJO DE GRADO**

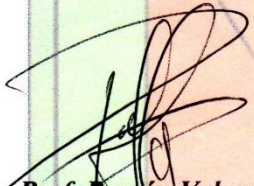


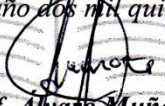
Nosotros, Miembros del Jurado Examinador del Trabajo de Grado titulado "Diseño de una Arquitectura de Integración de aplicaciones sobre la base de intercambio de documento, caso de estudio Alcaldía del Municipio Iribarren, Estado Lara", presentado por la Ing. Ambar Arelys Mendoza Colmenárez, titular de la Cédula de Identidad N° 15.960.266, como requisito para optar al grado académico de **MAGISTER SCIENTIARUM EN CIENCIAS DE LA COMPUTACION**, ofrecido por el programa: Maestría en Ciencias de la Computación del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", hacemos constar que hoy, Cuatro de Mayo de dos mil quince, a las nueve y quince minutos de la mañana (09:15 a.m.) se realizó el examen público de Defensa de Trabajo de Grado, de acuerdo a lo establecido en la Normativa sobre Trabajos de Grado de la UCLA. Una vez rendido el examen, este Jurado emite el siguiente veredicto:

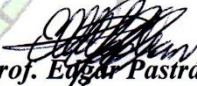
El Trabajo de Grado fue:

\*\*\*\*\* **APROBADO** \*\*\*\*\*

Dando fe de ello, levantamos la presente acta en la ciudad de Barquisimeto a los Cuatro días del mes de Mayo del año dos mil quince.

  
**Prof. Ramón Valera**  
 Jurado Principal  
 C.I. N° 12.245.419

  
**Prof. Alvaro Muñoz**  
 Presidente de Jurado  
 C.I. N° 14.746.379

  
**Prof. Edgar Pastrán**  
 Jurado Principal  
 C.I. N° 15.447.007



*Una voz del pensamiento*

Coordinación de Postgrado, Decanato de Ciencias y Tecnología, Núcleo Obelisco  
 Avenida Las Industrias, Módulos C y D, Barquisimeto, Estado Lara  
 Teléfonos (0251) 2591712-2591618 Fax: 2591718 <http://postgrado.ucla.edu.ve/ciencias/>

## **DEDICATORIA**

A Dios por sobre todas las cosas, a mis padres, mi hermana,  
mi sobrino, mi esposo y mi hermosa hija.

## **AGRADECIMIENTO**

Primeramente A Dios, por las abundantes bendiciones que derrama sobre mi vida, por darme la fuerza para seguir adelante, por ayudarme y acompañarme en todo momento.

A mi mamá Por el esfuerzo tan grande que has hecho durante toda mi vida y la motivación constante que me ha permitido ser una persona de bien, porque eres mi apoyo incondicional, mi madre, mi amiga. A mi papá el ejemplo de constancia que lo caracterizan, por cultivar e inculcar ese sabio don de la responsabilidad.

A mi hermosa niña Rubí, este logro es tuyo hija, este esfuerzo que hicimos juntas valió la pena porque lo hemos logrado, espero que sea para ti un ejemplo a seguir.

A mi hermana, que con su amor me ha enseñado a salir adelante. Gracias por toda tu ayuda y apoyo incondicional.

A mi Sobrino, gracias por permitirme ser parte de tu vida. Espero que este logro sea para ti un ejemplo a seguir.

A mi Esposo, por estar siempre a mi lado apoyándome y brindándome todo su apoyo, amor, estímulo y comprensión. Por acompañarme a estudiar en mis noches de desvelo.

A mi tutor por su profesionalismo por compartir sus conocimientos, por sus enseñanzas, dedicación, apoyo y comprensión en el transcurso de la investigación.

A todas las personas que de una u otra manera estuvieron apoyándome y recordándome que tenía que continuar.

Gracias...

## ÍNDICE GENERAL

ÍNDICE GENERAL .....	vi
ÍNDICE DE CUADROS .....	ix
ÍNDICE DE ILUSTRACIONES.....	x
RESUMEN.....	xii
INTRODUCCIÓN.....	1
CAPITULO	
I    EL PROBLEMA .....	3
Planteamiento del Problema.....	3
Objetivos de la Investigación .....	8
Objetivo General .....	8
Objetivos Específicos .....	8
Justificación e Importancia.....	9
Alcances y Limitaciones .....	10
II   MARCO TEÓRICO.....	11
Antecedentes .....	11
Bases Teóricas .....	16
Tecnologías de Información y comunicación (TIC).....	17
Sistemas de Planeación de Recursos Empresariales (ERP) .....	18
Metodologías de Desarrollo .....	19
Ingeniería de Requisitos Orientada a Objetivos .....	21
Enfoques de Ingeniería de Requisitos Orientada a Objetivos .....	22
NFR framework.....	22
KAOS.....	24
Framework I* .....	25
Casos de Uso .....	30
Diagrama de Casos de Uso .....	30
Narración de Casos de Uso .....	32
Integración de Aplicaciones (EAI) .....	32
Integración Orientada a la Información .....	33
Integración Orientada a Procesos .....	33
Integración Orientada a Servicios.....	34
Integración Orientada a los Portales .....	35

Patrones de Integración Empresarial .....	41
Patrones de canal del mensaje .....	42
Patrones de construcción del mensaje.....	45
Patrones de enrutamiento del mensaje .....	48
Patrones de transformación del mensaje .....	53
Patrones de puntos de acceso (EndPoint).....	56
Patrones de gestión del sistema .....	60
Arquitectura Orientada a Servicios (SOA) .....	62
Servicio Web.....	63
Bus de Integración de Servicios (ESB) .....	66
Plataforma de Integración .....	67
III MARCO METODOLÓGICO .....	69
Naturaleza de la Investigación .....	69
Investigación Teórica Documental .....	71
Definición de la Propuesta sobre un caso de Estudio .....	71
Fase 1: Diagnóstica .....	71
Fase 2: Estudio de Factibilidad.....	71
Fase 3: Diseño de Propuesta.....	73
IV PROPUESTA DEL ESTUDIO.....	76
Etapa I: Realizar un diagnóstico sobre la manera en que se produce el intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos .....	76
Etapa II: Realizar un análisis orientado a servicios para conocer los requisitos funcionales y no funcionales presentes para el buen intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.....	81
Modelar Objetivos.....	83
Iteración 1.....	84
Iteración 2.....	86
Iteración 3.....	88
Generar Casos de Uso. ....	90
Iteración 1.....	90
Iteración 2.....	92
Catálogo de Servicios .....	99
Etapa III: Diseñar una arquitectura de integración de aplicaciones que satisfaga los requerimientos presentes en el caso de estudio, específicamente en la Alcaldía de Iribarren del Estado Lara .....	100
Iteración 1.....	101

Iteración 2.....	105
Iteración 3.....	114
IV CONCLUSIONES Y RECOMENDACIONES.....	132
Conclusiones.....	132
Recomendaciones.....	134
REFERENCIAS BIBLIOGRÁFICAS.....	135
ANEXOS.....	138
Currículum Vitae del Autor.....	139



## ÍNDICE DE CUADROS

### CUADRO

1	Plantilla de narración de Casos de Uso .....	32
2	Patrones de canal del mensaje .....	42
3	Patrones de construcción del mensaje .....	46
4	Patrones de enrutamiento del mensaje .....	48
5	Patrones de transformación del mensaje .....	54
6	Patrones de puntos de acceso (EndPoint) .....	56
7	Patrones de gestión del sistema .....	60
8	Caso de uso: Liquidar Impuesto .....	92
9	Caso de uso: Consultar Datos de Contribuyente .....	93
10	Caso de uso: Consultar parámetros de cálculo .....	93
11	Caso de uso: Generar Cálculo .....	94
12	Caso de uso: Generar Factura .....	94
13	Caso de uso: Realizar Compensación. ....	95
14	Caso de uso: Procesar Archivos Bancarios.....	95
15	Caso de uso: Generar Solvencia .....	96
16	Caso de uso: Generar Constancia .....	96
17	Caso de uso: Generar Indicadores.....	97
18	Caso de uso: Consultar Trámites.....	97
19	Caso de uso: Procesar Facturación.....	98
20	Caso de uso: Actualizar Información .....	98
21	Caso de uso: Catálogo de Servicios .....	99
22	Patrones de integración de aplicaciones basados en mensajes .....	102

## ÍNDICE DE ILUSTRACIONES

### IMAGEN

1	Elementos de NFR framework .....	24
2	Construcción básica del framework KAOS .....	25
3	Representación de Objetivos en i* .....	27
4	Representación de Actores en i* .....	27
5	Representación de Tareas en i* .....	27
6	Representación de Recursos en i* .....	27
7	Relaciones de Dependencia en i* .....	28
8	Constructores en i* .....	29
9	Elementos del Diagrama de casos de uso.....	31
10	Integración Orientada a la Información.....	33
11	Integración Orientada a los Procesos de Negocio.....	34
12	Integración Orientada a los Servicios .....	35
13	Integración Orientada a los Portales .....	38
14	Transferencia de Archivos .....	38
15	Base de Dato Compartida .....	39
16	Invocación de procedimiento remota.....	40
17	Mensajería .....	41
18	Componentes de un Mensaje SOAP .....	64
19	Estructura de un Documento WSDL.....	65
20	Arquitectura Mule .....	68
21	Proceso de Integración SAP y Aplicación Web de Pago de Impuesto .....	80
22	Secuencia de actividades Elicitación de servicios .....	82
23	Secuencia de actividades para modelar Objetivos usando Framework i* .....	83
24	Diagrama de Objetivo .....	85
25	Modelo SD con I* .....	87
26	Modelo SR con I* .....	89
27	Secuencia de actividades para modelar casos de uso UML .....	90
28	Diagrama de casos de uso UML .....	91
29	Secuencia de actividades para crear arquitectura.....	101
30	Diagrama de Decisión para seleccionar el patrón correcto .....	104

31	Diagrama de Actividad Consultar Contribuyente .....	105
32	Diagrama de Actividad Consultar Parámetros de Calculo .....	106
33	Diagrama de Actividad Generar Calculo.....	107
34	Diagrama de Actividad Generar Factura .....	108
35	Diagrama de Actividad Compensar Pago.....	109
36	Diagrama de Actividad Procesar Archivos Bancarios .....	110
37	Diagrama de Actividad Generar Solvencia .....	111
38	Diagrama de Actividad Generar Constancia .....	112
39	Diagrama de Actividad Generar Indicadores.....	112
40	Diagrama de Actividad Consultar Tramite .....	113
41	Diagrama de Actividad Procesar Facturación.....	113
42	Diagrama de Actividad Actualizar Información .....	114
43	Flujo con Mule de Consultar Contribuyente .....	116
44	Flujo con Mule de Consultar Parámetros de Calculo.....	118
45	Flujo con Mule de Generar Cálculo .....	119
46	Flujo con Mule de Generar Factura .....	120
47	Flujo con Mule de Generar Pago .....	121
48	Flujo con Mule de Generar Compensaciones.....	122
49	Flujo con Mule de Procesar Reportes Bancarios .....	123
50	Flujo con Mule de Generar Solvencia .....	124
51	Flujo con Mule de Generar Constancia .....	125
52	Flujo con Mule de Generar Indicadores .....	126
53	Flujo con Mule de Consultar Trámite .....	127
54	Flujo con Mule de Procesar Facturación.....	128
55	Flujo con Mule de Actualizar Información .....	129
56	Arquitectura de Integración de Aplicaciones.....	131

UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”  
DECANATO DE CIENCIAS Y TECNOLOGÍA  
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

**DISEÑO DE UNA ARQUITECTURA DE INTEGRACIÓN DE  
APLICACIONES SOBRE LA BASE DE INTERCAMBIO DE DOCUMENTOS,  
CASO DE ESTUDIO ALCALDIA DEL MUNICIPIO IRIBARREN, ESTADO  
LARA**

**Autor (a):** Ambar A. Mendoza C.

**Tutor (a):** Ramón Valera

**RESUMEN**

Las organizaciones para permanecer competitivo en el mercado buscan constantemente estar en la vanguardia, para ello deben enfrentarse a nuevos desafíos y planear estrategias que le permitan alcanzar tal fin. El valor de la información real y oportuna es un instrumento necesario para que las empresas tomen decisiones acertadas. Es por ello que nuestra investigación se centra en el buen manejo de los datos, en diseñar una arquitectura de integración de aplicaciones sobre la base de intercambio de documentos, que permita que la información llegue al momento y al lugar que se requiera, de esta forma facilitar la toma de decisiones, realizar los procesos de forma transparente sin la intervención del personal y dar respuesta oportuna al cliente. Para dar solución al caso la presente investigación tiene como fin diseñar una arquitectura de integración acoplado un bus de integración (ESB) utilizando patrones de integración basados en Mensajes. Definiendo servicios resultantes de la interacción en casos de usos y de herramientas como el framework i\* involucrando actores y procesos del entorno organizacional. Dicha investigación se basará en técnicas y métodos de ingeniería de software, usando Programación Extrema (XP) como metodología ágil de desarrollo y generando un conjunto artefactos entregables, diagramados en modelos bajo UML y utilizando Mule ESB como plataforma de integración que permitirá diseñar los servicios resultantes.

**Palabras Clave:** Integración de Aplicaciones, mecanismos de integración, documentos, arquitectura.

## INTRODUCCIÓN

La Tecnología de la Información (TI) y la Ingeniería de Software, poseen herramientas que al aplicarlas mejoran la producción de software, es por ello que las organizaciones cada día hacen más uso de esta tecnología para mejorar sus procesos y poder dar a sus clientes respuestas rápidas, eficientes y a su vez mejore la toma de decisiones.

La tecnología es considerada una herramienta integrada en los procesos de negocios y es de vital importancia para la toma de decisiones.

Para que los procesos estén integrados las organizaciones recurren a los llamados sistemas ERP (Enterprise Resource Planning) que está compuesto por distintos módulos que integran las actividades empresariales, dichos módulos hacen uso de una base de dato compartida, esto permite una integración de procesos logrando facilitar la toma de decisiones, reducción de costos, un mayor control directivo y logrando el uso efectivo de los recursos.

En ciertos momentos las organizaciones se ven en la obligación de crear nuevos módulos o nuevas aplicaciones y requieren que los sistemas estén integrados con el sistema ERP para que los procesos trabajen en conjunto, dar respuesta rápida y eficiente al cliente. Existen situaciones en que las empresas necesitan realizar negocios con funciones particulares que no existen en aplicaciones de terceros especializadas en el mismo o el trabajo involucrado para solucionarlo con sistemas ERP resulta demasiado costoso. Por lo que deciden realizar sus funciones con los sistemas existentes dentro de la organización para funciones particulares, las aplicaciones para realizar este tipo de funciones en algunas organizaciones se encuentran aisladas, en plataformas diferentes por lo que requieren de un adecuado mecanismo que permita una acorde integración para facilitar la toma de decisiones y que sus procesos se realicen de manera transparente sin la intervención del personal y con una seguridad especial.

Por lo antes expuesto, surge la investigación para realizar un diseño de arquitectura para integrar aplicaciones basadas en documentos.

Este estudio se encuentra estructurado de cinco (3) capítulos:

Capítulo I - El Problema, donde se presenta el planteamiento del problema que da inicio a esta investigación, adicionalmente se mencionan los objetivos que pretenden ser alcanzados al finalizar la misma. Finalmente, se expone brevemente la justificación y los alcances que tendrá la investigación.

Capítulo II - Marco Teórico, se analizan y exponen los antecedentes y bases teóricas consideradas válidas y confiables que soportan el conocimiento de los conceptos tratados durante el desarrollo de la investigación. En este capítulo se da punto de partida para comprender la situación actual del problema, realizando investigaciones que permitan conseguir avales teóricos y los trabajos hechos con anterioridad que orientarán el estudio.

Capítulo III - Marco Metodológico, describe la metodología a utilizar durante el estudio, la cual contempla: diseño, métodos y fases del mismo que son necesarios aplicar para lograr resolver el problema planteado, es decir, describe las consideraciones metodológicas del proceso de investigación.

## **CAPITULO I**

### **EL PROBLEMA**

#### **Planteamiento del Problema**

La tecnología es considerada una herramienta integrada en los procesos de negocios y es de vital importancia para la toma de decisiones. Esta trae consigo grandes transformaciones en las organizaciones que la llevan a estar constantemente en la vanguardia y que la introducen en el mercado competitivo. Sobre innovación y competitividad Orellana (2010) considera que estas dos palabras son claves para toda organización las cuales le generan valor y dan a sus accionistas respuestas satisfactorias. Al hablar de respuesta satisfactoria a los accionistas se refiere a alcanzar los objetivos de toda organización, lograr alcanzar el nivel de competitividad necesaria.

Un recurso importante que repercute en la competitividad en las organizaciones, es el buen manejo de la información, debido a que es un sustento a la hora de tomar decisiones. Es por eso que no es solo poseer información, sino que debe ser de calidad y actualizada. Y para poder utilizar este recurso es necesario poder procesarla y transmitirla a quien la necesite de manera adecuada.

Según Heredero et al (2011), define la tecnología como "Un conjunto sistematizado de conocimientos aplicados a las diferentes áreas del ser humano, unidos para la consecución de un fin, que es la creación o invención de algo, que puede ser desde la fabricación o mejora de un producto hasta la simplificación o el cambio de un determinado proceso."

Suárez (2012) la define como la ciencia que estudia los medios técnicos y los procesos empleados en las diferentes ramas de la industria y de los negocios.

En este sentido podemos hablar de tecnología de información. Suárez (2012) define la tecnología de información como la ciencia que estudia las técnicas y procesos automatizados que actúan sobre los datos y la información. Considera que las tecnologías de la información están presentes en todos los niveles de nuestra sociedad actual, desde las más grandes corporaciones multinacionales, a las pymes, gobiernos y asociaciones, profesionales y particulares.

En este sentido podemos hablar de información como recurso estratégico para las organizaciones, y es necesario que la misma sea canalizada y transformada de forma adecuada para que sea aprovechada en la toma de decisiones. Por ello Caiceo, J (2012) hace énfasis que para aumentar el valor de la información es necesario integrarla, validarla, sintetizarla y dejarla disponible en el momento adecuado para ser analizada en el momento de la toma de decisiones.

Aprovechar el avance de la tecnología de información mejoraría dentro de las organizaciones el manejo y la difusión de la información. Cruz, F (2013) Existen sistemas como los Enterprise Resource Planning (ERP) que integran todas las áreas de gestión de las empresas, manteniéndolas a disposición de toda la organización, permitiendo el manejo eficaz de la información, logrando tomar decisiones acertadas en los momentos oportunos, gracias a la veracidad de los datos que se manejan mediante el ERP.

Rivera, I. (2013) define los ERP como "herramientas informáticas que utilizan las organizaciones para gestionar de mejor manera sus datos y por lo tanto sus recursos. Estos sistemas se caracterizan por contar con una base de datos común para todos o la mayor parte de los departamentos de la organización, lo cual ayudará a automatizar y sincronizar las transacciones de las diferentes actividades de la empresa."

Sin embargo hay situaciones dentro de las organizaciones que los sistemas ERP no son totalmente la solución. Badenes O. (2013) "Los ERP son aplicaciones informáticas genéricas concebidas para soportar la gestión de los diferentes procesos de cualquier tipo de organización". Sin embargo, existen organizaciones de sectores



industriales distintos, tienen procesos de negocio diferentes, que les son propios y no se adaptan a los planteados en este tipo de sistemas. Debido a ello, las empresas utilizan distintas aplicaciones desplegadas en diferentes tecnologías, lenguajes, plataformas como por ejemplo los sistemas ERP y otros sistemas necesarios dentro de la organización, dando como resultados funcionalidades aisladas, actividades manuales redundantes, costos más altos y respuestas ineficientes para sus clientes.

Para que entre los sistemas que intervienen en los negocios empresariales haya una adecuada comunicación y retroalimentación se requiere que dentro de la organización se utilice los avances de la tecnología para lograr una acorde Integración entre aplicaciones.

García (2012) La integración de aplicaciones empresariales, por sus siglas en inglés EAI (Enterprise Application Integration) es un término utilizado comercialmente para denominar los planes, métodos y herramientas utilizadas para integrar y coordinar las aplicaciones en una empresa. Ulric (2014) considera EAI como un enfoque que combina procesos, descargas, estándares y hardware para unir dos o más sistemas en conjunto, lo que les permite actuar como un solo sistema.

El problema de la integración de aplicaciones se está convirtiendo en una necesidad común en las organizaciones. Cuando se logra que las aplicaciones aisladas o heterogéneas trabajen en conjunto o compartan información se alcanza una mejor gestión dentro de la empresa, permitiendo una rápida y oportuna toma de decisiones y dándoles a los clientes respuesta a tiempo.

A raíz de esto y a que hoy en día el grado de competencia que existe en las empresas por sobresalir en el mercado, las organizaciones deben generar impacto de manera que sus clientes queden satisfechos con los bienes y servicios que ellos les ofrecen, por ello es muy importante manejar la información, integrar las aplicaciones para lograr una adecuada retroalimentación de los datos y que la información requerida llegue de manera oportuna a las manos de la persona adecuada.

En este punto, existen diversas soluciones de integración que lleva a la integración de aplicaciones empresariales. Soluciones que se centran en el uso de la tecnología para luchar en contra de las dificultades presentadas al tener aplicaciones

heterogéneas en las organizaciones, por ello es necesario estudiar los diferentes enfoques para lograr una integración de aplicaciones, de acuerdo a Hohpe y Woolf (2012), los diferentes enfoques se pueden resumir en cuatro estilos principales de integración, en otras palabras es el porqué y el para qué se integran las aplicaciones empresariales. Los elementos de este grupo se describen a continuación:

- Por Transferencia de archivo, cuando existen aplicaciones que intercambian archivos.
- Por Base de datos compartida, cuando las aplicaciones guardan información en un lugar común.
- Por Invocación de procesos remotos, cuando cada aplicación publica procedimientos para que puedan ser invocados remotamente, y al mismo tiempo otras aplicaciones hacer uso de estos y ejecutar una labor de sistema e intercambio de información.
- Por Mensajería, cuando cada aplicación se conecta a un sistema de mensajería común, comparte información y ejecuta funciones mediante el uso de mensajes.

Una aplicación puede utilizar diferentes estilos de integración con diferentes aplicaciones. Algunos métodos de integración pueden resultar como un híbrido de varios estilos.

Hohpe y Woolf (2012), consideran que los sistemas de mensajería, se muestran como los medio de comunicación más adecuado para la intercomunicación de aplicaciones empresariales. Los autores proponen varios patrones que ayudan a diseñar una solución de integración usando estilo de integración de mensajería como pilar principal para la integración.

Según los autores, “Los Mensajes son más inmediatos que transferencia de archivos, mejor encapsulado que base de dato compartida y más seguro que invocación de procesos remotos”.

Una solución de integración usando sistemas de mensajería permite tener, entre otras ventajas, bajo acoplamiento y comunicación asíncrona, acoplamiento es el conocimiento que una aplicación involucrada en la integración tiene sobre otra

aplicación y asíncrona permite que una aplicación puede enviar un mensaje a otra sin que la aplicación destino esté lista para recibirlo. Estas características son muy importantes a la hora de escoger un estilo de integración y diseñar dicha solución.

Por lo antes expuesto el presente trabajo de investigación propondrá una Arquitectura de Integración de Aplicaciones sobre la Base de Intercambio de Documentos. Para el cual se tomara como caso de estudio, la Alcaldía del Municipio Iribarren del Estado Lara, donde actualmente existe dificultad para realizar procesos de negocios, algunos de los cuales se presentan a continuación:

- Carencia de un buen mecanismo de integración de los sistemas que se manejan dentro de la Alcaldía del Municipio Iribarren del Estado Lara, problema que plantea el Coordinador de Informática del SEMAT (Servicio Autónomo de Administración Tributaria) mediante una entrevista informal realizada a su persona.

- La Alcaldía cuenta con una aplicación web que permite a los contribuyentes o ciudadanos del Municipio, realizar los trámites para el pago de las obligaciones con el municipio, el pago de los tributos, desde sus casas con el uso del Internet. Dicha aplicación cuenta con limitantes, las respuestas a los clientes no son inmediatas debido a que la misma no está en línea con el sistema administrativo SAP (Sistemas, Aplicaciones y Productos), sistema utilizado para realizar sus procesos administrativos.

Tomando en consideración las limitantes presentadas, se hace necesario un sistema eficiente y eficaz que se integre con el sistema administrativo SAP y demás aplicaciones, para mejorar la respuesta a los ciudadanos y de esta forma facilitarles a los mismos cumplir con las obligaciones con el municipio y de esta forma mejorar la toma de decisiones en el organismo gubernamental.

En consiguiente el presente trabajo de grado propone el diseño de una Arquitectura basada en una estrategia de integración de aplicaciones que permita mejorar el problema que se presenta en la Alcaldía del Municipio de Iribarren del Estado Lara, ya que la calidad de integración de sus aplicaciones es poco óptima.

Por lo antes expuesto se derivan las siguientes interrogantes.

¿Qué método es usado en la actualidad para el intercambio de información entre

el sistema ERP SAP y la aplicación web de pago de impuesto?

¿Cuáles son los requisitos que se deben cumplir para lograr un adecuado intercambio de información entre el sistema ERP SAP y la aplicación web de pago de impuesto?

¿Cómo estaría diseñada una arquitectura de integración de aplicaciones que logre satisfacer los requerimientos presentes en el caso de estudio, específicamente en la Alcaldía de Iribarren del Estado Lara?

Para responder estas interrogantes, se realizará investigación de tecnologías existentes, estudiarlas y evaluarlas con el fin de diseñar una arquitectura que permita integrar aplicaciones satisfactoriamente.

## **Objetivos de la Investigación**

### ***Objetivo General***

Diseñar una Arquitectura de Integración de Aplicaciones sobre la base de intercambio de documentos. Caso de estudio Alcaldía del Municipio Iribarren. Estado Lara.

### ***Objetivos Específicos***

Realizar un diagnóstico sobre la manera en que se produce el intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.

Realizar un análisis orientado a servicios para conocer los requisitos funcionales y no funcionales presentes para el buen intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.

Diseñar una arquitectura de integración de aplicaciones que satisfaga los

requerimientos presentes en el caso de estudio, específicamente en la Alcaldía de Iribarren del Estado Lara

### **Justificación e Importancia**

Las organizaciones se hacen más competitivas a medida que hacen uso de los avances tecnológicos para mejorar sus procesos y el manejo de la información con el fin de garantizar una adecuada toma de decisiones. La optimización de los procesos, el acceso a la información y documentación de la organización en tiempo real que facilite la toma de decisiones, va a depender del buen manejo de las aplicaciones y de los procesos, para ello se destaca el disponer de estrategias que garanticen la integración de aplicaciones para lograr una buena gestión dentro de la organización. Esta Integración de aplicaciones debe permitir solucionar problemas que acarrea la presencia de aplicaciones heterogéneas. Obteniendo aplicaciones integradas, la organización debe ser capaz de propiciar su buen funcionamiento manteniendo un buen manejo de sus procesos sin la necesidad de cambiar las aplicaciones que pertenecen a la integración.

El seleccionar una estrategia de integración adecuada para las organizaciones genera beneficios expresados en la optimización de los procesos, en mejorar la calidad de la información mejora la respuesta a las personas involucradas en forma oportuna y eficiente para el beneficio de toda la empresa y mantener un nivel de satisfacción. Esta investigación también se justifica desde el punto de vista práctico ya que los problemas antes mencionados ya se están presentando en la Alcaldía del Municipio Iribarren, Estado Lara, se hace evidente la necesidad de realizar este proyecto, cuya importancia se ve reflejada en los beneficios que traerá tanto a la organización como a los ciudadanos, mejorando los procesos, la toma de decisiones a la organización y para los ciudadanos haciéndole más fácil cumplir con sus obligaciones con el municipio usando la aplicación web.

Desde el punto de vista de ingeniería de software, esta investigación generará un aporte que podrá ser aplicado por desarrolladores de software, ya que de alguna manera u otra, se confrontan teorías como estrategias de integración, el diseño de una arquitectura, el manejo de documentos que podrán servir de guías para trabajos futuros.

### **Alcances y Limitaciones**

Este trabajo servirá como base para obtener una estrategia para la integración de aplicaciones usadas en la Alcaldía del Municipio Iribarren, proporcionando una solución de integración directa y sencilla cuya función principal es optimizar los procesos que intervienen en el ámbito de pago de impuestos municipales, logrando una mejor toma de decisiones y dar una rápida respuesta a los usuarios.

Además con este trabajo se proporciona una estrategia para tratar la integración de aplicaciones que sea útil para las organizaciones tanto públicas como privadas que manejen aplicaciones heterogéneas.

## CAPITULO II

### MARCO TEÓRICO

#### Antecedentes

En las organizaciones para mejorar el funcionamiento de sus procesos, se hace necesaria la inclusión de mecanismos que permitan integrar sus aplicaciones, permitiendo así mejorar la productividad, con procesos transparentes sin la intervención del personal, ya que actualmente se plantea lograr que las máquinas puedan leer e interpretar de forma automática la información, haciendo usos de sistemas que cataloguen y clasifiquen la información para facilitar y disminuir el tiempo y esfuerzo otorgado para resolver un problema.

Mediante investigaciones y experiencias de implantación se tiene que EAI permite la integración de aplicaciones dentro de la misma empresa, permitiendo mantener las aplicaciones funcionando en sincronía y de forma exógena para integrar sus funcionalidades.

Para dar inicio a los antecedentes de la investigación, se comienza con un trabajo de maestría denominado “**Integración de aplicaciones empresariales mediante el Enterprise service bus: caso de uso: Programa Nacional de Apoyo Directo a los Más Pobres-PCM**”, desarrollada por Aguilar (2010), la cual basa su investigación a Bus de Servicios Empresariales (ESB), que es el encargado de estandarizar la comunicación que viaja en él y orquestarlas funcionalidades del software para dar soporte a los procesos del negocio.

Para esta propuesta se utilizó la especificación Java Business Integration (JBI)

para Diseñar el Enterprise Service Bus que permita integrar las distintas aplicaciones de la organización y los servicios proporcionados por otras instituciones (salud, educación e identidad).

Aguilar en su trabajo de grado utiliza servicios y se hizo un estudio de ellos. Obteniendo la respuesta de que el uso de servicios puede transformar los procesos empresariales en servicios reutilizables y flexibles, lo que permite mejorar y optimizar los nuevos procesos. El uso del ESB es el punto de entrada para distintas aplicaciones para conectar la infraestructura con eficiencia, integrando a todas las personas, procesos e información de la organización. Al tener conexiones flexibles entre los servicios y el todo el entorno, se puede tomar un procesos de negocio ya existente y ofrecerlo sin mucho esfuerzo a través de otro canal empresarial.

De la investigación antes descrita, se toman las mismas motivaciones que justifican la presente investigación, por lo que proporciona al trabajo en estudio un marco de referencia para conocer el funcionamiento de ESB y los servicios.

López (2011), realiza una investigación referida a la integración de sistemas de información con el fin de generar beneficios o mejoras continuas dentro de una institución para ampliar la efectividad en sus procesos y dar respuesta oportuna en la toma de decisiones. El trabajo de maestría de López se centra en un caso de estudio realizado a la UNEXPO la cual cuenta con sistemas de información con buenas plataformas pero que no poseen la efectividad de dar respuestas oportunas a las distintas solicitudes. Es por ello que López presenta su trabajo titulado "**Propuesta de Integración de los sistemas de Información para la UNEXPO basada en ROA**" puesto que en la institución se hace inminente la necesidad de un enfoque de integración de aplicaciones basada en una arquitectura orientada a recursos. Apoyándose en el uso del paradigma REST enfocado en ROA que se ha hecho muy popular porque es un enfoque ligero que no introduce la sobrecarga de información de los mensajes SOAP y el protocolo HTTP que permite acceder a los recursos a través de internet.

López al finalizar su estudio concluye que es necesario lograr el aprovechamiento de la información que se encuentra en cada aplicación que hace



vida en la organización, por lo que la propuesta en una arquitectura enmarcada en una ROA habilita a las diferentes aplicaciones a trabajar conjuntamente en forma eficaz, con un rendimiento cercano al tiempo real.

Este trabajo de investigación aporta a la presente investigación la experiencia satisfactoria de integrar sus sistemas enmarcada en una arquitectura ROA que habilita a las diferentes aplicaciones a trabajar conjuntamente de forma eficaz.

Paco (2011) realiza una investigación referida al Intercambio de Información del CIAT (Centro Interamericano de Administraciones Tributarias.) y la OCDE (Organización para la Cooperación y el Desarrollo Económico), este artículo describe las nuevas tecnologías que pueden aplicarse para el intercambio automático de información, que constituye un medio eficiente y eficaz debido al costo beneficio que representan para el logro y cumplimiento de los objetivos y metas de las Administraciones Tributarias. Esta experiencia se constituye además en un valioso aporte, al momento de implantar un mecanismo de transferencia de información de manera formal, automática y sistemática con otras entidades gubernamentales u otros países. El artículo de Paco esta titulado **“Uso de Nuevas Tecnologías para el Intercambio Automático de Información (IAI) en Administraciones Tributarias a nivel intergubernamental e internacional”** publicada en revista de nombre (Revista de Administración Tributaria).

Paco se apoya en el uso de arquitectura SOA basado en servicios web porque son una combinación ideal de arquitectura y tecnología que manejan servicios robustos y reutilizables que soportan las necesidades de los negocios y que se pueden adaptar para satisfacer las necesidades cambiantes del negocio.

Paco al finalizar su artículo concluye que con el uso de la arquitectura SOA y los servicios web se puede realizar integración de sistemas, permitiendo el rápido y eficiente intercambio automático de información entre instituciones gubernamentales, sin que la plataforma, sistemas operativos ni lenguajes de programación sean un inconveniente, porque el formato de datos usados son estándar e interoperable.

Este artículo proporciona a este estudio un marco de referencia para conocer los beneficios que proporciona la arquitectura SOA basada en servicios web aplicada a

entidades gubernamentales.

Valera (2011), en su tesis titulada “**Construcción de un Motor de Búsqueda de Contenidos en Repositorios Confiables, Basado en Crawlers, Enmarcado en una Arquitectura Orientada a Servicio**”, busca desarrollar un motor de búsqueda en repositorios confiables, usando arquitectura orientada a servicios (SOA), acoplado un bus de integración (ESB) como middleware a través de un intercambio de mensajes estandarizados dando escalabilidad y reusabilidad empleando patrones de integración. Emplea tecnología NoSQL como almacén de datos; de esta forma dar respuesta a las necesidades de usuarios, en encontrar información de calidad en La Web y poder abarcar la información de La Web Invisible.

Valera en su trabajo para poder darle acceso a los componentes del motor de búsqueda a sistemas externos utilizó una Arquitectura Orientada a Servicios (SOA) y para darle escalabilidad a la arquitectura y permitir que distintas arquitecturas se puedan comunicar implementó el uso de un bus de integración (ESB). Por otro lado, para que los sistemas se puedan comunicar utilizó XML, sin embargo Valera considera que el uso de XML no es suficiente para que varios actores independientes puedan colaborar para conseguir sus objetivos. Para ello, requirió necesario definir una ontología, es decir, describir los conceptos y las relaciones de un dominio común para que pueda existir una comunicación entre los distintos actores.

Valera luego de finalizado su estudio concluyó que en base a SOA, ESB, NoSQL e Indexación en la búsqueda de información en la web garantizan el incremento y rapidez de respuesta del sistema. Y en base a los patrones de integración basados en mensajes, generaron una arquitectura basada en servicios con componentes desacoplados, colaborando en el proceso de orquestación y garantizando alguno de los requisitos no funcionales.

Este trabajo de investigación proporciona a este estudio un marco de referencia para conocer cómo trabajan en conjunto SOA, ESB y el uso de Patrones de integración basados en mensajes.

Rivero (2012), realizó una tesis titulada “**Propuesta de un modelo de integración de aplicaciones empresariales para CVA Empresa comercializadora**”

**de insumos y servicios s.a usando BPM y SOA**”, en la cual refiere su investigación en técnicas de integración como lo es Business Process Management (BPM), para el modelaje de los procesos y la Arquitectura Orientada a Servicios (SOA) para realizar cambios a los procesos desde el punto de vista tecnológico generando nuevos procesos que ayuden a mejorar las funcionalidades dentro de las organizaciones.

Rivero en su trabajo de grado utiliza servicios web para implementar SOA, al igual que ESB ya que forma parte de la Arquitectura SOA y para el manejo del transporte de los datos utilizo los protocolos SOAP y HTTP

Al finalizar su estudio concluyo que al trabajar con SOA se obtienen componentes como servicios tales como servicios web, modelar los procesos de negocios de la empresa lo cual brinda la posibilidad de integrar los procesos haciendo uso de los sistemas existentes en una organización. Y al trabajar con BPM se logra plasmar el flujo de trabajo del negocio de una organización facilitando el entendimiento del funcionamiento de los procesos de negocios.

Y al combinar ambos paradigmas se logra diseñar un modelo de implantación de aplicaciones donde se observa una arquitectura basada en servicios y el conocer el procedimiento lógico de los procesos de negocios.

Este trabajo de investigación proporciona al trabajo en estudio un marco de referencia para conocer cómo trabajan en conjunto SOA con BMP y el manejo de los servicios web.

Continuando con la revisión de antecedentes, Torres (2013) en su investigación denominada “**Modelo de integración de aplicaciones orientado a objetivos e ingeniería de métodos**”, plantea un modelo en busca de mejoras en los procesos de comunicación entre unidades organizacionales y sus sistemas heterogéneos, apoyándose en el uso de herramientas que permitan establecer con claridad cuáles son las interacciones presentes y necesarias entre ellas, y en función de estas establecer un punto de partida para un esquema integrador, destacándose la necesidad de una visión amplia para obtener un esquema general de la organización.

Torres aclara que una solución de integración debe garantizar que a pesar que se requiere integrar datos y procesos, esto debe lograrse por medio de opciones que

permitan mantener la independencia operativa de las unidades, por lo que en su investigación utilizo los servicios como la transformación de aquellas funcionalidades descubiertas producto del refinamiento de objetivos, estos representan funcionalidades útiles, expuestas para ser consumidas, combinadas y reutilizadas.

En dicha investigación para identificar los objetivos y funcionalidades torres utilizó herramientas de diagramado tales como i\* que destaca por su riqueza gráfica y adaptabilidad, así como UML para el diagramado de casos de uso en función de la licitación de procesos y actores que interactúan en torno a la satisfacción de objetivos organizacionales.

Al finalizar su investigación torres elaboro un modelo de integración Orientado a Objetivos e Ingeniería de Métodos que presenta por su carácter genérico características que lo hacen aplicable a diversidad de contextos organizacionales.

Este trabajo de investigación proporciona al trabajo en estudio un marco de referencia para conocer el uso de herramientas como i\*, UML para el desarrollo de un modelo de integración orientado a objetivos e Ingeniería de Software y conocer la aplicabilidad que tiene este tipo de modelos en contextos organizacionales y como herramienta para definición de los procesos, elemento a ser utilizada por la presente investigación en la transformación de estos en servicios a ser apoyados en la arquitectura a diseñar.

## **Bases Teóricas**

Las bases teóricas sustentan la investigación sobre lo referente a integración de aplicaciones, para ello se hace uso de las teorías que se relacionan con el trabajo especial de grado, como lo es concepto de integración de aplicaciones, concepto de EAI, tipos de EAI, estilos de integración, patrones de integración, servicios web entre otros conceptos de interés que serán de gran utilidad para la presente investigación.

## *Tecnologías de Información y comunicación (TIC)*

Las tecnologías de la información y la comunicación (TIC) están presentes en todos los niveles de la sociedad, desde las más grandes corporaciones multinacionales, a las pymes, gobiernos, administraciones, universidades, centros educativos, organizaciones socioeconómicas y asociaciones, profesionales y particulares. La tecnología se ha convertido en imprescindible para muchas personas y empresas.

Pablos C. (2004) define a TIC como un conjunto sistematizado de conocimientos aplicados a las diferentes áreas del ser humano, unidos por la consecución de un fin, que es la creación o invención de algo, que puede ser desde la fabricación o mejora de un producto hasta la simplificación o el cambio de un determinado proceso.

Balado E. (2005) define a TIC como un conjunto de tecnologías y recursos asociados a los sistemas de información y comunicación, es el conjunto de tecnología que nos asegura la gestión eficiente de la información que se genera en una empresa.

La Adopción de TIC genera beneficios como:

- Incremento en el rendimiento de la empresa
- Capacidad de la empresa para adaptarse e innovar
- Permite mejorar la productividad
- Persigue la consecución de objetivos respecto al tratamiento de la información: confiabilidad, eficiencia, efectividad, cumplimiento de la legislación vigente, confidencialidad, integridad y disponibilidad

Suárez R. (2010) por su parte define la TIC en términos que la componen como son:

Tecnología: Es la ciencia que estudia los medios técnicos y procesos empleados en las diferentes ramas de la industria y de los negocios.

Tecnología de la Información: Es también llamada informática, es la ciencia que estudia las técnicas y procesos automatizados que actúan sobre los datos y la información

Tecnología de la comunicación estudian las técnicas y los procesos que

permiten el envío y la recepción de información a distancia.

Todos los conceptos evaluados coinciden en que la tecnología de la información y la comunicación es el conjunto de tecnología que nos asegura la gestión eficiente de la información que se genera en una empresa. Es un aspecto clave para el desarrollo de las organizaciones, dando indudablemente beneficios económicos, sociales y culturales.

Estos conceptos serán de utilidad en esta investigación para determinar cómo, dónde y con qué tecnologías implementar una arquitectura que se adapte o satisfaga las funcionalidades de una organización permitiendo alcanzar beneficios y objetivos satisfactorios.

En el entorno actual, en la que las decisiones a tomar deben ser rápidas, minimizar riesgos e incrementar los beneficios es necesario que la tecnología y los procesos trabajen de forma conjunta. Las organizaciones deben centrar su esfuerzo en optimizar los procesos haciendo uso del mejor recurso disponible que es la tecnología.

### ***Sistemas de Planeación de Recursos Empresariales (ERP)***

González L. (2004) Es un software integrado que permite a las empresas evaluar, controlar y gestionar fácilmente su negocio en todos los ámbitos. Son un sistema de planificación de los recursos y de gestión de la información que, de una forma estructurada satisface la demanda de necesidades de la gestión empresarial.

Kendall K. (2005) Define ERP como un término empleado para describir un sistema de información empresarial integrado, es un software que ayuda al flujo de información entre las áreas funcionales de la organización.

Martínez E. (2014) Las empresas como sistemas están relacionados con su entorno, del cual reciben entradas en forma de recursos (humanos, financieros, materiales, entre otros.) estas entradas son transformadas en forma de productos y/o servicios. Para garantizar una buena gestión de los procesos las empresas deben contar con un sistema de gestión que permita la planificación, organización, información y control empresarial, dichos sistemas son los llamados ERP, que son

sistemas de información que integra aplicaciones informáticas para gestionar todas las funciones y áreas de una empresa como pueden ser los departamentos de contabilidad, ventas, compras, recursos humanos, financiero, gerencia, producción entre otros.

Los principales objetivos de un sistema ERP son:

- Optimización de los procesos
- Integridad de Datos, acceso a la información de forma fiable, precisa y oportuna.
- Posibilita compartir información entre todos los departamentos.
- Elimina Datos Innecesarios.
- Consolida datos y procesos.

De lo anteriormente planteado se puede decir que los ERP son las soluciones de referencia para las empresas. Los ERP permiten alcanzar la eficiencia corporativa y proporcionan beneficios en la coordinación, la comunicación y la gestión de procesos de la organización. Además, permiten adaptar los entornos de trabajo, así como apoyar a la toma de decisiones en tiempo real.

Es importante destacar, que los sistemas ERP no cubren en totalidad las funcionalidades de algunas organizaciones con características muy particulares, lo cual condiciona la permanencia o la necesidad del uso de otras aplicaciones, que de alguna manera cohabitan dentro de la organización sin integrarse al resto; por tal razón se podría inferir que en algunos casos la integración con esta solución no es total, comprometiendo la necesidad de crear una arquitectura de integración que permita la integración de este tipo de sistemas con otros tipos de aplicaciones existentes en las organizaciones.

### ***Metodologías de Desarrollo***

Como se dijo anteriormente las organizaciones necesitan aplicar TIC para ser más competitivas y eficientes, pero además necesitan combinar estas tecnologías

adecuadamente con otros recursos para poder alcanzar tal fin. Para ello debe crear sistemas de información basándose en metodologías que permitan desarrollar e implantar las diferentes herramientas de un sistema alcanzando una aproximación estructurada y por parte, reduciendo las complejidades y los costes de los mismos. Con un enfoque metodológico se plantean soluciones de modo ordenado, especificando las situaciones a satisfacer, planeando modelos que permitan resolver los problemas, desarrollando esos modelos y probándolos.

De Pablos C. (2012) Define las metodologías de desarrollo como el conjunto de técnicas, modelos y métodos que permiten planificar y controlar proyectos de creación de sistemas de información y desarrollos de software.

Pérez D. y otros (2011) Existen diversas metodologías, las que se agrupan en ágiles y tradicionales. Las metodologías ágiles promueven las relaciones interpersonales, son adaptativas, se enfocan en las personas, caracterizándose por existir una estrecha colaboración entre el cliente y el equipo de desarrollo. Programación Extrema (XP) se encuentra entre las metodologías ágiles más utilizadas. Parra E. (2011) considera que la programación extrema (XP) se caracteriza porque los requerimientos del software se plantean como escenarios llamados historias de usuarios, los cuales se implementan como una serie de tareas que deben desarrollarse. Aplicando esta metodología los programadores trabajando en parejas hacen pruebas antes de escribir los programas. En general XP implica las siguientes prácticas (Sommerville, 2005):

- a. Entregas pequeñas y frecuentes a los clientes, en vez de la entrega del sistema en su totalidad.
- b. Participación directa de los clientes o sus representantes en el equipo de desarrollo del proyecto. Además, ellos son responsables de la aceptación de pruebas aplicadas al software.
- c. Cuidado en el proceso de desarrollo, de manera que se priorice el trabajo en parejas, la responsabilidad colectiva sobre el programa y racionalización de las cargas de trabajo evitándose jornadas excesivas.
- d. Desarrollos previamente probados de las entregas, antes de su integración al



sistema global.

- e. Utilización de diseños sencillos y reutilización de programas o partes de éstos.

Para el desarrollo de esta investigación y el diseño de la arquitectura de integración de aplicaciones se utilizara este tipo de metodología por considerarse la más utilizada y por permitir realizar entregas pequeñas y frecuentes que se pueda fácilmente identificar y arreglar de manera rápida y sencilla.

### ***Ingeniería de Requisitos Orientada a Objetivos***

Baixauli1, B (s.f.) La Ingeniería de Requisitos orientada a objetivos es un enfoque que promueve la utilización de objetivos como base para obtener los requisitos y de esta forma conocer el propósito del sistema, permite que los interesados expresen sus necesidades de una manera natural, centrándose en los objetivos. A partir de los objetivos, los requisitos se pueden derivar como maneras de alcanzar esos objetivos.

Thomas P. (2005) La Ingeniería de Requisitos busca producir una descripción detallada de un problema, con el fin de construir un Sistema de Software, que satisfaga las “necesidades y objetivos” de la organización donde funcionará dicho sistema. Estos objetivos constituyen el fundamento del sistema, y son usualmente definidos como las metas a ser cumplidas por el sistema y su entorno.

Cuando se va a construir un sistema de software no solo basta con conocer un lenguaje de programación, también es necesario analizar el problema y diseñar la solución cuidadosamente y así entender que se tiene que desarrollar sino también porque se está desarrollando un sistema dado.

Para ello se debe seguir un proceso de desarrollo de software que permita planificar las distintas actividades y la forma de relacionar los productos involucrados, esto se logra por medio del enfoque Orientación a Objetivos de Ingeniería de Requisitos que mediante un proceso de refinamiento y descomposición de objetivos

del sistema es útil en la elicitación y definición de requisitos.

Sobre elicitación y definición de requisitos Teruel M. (2011) considera la Ingeniería de Requisitos, el enfoque orientado a objetivos las técnicas de análisis más tradicionales, se centran en el porqué de la construcción de un sistema, proporcionando la motivación y la lógica para justificar los requisitos del software. No solo son útiles para el análisis de objetivos, sino también para elaborarlas y refinarlas.

La Ingeniería de Requisitos orientada a objetivos es definida por Teruel M. (2011) como un enfoque que promueve la utilización de objetivos como base para obtener los requisitos, incorporando el propósito del sistema, un punto de vista donde los interesados expresen sus necesidades de una manera más natural, centrándose en los objetivos. Por tanto, los objetivos ayudan a identificar, justificar y organizar los requisitos.

Según Teruel es posible verificar la completitud de un conjunto de requisitos con respecto a las metas u objetivos planteados, mediante el uso de un grafo dirigido por medio de un refinamiento de los objetivos de los sistemas, para posteriormente, refinarlos en sub metas o sub objetivos mediante un árbol AND/OR. El refinamiento finaliza cuando cada sub meta o sub objetivo es asignable a un agente capaz de realizarla, normalmente un agente software.

Existen diversos enfoques dentro de la IR orientada a objetivos, de los cuales a continuación se describen 3 enfoques: NFR framework, KAOS e i\*framework.

### ***Enfoques de Ingeniería de Requisitos Orientada a Objetivos***

- ***NFR framework***

Teruel y otros (2011) señalan que el framework NFR tiene como objeto hacer frente a los requisitos no funcionales (NFRs), también conocidos como Requisitos de Calidad. Los NFRs especifican restricciones sobre el sistema, así como consideraciones particulares de los factores de calidad que un sistema debería cumplir, como son la precisión, usabilidad, seguridad, rendimiento, fiabilidad o seguridad. Al

contrario que los requisitos funcionales, Por lo tanto, se puede afirmar que mientras que los requisitos funcionales describen “qué” debería hacer el sistema, los NFRs establecen restricciones sobre “cómo” realizar esas tareas.

Este Framework ve a los objetivos NFRs como objetivos que pueden tener conflicto entre ellos y que deben ser representados como objetivos del software que deben ser satisfechos.

Cada objetivo de software se descompone en objetivos menores, representados por una estructura de grafo inspirada en árboles AND/OR usados en la solución del problema. Este proceso continua hasta que el ingeniero de requerimiento considere que los objetivos del software son satisfechos a nivel operacional.

Los elementos que componen al NFR framework son los siguientes:

- **Softgoal:** es un objetivo para el cual su satisfacción no siempre se puede establecer de manera clara. Así, un softgoal puede considerarse alcanzado o no, pero también parcialmente satisfecho y parcialmente insatisfecho. Por tanto, no se habla de satisfacción sino de grado de satisfacción. Representado por nubes de trazos finos.
- **Personalización (Operationalization):** permite enlazar los requisitos funcionales con los no funcionales. Representado por nubes de trazos gruesos.
- **Afirmaciones (Claim):** modifican las correlaciones según sea cierta o no la afirmación. Representado por nubes de trazos punteadas.
- **Contribution Link:** son categorizados como contribuciones AND OR. Estos enlaces permiten los NFRs hasta el punto en el que se puede afirmar que las operacionalizaciones de los NFRs relacionados se han cumplido. Las relaciones AND se representados por líneas con un arco y las OR con dos arcos.
- **Enlaces de Correlación (Correlation Link):** Estos enlaces son la base para calcular el grado de satisfacción (BREAK, HURT, HELP, MAKE).

En la figura No. 1 pueden verse los elementos que conforman el NFR

framework.

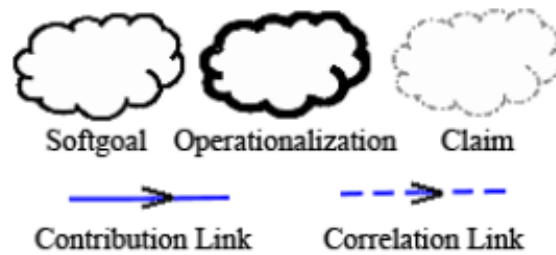


Figura N° 1. Elementos de NFR framework.

Fuente: Teruel M. (2011 p.36).

- **KAOS**

Adquisición del Conocimiento en una Especificación Automatizada (KAOS) es una metodología de elaboración de requerimientos que provee un lenguaje de especificación multiparadigma y un método de elaboración orientado a metas u objetivos. KAOS esta desarrollado para la obtención, indicación y análisis de los objetivos, requisitos, escenarios y asignación de responsabilidades.

Está compuesto por varios elementos descritos a continuación

- **Objetivo (Goal):** un objetivo describe un conjunto de comportamientos admisibles del sistema y deben ser definidos de una forma clara para que puedan verificar si el sistema lo satisface o no.
- **Softgoals:** son utilizados para documentar las preferencias entre los comportamientos de sistemas alternativos, los objetivos se considerados aceptables ya que no existe un criterio exacto que defina su grado de satisfacción.
- **Agentes (Agent):** Un agente puede ser un humano, un dispositivo o un componente software. Se define como un componente activo del sistema que busca satisfacer un objetivo dentro del sistema.
- **Descomposición AND / OR:** relacionan un objetivo con un conjunto de sub-objetivos. Considera un objetivo satisfecho si todos los sub-objetivos se cumplen (AND) o al menos uno de ellos es satisfecho (OR).

- **Conflicto potencial:** es un enlace que Indica que satisfacer un objetivo puede evitar alcanzar que otro objetivo cumpla la meta. Bajo algunas condiciones.
- **Asignación de Responsabilidad:** es un enlace que representa la responsabilidad de un agente frente a la satisfacción de un objetivo.

La figura No. 2 representa la construcción básica del framework KAOS identificando los elementos que lo conforman.

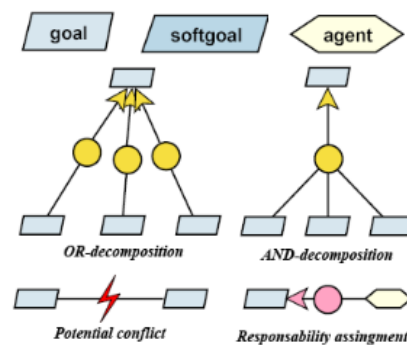


Figura N° 2. Construcción básica del framework KAOS.

Fuente: Teruel M. (2011 p.37).

- **Framework I\***

El framework i\* consiste en un enfoque Orientado a Objetivos que permite expresar en forma clara las metas de los actores, centrado en la elicitación de requisitos en las fases iniciales de la Ingeniería de requisitos (IR), es decir en el modelado de negocio. El framework i\* define varias fases en el proceso de desarrollo como son:

- *Fases de análisis de requisitos temprana:* Durante esta fase el ingeniero de requisitos obtiene y analiza las intenciones de los stakeholders. Esos requisitos son modelados como objetivos, los cuales, a través de algún análisis de objetivos, derivarán en requisitos funcionales y no funcionales del futuro sistema. En i\*, se asume que los requisitos tempranos involucran a actores sociales que dependen los unos de los otros para

satisfacer objetivos, realizar tareas, y utilizar recursos.

- *Fase de análisis tardío*: Deriva en una especificación de requisitos que describe los requisitos funcionales y no funcionales del futuro sistema.
- *Fase de diseño arquitectónico*: se elegirá entre distintos estilos arquitectónicos usando como criterio las cualidades deseadas identificadas previamente en el proceso.
- *Fase de diseño detallado*: se introducen detalles adicionales sobre los componentes arquitectónicos del sistema.

El framework i\* cuenta con una notación gráfica con pocos elementos, que permite tener una vista unificada del negocio. Descritas a continuación

- **Objetivos (Goal)**: objetivo de alto nivel de la organización y del sistema. Capturan el propósito del sistema. Diferenciados en dos tipos:

**Hardgoal**: objetivo funcional del sistema. Se caracteriza por la posibilidad de verificar la satisfacción o el no cumplimiento de un objetivo.

**Softgoal**: objetivo que solo se podrá afirmar que se ha cumplido o incumplido de manera parcial.

- **Actores**: se define como una entidad que existe en la organización que tiene objetivos y que puede realizar tareas o utilizar recursos para alcanzar dichos objetivos o ayudar a otros a alcanzar sus objetivos. Es destacable además la posibilidad de especialización de los actores en:

**Roles**: caracterización del comportamiento de un actor dentro de un contexto de dominio especializado, representa un estado de generalización.

**Agente**: representa un actor (personas, hardware, software) con características concretas que lo diferencian de los demás, siendo un estado de especialización.

**Posición**: representa un punto intermedio entre Roles y Agentes, utilizada para representar funciones desempeñadas por agentes para cubrir determinados roles.

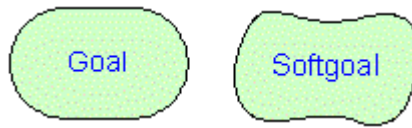


Figura N° 3. Representación de Objetivos en i\*.  
 Fuente: [http://istar.rwth-aachen.de/tiki-view\\_articles.php](http://istar.rwth-aachen.de/tiki-view_articles.php)

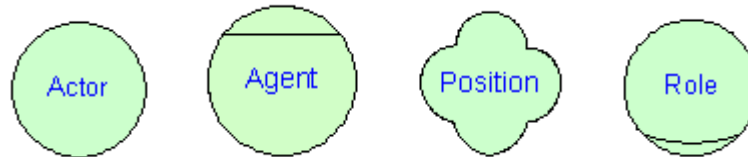


Figura N° 4. Representación de Actores en i\*.  
 Fuente: [http://istar.rwth-aachen.de/tiki-view\\_articles.php](http://istar.rwth-aachen.de/tiki-view_articles.php)

- **Tarea (Task):** representa la manera de alcanzar un objetivo. Puede ser de bajo nivel como un algoritmo o un procedimiento, o de un mayor nivel como un dispositivo, entorno GUI.

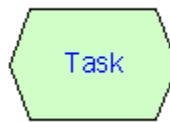


Figura N° 5. Representación de Tareas en i\*.  
 Fuente: [http://istar.rwth-aachen.de/tiki-view\\_articles.php](http://istar.rwth-aachen.de/tiki-view_articles.php)

- **Recursos (Resource):** representa una entidad física o información, objeto que es necesario para realizar una tarea o para alcanzar un objetivo

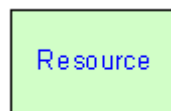


Figura N° 6. Representación de Recursos en i\*.  
 Fuente: [http://istar.rwth-aachen.de/tiki-view\\_articles.php](http://istar.rwth-aachen.de/tiki-view_articles.php)

El framework i\* está formado por modelos que se complementan para

proporcionar una visión completa de la naturaleza de la organización. Se describen a continuación:

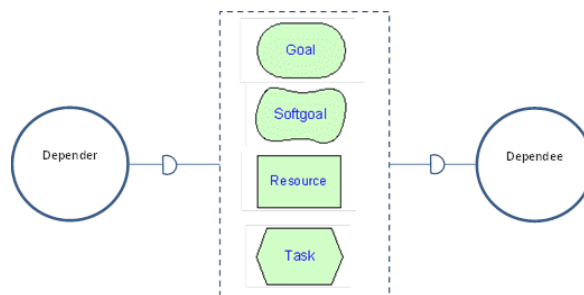
El Modelo de Dependencias Estratégicas (SD): modela las dependencias que existen entre los actores de un proceso de negocio. El modelo se representa gráficamente a través de un grafo con elementos como los actores, las dependencias entre actores, las relaciones externas entre los actores.

Actores primitivos:

- a) depender: el actor dependiente.
- b) dependee: el actor del cual se depende.
- c) dependum: el objeto alrededor del cual se centra la relación de dependencia.

Relaciones de Dependencias:

1. Dependencia de Meta, se depende que un actor realice una actividad, pero sin prescribir la forma en la que debe ser realizada.
2. Dependencia de Recurso, se depende de que otro actor realice la entrega de un recurso que puede ser material o informacional.
3. Dependencia de Tarea, se obtiene de las operaciones donde el actor prescribe la forma en que debe ser realizada la actividad.
4. Dependencia de Meta-suave es semejante a la dependencia de meta, con la diferencia de que no es posible establecer con precisión el criterio de aceptación de la meta.



*Figura N° 7. Relaciones de Dependencia en i\**  
*Fuente: Torres J. (2011 p.87).*

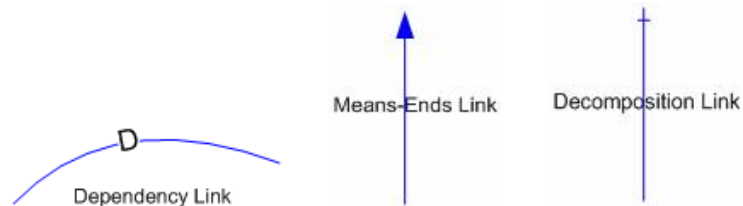
El Modelo de Razones Estratégicas (SR): es utilizado para describir los intereses y motivaciones de los participantes, Habilitar la valoración de posibles



alternativas en la definición de procesos y especificar con mayor detalle las razones de la existencia de dependencias entre varios actores.

Este modelo extiende al modelo de dependencias estratégicas incorporando dos tipos de constructores que permiten relacionar los objetos (objetivos, actores, tareas, recursos) entre sí mediante el siguiente conjunto de relaciones. Se describen a continuación dichos constructores:

- **Means-end links:** representa las diversas alternativas que pueden tomarse para lograr una meta o una tarea.
- **Decomposition links:** permite detallar un conjunto de actividades para lograr un objetivo.
- **Contribution links:** representa elementos que colaboran de forma positiva o negativa con la resolución de una tarea o proceso.



*Figura N° 8. Constructores en i\**  
*Fuente: Torres J. (2011 p.88).*

Este enfoque representa de manera gráfica los objetivos estratégicos, decisionales e informacionales de cada actor que interviene en los procesos y las relaciones entre los mismos.

Con base a las características anteriormente mencionadas se puede decir que el framework *i\** proporciona una visión completa de la naturaleza de la organización, mostrando dentro de sus diagramas actores, dependencias, recursos y tareas, información que resulta útil para visualizar cuáles son las tareas que se esperan que cada actor desempeñe, así como para analizar la repercusión que tiene el cumplimiento o incumplimiento de las tareas asignadas a los actores.

Es de destacar, según el estudio realizado por Teruel referenciado por (Torres, ob.cit), “*i\** resulta el más apropiado para sistemas colaborativos, lo cual lo hace su

aplicación apropiada para la presente investigación”.

### ***Casos de Uso***

La técnica más importante al momento de construir un sistema es la extracción iterativa y el refinamiento de los requerimientos del producto. Una mala definición de las necesidades funcionales afecta el mal funcionamiento del sistema y es difícil de corregir más adelante. Los casos de uso son la técnica utilizada por los ingenieros de software para el descubrimiento de requerimientos.

Fernández V. (2006) considera los casos de usos como la técnica de modelado que ha alcanzado popularidad ya que permite estructurar los requerimientos de forma que sean comprensibles tanto por los propietarios y usuarios del sistema como para los diseñadores y analistas. Permite modelar las funcionalidades de un sistema en términos de eventos, quien inicia y como responde el sistema a estos eventos.

El modelado de casos de usos está formado por dos elementos:

- **Diagrama de casos de usos:** muestra el comportamiento del sistema a partir de los usuarios que interactúan con el sistema. Este diagrama describe de forma grafica quien utiliza el sistema y la forma en que los usuarios esperan interactuar con el sistema. Se requiere de herramientas especializadas y estandarizadas para ellos estos diagramas siguen la especificación de Lenguaje de Modelado Unificado (UML). UML es el estándar de facto para el modelado orientado a objetos. Los diagramas de casos de usos cuentan con tres elementos:
  - *Casos de Uso:* es el primer elemento que contiene un modelo de casos de usos, el mismo caso de uso (use case). Describen las funciones básicas o simples del sistema desde la perspectiva de los usuarios externos. Se representa gráficamente a través de una elipse.
  - *Actores:* es un elemento externo que interactúa con el sistema. Representa un papel mientras que un individuo o sistema externo representa uno o varios papeles al mismo tiempo. Son los encargados de iniciar los casos de uso que representan las actividades que el sistema de información

debe realizar.

- *Relaciones:* representa la interacción entre un actor y un caso de uso. Los actores y los casos de usos se interconectan a través de distintos tipos de líneas.
  - *Asociación:* representa la interacción entre un actor y un caso de uso. Gráficamente es representada a través de una línea sólida entre un actor y un caso de uso.
  - *Inclusión:* Relación que define una instancia de un caso de uso *como* un curso obligatorio en otro caso de uso. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<include>>.
  - *Extensión:* Un caso de uso extiende a otro cuando sin alterar a este, se incorpora su funcionalidad como parte integral del primero. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<extend>>.
  - *Generalización:* Relación que define un caso de uso como una generalización de otro caso de uso. Un caso de uso hereda el comportamiento y significado de otro caso de uso. Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de “**Uso**” (<<uses>>) o de “**Herencia**” (<<extends>>).

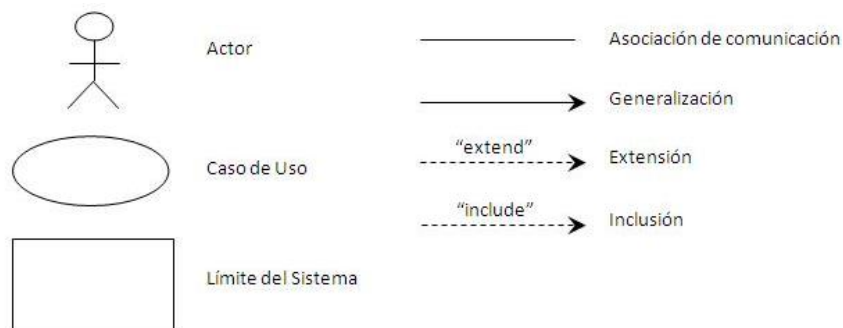


Figura N° 9. Elementos del Diagrama de casos de uso  
Fuente: Torres J. (2011 p.88).

- **Narraciones de casos de usos:** describen de forma escrita los eventos de negocio y cómo interactúan los usuarios con el sistema, es decir la descripción paso a paso de las actividades que conforman el caso de uso. Se describe a través de una tabla, UML no especifica un formato rígido; puede modificarse para atender las necesidades y lograr un conocimiento más profundo de los procesos y de los requerimientos.
  - *Caso de uso:* nombre del caso de uso.
  - *Actores:* lista de actores en la que se indica quién inicia el caso de uso.
  - *Tipo:* que puede ser Primario, Secundario, Opcional, entre otros.
  - *Descripción:* repetición del caso de uso de alto nivel o alguna síntesis similar.
  - *Curso Normal:* describe los detalles de la conversión interactiva entre los actores y el sistema. Explica la secuencia normal de las actividades y el término exitoso de un proceso. No incluye situaciones alternativas.
  - *Cursos Alternativos:* describe importantes opciones o excepciones que pueden presentarse en relación al curso normal. Si son éstas son complejas, se pueden expandir y convertir en nuevos casos de uso.

<b>Ref.:</b>	
<b>Caso de Uso:</b>	
<b>Descripción :</b>	
<b>Pre-condición:</b>	
<b>Curso Normal</b>	
<b>Curso Alterno</b>	
<b>Post-condición:</b>	

Cuadro N°1. Plantilla de narración de Casos de Uso

Fuente: El Investigador

En función de lo anterior, se utilizara caso de uso para especificar los requisitos en función de las necesidades de los usuarios, los casos de uso representan las

funciones que el sistema puede ejecutar.

### ***Integración de Aplicaciones (EAI)***

La clave para obtener agilidad de negocio es una integración eficiente. Vera M. (2007) diferentes programas funcionando en diferentes áreas, tiene la capacidad de compartir y transferir información entre sistemas, es lo que se llama Integración de Aplicaciones Empresariales (EAI).

Una solución de integración según Frantz (2008) es una solución de integración que mantiene un conjunto de aplicaciones sincronizadas, permite conectar dos o más aplicaciones, Es decir, una solución EAI aporta funcionalidad de una aplicación a la otra, o incluso, nuevas funcionalidades, además no es necesario modificar las aplicaciones para que sean integradas, cada aplicación se mantiene independiente.

El intercambio de información según el Club-BPM (2010) entre diferentes aplicaciones (nuevas, existentes heredadas o los paquetes de software), y procesos de negocios dentro y entre organizaciones proporcionan las funcionalidades necesarias para soportar los procesos de la empresa y permite mantener el ritmo de los cambios del mercado y del negocio.

Para el proceso de integración existen diferentes formas de integrar aplicaciones que se adecuan de acuerdo a las necesidades que existen dentro de las organizaciones, se describen a continuación los diferentes tipos de integración según una investigación realizada por Risi (2007):

- **Integración Orientada a la Información.** Consiste en el paso de información o mensajes de un sistema a otro.



*Figura 10.* Integración Orientada a la Información.  
Autor: RISI (2007)

La integración orientada a la información no requiere que las aplicaciones que se han integrado se modifiquen, las mantiene intactas para que sean independientes solo se requiere la implementación del mecanismo del paso de información entre los repositorios de datos de las aplicaciones respectivas.

Este tipo de integración no necesariamente esto requiere que tal integración se realice pura y exclusivamente usando tecnología de base de datos. También podría realizarse mediante archivos planos, APIs de las aplicaciones, o incluso servicios de mensajería. La clave de este tipo de integración no está en el medio técnico, sino en el hecho de que lo que se integra es información y no procesos o servicios.

•**Integración Orientada a Procesos.** Trata sobre integración orientada a los procesos de negocio, automatiza los diferentes pasos de un proceso de negocio a través de una o más aplicaciones.

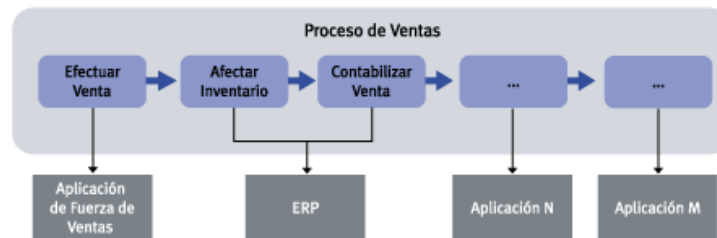


Figura 11. Integración Orientada a los Procesos de Negocio.  
Autor: RISI (2007)

•**Integración Orientada a Servicios.** Es la integración orientada a servicios, este tipo de integración es la más usada hoy en día. En este modelo, una aplicación expone una serie de servicios de negocio que pueden ser usados por otras aplicaciones. Este tipo de integración busca la reutilización, que la lógica de negocio sea implementada una única vez.



Figura 12. Integración Orientada a los Servicios.  
 Autor: RISI (2007)

A partir de este tipo de integración se crean las llamadas aplicaciones compuestas, que son las que surgen de los servicios existentes en la organización. La orientación a servicios también permite la creación de las llamadas Aplicaciones Compuestas. Nuevas aplicaciones que surgen a partir de la unificación de diferentes servicios preexistentes en la organización.

La orientación a servicios y la orientación a procesos no son excluyentes, sino complementarias. El caso ideal de la integración es aquel en el cual los procesos de negocio pueden conectarse a partir de diferentes servicios, que satisfacen las necesidades de los distintos subprocesos.

• **Integración Orientada a los Portales.** Este tipo de integración consiste en el uso de una interfaz visual para poder agrupar varias aplicaciones. Esta interfaz normalmente es un portal web.

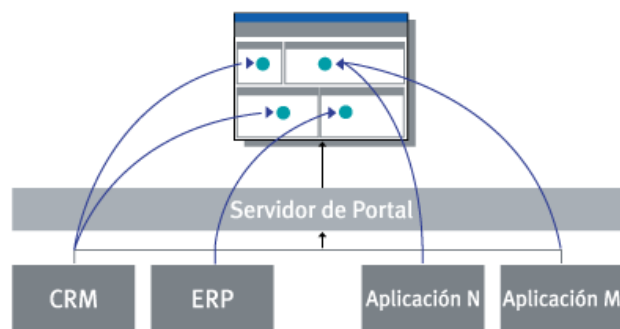


Figura 13. Integración Orientada a los Portales.  
 Autor: RISI (2007)

La integración mediante portales tampoco es excluyente de los otros tipos de integración. El portal Web puede alimentarse a través de servicios, y al mismo tiempo puede soportar la participación de actores humanos en procesos de negocio.

En el mismo orden de ideas, Integración de la empresa es la tarea de hacer que aplicaciones independientes trabajen juntas para producir un conjunto unificado de funcionalidad. Las aplicaciones probablemente se ejecutan en varios equipos, que pueden representar múltiples plataformas, y pueden estar dispersos geográficamente. Algunas de las aplicaciones se pueden ejecutar fuera de la empresa por los socios de negocios, clientes y otras aplicaciones pueden necesitar ser integrado a pesar de que no fueron diseñados para la integración y no se puede cambiar. Estos temas y otros como ellos son los que hacen difícil la integración de aplicaciones.

La integración de aplicaciones implica un esfuerzo tecnológico, se deben tomar una serie de consideraciones y consecuencias que se deben tener en cuenta para cualquier oportunidad de la integración. Existen diferentes criterios que se deben tener presentes a la hora de integrar aplicaciones, hasta la más pequeña empresa tiene múltiples aplicaciones que necesitan trabajar juntas para proporcionar una experiencia unificada para los empleados de la empresa, socios y clientes. Existen otros criterios a tomar en cuenta para cualquier integración, que se presentan a continuación.

- **Aplicación de acoplamiento:** Incluso las aplicaciones integradas deben minimizar las dependencias entre sí de modo que cada uno pueda evolucionar sin causar problemas a los demás. La interfaz para integrar aplicaciones deben ser lo suficientemente específico para implementar la funcionalidad útil, pero lo suficientemente general como para permitir que la aplicación pueda cambiar cuando sea necesario.

- **Sencillez Integración:** Cuando se integra una aplicación en una empresa, los desarrolladores deben tratar de cambiar la aplicación y reducir al mínimo la cantidad de código necesaria integración.

- **Tecnología de integración:** Diferentes técnicas de integración requieren cantidades variables de software especializado y hardware. Estas herramientas especiales pueden ser costosos, pueden conducir a los proveedores de tecnología, y aumentar la



carga de los desarrolladores a entender cómo usar las herramientas para integrar aplicaciones.

- **Formato de datos:** Aplicaciones integradas deben ponerse de acuerdo sobre el formato de los datos que intercambian, o debe tener un traductor intermedio para unificar las aplicaciones que insisten en formatos de datos diferentes.

- **Puntualidad de los datos:** se debe minimizar el tiempo usado en compartir datos entre una aplicación y otra. Los datos deben ser cambiados con frecuencia en pequeños trozos, sin esperar que se realice el intercambio de un conjunto de datos de mayor tamaño e informar al momento de compartir estos datos.

- **Datos o funcionalidades:** Aplicaciones integradas pueden compartir datos y funcionalidades logrando que cada aplicación pueda invocar la funcionalidad de los demás.

- **Asincronicidad:** en las aplicaciones integradas, donde la aplicación a distancia no se estén ejecutando o la red no esté disponible, la aplicación de origen podría simplemente hacer disponibles los datos compartidos o iniciar una solicitud para una llamada subprocedimiento, pero luego pasan a otro trabajo Confiamos en que la aplicación remota actuará en algún momento posterior.

Tomando en consideración estos tipos de criterios se puede elegir y diseñar un enfoque de integración decidiendo la mejor que se adapte. Hay más de un enfoque para la integración de aplicaciones. Los diferentes enfoques se pueden resumir en cuatro estilos principales de integración:

- **Transferencia de archivos:** las organizaciones para el intercambio de información entre las distintas aplicaciones realizan solicitud para la producción de archivos de datos compartidos para el consumo de otros, y consumen los archivos que otros han producido.

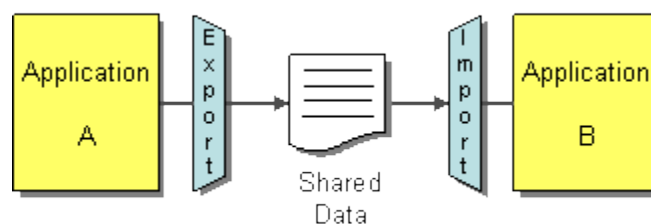
Los archivos son un mecanismo de almacenamiento universal, integrado en cualquier sistema operativo que la empresa disponga, y disponible desde cualquier lenguaje empresarial. El enfoque más simple sería integrar de alguna manera las aplicaciones que utilizan archivos.

La transferencia de archivos tiene ventajas como:

- Los integradores no necesitan conocer internamente la aplicación con las cuales van a trabajar su integración. El equipo de aplicación en sí por lo general proporciona el archivo. El contenido y el formato del archivo se negocian con los integradores. Las aplicaciones están desacopladas una de la otra. Cada aplicación puede hacer cambios internos libremente sin afectar a otras aplicaciones, siempre y cuando produzca los mismos datos en los archivos y el mismo formato.

Y desventajas tales como:

- Se debe tomar la decisión del formato a utilizar en los archivos. Los Integradores deben tomar la responsabilidad de transformar los archivos en diferentes formatos.
- Otro problema con los archivos es cuando para producirlos y consumirlos. Los archivos se producen y se procesan con cierta cantidad de esfuerzo, por lo general no quieren trabajar con ellos con demasiada frecuencia. Se lleva a la decisión de producirlos o ejecutarlos ya sea en el horario nocturno, semanal, trimestral, etc. Las Aplicaciones solo pueden usar los archivos cuando esté disponible.
- Uno de los problemas más obvios de transferencia de archivos son las actualizaciones, tienden a ocurrir con poca frecuencia, dando como resultado la falta de sincronización. El manejo de la información entre las distintas aplicaciones en ocasiones por un retraso en la transferencia de archivos da como resultado del uso de información obsoleta.

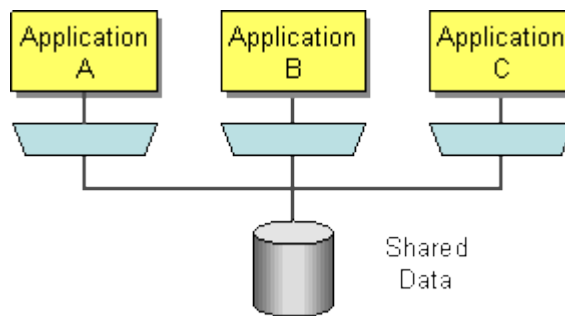


*Figura 14.* Transferencia de Archivos.  
Autor: Hohpe y Woolf (2003 p.66)

- **Base de datos compartida:** Las aplicaciones almacenan los datos que desean compartir en una base de datos común, logrando consistencia en cualquier momento. Considerado esto como una ventaja de usar base de dato compartida, al igual de ser un sistema fácil de manejar por el uso de SQL basados en bases de datos relacionales.

Las desventajas que presenta son:

- Diseño de un esquema unificado adecuado que satisfaga las necesidades de múltiples aplicaciones.
- No trabaja con paquetes externos, la probabilidad de adaptación es limitada.
- Bloqueo de acceso a la base de dato compartida, Múltiples aplicaciones que utilizan una base de datos compartida para leer con frecuencia y modificar los mismos datos pueden causar cuellos de botella e incluso bloqueos, cada aplicación se bloquea cuando otras aplicaciones acceden a la base de datos a nivel local. Una base de datos distribuida con bloqueo los conflictos pueden convertirse fácilmente en una pesadilla rendimiento.



*Figura 15.* Base de Dato Compartida.  
Autor: Hohpe y Woolf (2003 p.69)

- **Invocación de procedimiento remota:** Invocación a procedimiento remoto se aplica el principio de encapsulación para la integración de aplicaciones. Si una aplicación necesita cierta información que es propiedad de otra aplicación, se solicita a la aplicación directamente. Si una aplicación necesita modificar los datos de otra, esta se realiza mediante una llamada a otra aplicación. Cada aplicación puede mante-

ner la integridad de datos de su propiedad. Además, cada aplicación puede alterar sus datos internos sin afectar a otra aplicación.

Invocación a procedimiento remoto permite que las aplicaciones compartan las mismas funciones, pero las mantiene firmemente en el proceso.

Las llamadas a procedimientos remotos se asemejan a las llamadas locales, pero no actúan de la misma manera, las remotos son más lentos y pueden fallar.

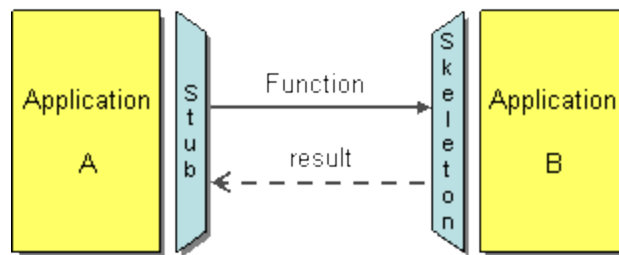


Figura 16. Invocación de procedimiento remoto.  
Autor: Hohpe y Woolf (2003 p.71)

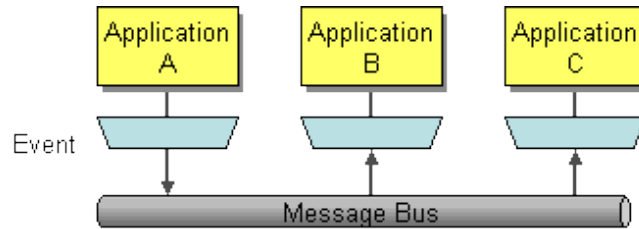
- **Mensajería:** las aplicaciones se conectan a un sistema de mensajería común, el intercambio de datos y la invocación se realiza a través de mensajes. Mensajería hace que al sistema de mensajería responsable de transferir datos de una aplicación a otra, por lo que las aplicaciones pueden centrarse en los datos que necesitan para compartir, pero no se preocupe tanto de cómo compartirla.

La integración de aplicaciones usando mensajes permite transferir paquetes de datos con frecuencia, inmediatamente, confiable y asíncrona, utilizando formatos personalizables.

Al hablar de asíncrona permite que las aplicaciones sean de acoplamiento flexible también hace la comunicación más fiable, porque las dos aplicaciones no tienen que estar en ejecución al mismo tiempo.

Sobre la transformación de Mensajes se puede realizar en tránsito sin que el remitente o el receptor conozcan la transformación. La transformación significa que las aplicaciones pueden tener diferentes modelos conceptuales. La capacidad de transformar mensajes tiene la ventaja de permitir que las aplicaciones sean mucho

más desacopladas entre sí que en la invocación a procedimiento remoto y transferencia de archivos.



*Figura 17. Mensajería.*  
Autor: Hohpe y Woolf (2003 p.73)

En esta investigación se hará uso del enfoque mensajería, tomado en consideración la evaluación y comparación realizada por Hohpe y Woolf (ob.cit) a los enfoques de integración antes mencionados, en el cual concluye que Mensajería es considerada el mejor enfoque para la integración de aplicaciones empresariales. Mensajería proporciona mayor disponibilidad de la información, mayor nivel de seguridad y mayor encapsulamiento por tal razón se considera Mensajería como la alternativa más idónea para usar en esta investigación.

### ***Patrones de Integración Empresarial***

Los patrones de integración empresarial, son una evolución de los patrones de diseño de software tradicionales, a un contexto específico, como lo es la integración de aplicaciones empresariales. Estos patrones representan las decisiones que deberán tomarse al momento de integrar dos o más aplicaciones en un contexto empresarial y las consideraciones que implica tomar esas decisiones.

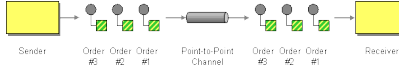
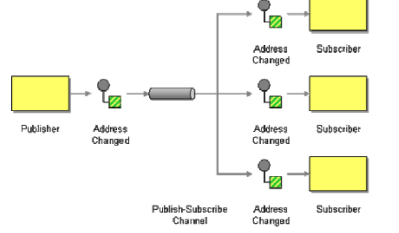
Según Hohpe y Woolf (2003) los patrones describen soluciones a problemas recurrentes, son un conjunto de reglas, principios y metodologías que seguidas, ayudan a que la integración entre distintas aplicaciones, plataformas y entre distintos servicios de una empresa. El objetivo principal de estos patrones de integración es dar una forma no restrictiva de compartir datos entre procesos de negocio, entendiendo

como proceso de negocio, cualquier proceso, aplicación o repositorio de datos de una empresa. El uso adecuado de los patrones ayuda a llenar la brecha entre la visión de alto nivel de integración y la implementación real del sistema.

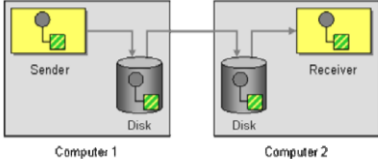
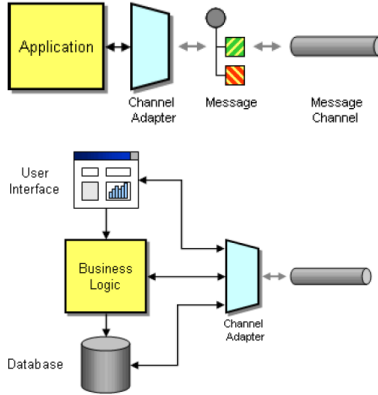
Existen diferentes tipos de patrones, de los cuales solo se detallará los referentes al tipo de estilo a utilizar en la presente investigación, los patrones de mensajería. A continuación se describen:

Los patrones de integración empresarial han sido agrupados en categorías, que reflejan el alcance y la abstracción de los patrones. Resolviendo de esta manera una situación específica dentro del proceso de integración.


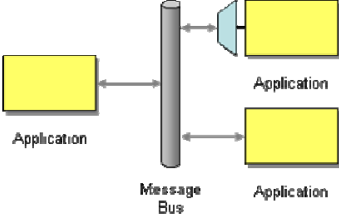
- **Patrones de canal del mensaje:** Estos patrones describen los aspectos ha tomar en cuenta a la hora de escoger el canal de comunicación entre las aplicaciones, cuando se desea integrar dos o más sistemas de software. Por lo tanto estos patrones resuelven distintos problemas de transporte de los mensajes entre aplicaciones. Los patrones que integran esta categoría se muestran en la siguiente tabla:

PATRONES DE CANAL DEL MENSAJE	DESCRIPCION	DISEÑO
<p><b>Point-to-Point Channel:</b></p>	<p>Envía un mensaje a solo un receptor disponible.</p>	
<p><b>Publish-Subscribe Channel:</b></p>	<p>Cuenta con un canal de entrada que se divide en múltiples canales de salida, uno para cada suscriptor, proporcionando una copia del mensaje a cada uno de los canales de salida. Cada canal de salida tiene solamente un suscriptor, que sólo se le permite consumir un mensaje a la vez y</p>	

	<p>una vez consumida las copias desaparecer de su canales.</p>	
<p><b>Datatype Channel.</b></p>	<p>Es la estructura y el formato del contenido del mensaje. La cabecera del mensaje contiene el tipo de datos que se envía. Un canal es como una colección y un receptor es como un repetidor. Se utiliza este tipo de canal de datos por separado para cada tipo de datos, de modo que todos los datos de un canal en particular sean del mismo tipo.</p>	
<p><b>Invalid Message Channel.</b></p>	<p>Mensaje no válido Cuando el receptor detecta que el mensaje que está tratando de procesar no es válido. El receptor mueve el mensaje incorrecto a un canal de mensajes no válido, que es un canal especial utilizado para todos los mensajes que no pueden ser procesados por sus receptores.</p>	
<p><b>Canal Dead Letter</b></p>	<p>Cuando un mensaje tiene algo malo en su encabezado que le impide ser entregado el sistema de mensajería elige mover el mensaje a un canal de letra</p>	

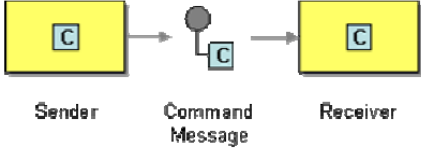
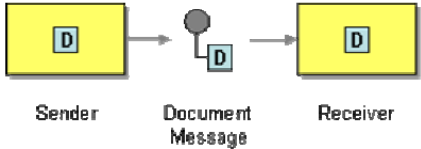
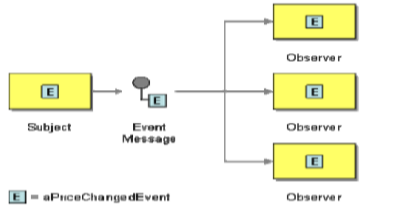
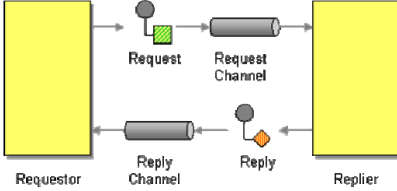
	<p>Muerto o también llamado cola de mensajes inactivos y no se puede entregar con éxito.</p>	
<p><b>Entrega garantizada</b></p>	<p>Con la entrega garantizada, el sistema de mensajería utiliza un almacén de datos integrada. Donde los mensajes se pueden almacenar localmente. Cuando el remitente envía un mensaje, la operación de envío no se completa éxito hasta que el mensaje se almacena de forma segura en el almacén de datos del remitente y no se elimina hasta que se entregue satisfactoriamente y sea reconocido por el receptor.</p>	
<p><b>adaptador de canal</b></p>	<p>El adaptador actúa como un cliente de mensajería para el sistema de mensajería e invoca funciones a través de una interfaz de aplicación suministrada. De esta manera, cualquier aplicación puede conectarse al sistema de mensajería y ser integrado con otras aplicaciones, siempre y cuando se tiene un canal adaptador adecuado.</p>	 <p><i>A Channel Adapter Connecting to Different Layers of an Application</i></p>

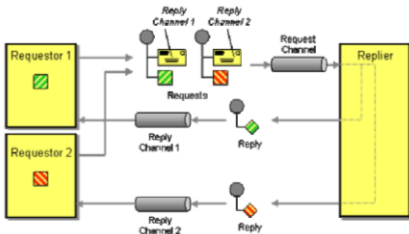
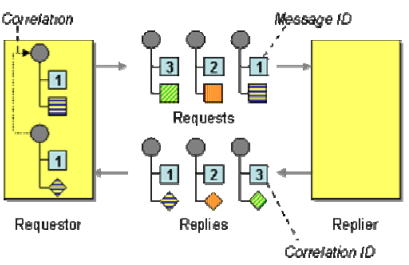
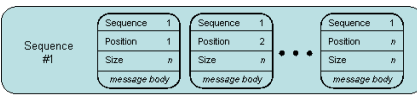


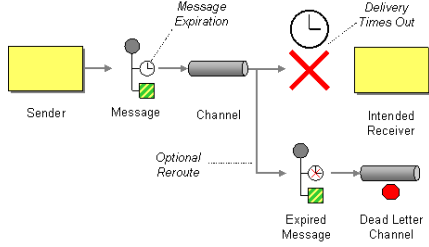
<p><b>Puente Mensajería</b></p>	<p>Es un conjunto de adaptadores de canal, donde el cliente puede ser otro sistema de mensajería, y donde cada par de adaptadores conecta un par de canales correspondientes. El puente actúa como un mapa de un conjunto de canales a la otra, y transforma el mensaje de un formato a otro.</p>	
<p><b>Mensaje Bus</b></p>	<p>Un Mensaje de bus es una combinación de un modelo común de datos, un conjunto de comandos comunes, y una infraestructura de mensajería para permitir diferentes sistemas para comunicarse a través de un conjunto compartido de interfaces.</p>	

*Cuadro N°2. Patrones de canal del mensaje  
Fuente: Hohpe y Woolf (2003) p.108-139.*

- Patrones de construcción del mensaje:** Estos patrones, contemplan los aspectos que se encargan de envolver los datos de interés en un mensaje, para que pueda ser transmitido a través de los canales, que llevaran la información hacia los interesados en el mensaje. De esta manera abordan el diseño de los mensajes que envían los diferentes participantes de una comunicación. Los patrones que integran esta categoría son:

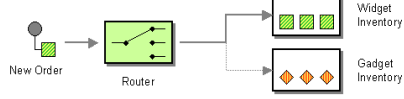
PATRONES DE CONSTRUCCIÓN DEL MENSAJE	DESCRIPCION	DISEÑO
<b>Mensaje de comando</b>	Un mensaje de comando es un mensaje normal que pasa a contener un comando. Se envían generalmente en un canal punto a punto, se consume y se invoca una vez. Utilizado para invocar un procedimiento en otra aplicación.	 <p>Sender      Command Message      Receiver</p> <p><b>C</b> = getLastTradePrice("DIS"),</p>
<b>Mensaje documento</b>	Utilizar un mensaje de documento para transferir de forma fiable una estructura de datos entre aplicaciones.	 <p>Sender      Document Message      Receiver</p> <p><b>D</b> = aPurchaseOrder</p>
<b>Mensaje Evento</b>	Para enviar un evento se transforma como un mensaje y se envía por un canal.	 <p>Subject      Event Message      Observer Observer Observer</p> <p><b>E</b> = aPriceChangedEvent</p>
<b>Solicitud-respuesta</b>	Un mensaje es enviado primero, luego recibido, de tal manera que el remitente y el receptor no pueden tener acceso al mensaje al mismo tiempo.  Enviar un par de mensajes de solicitud-respuesta, cada	 <p>Requestor      Request      Request Channel      Replier Reply Channel      Reply</p>

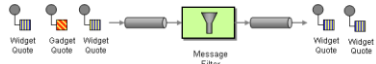
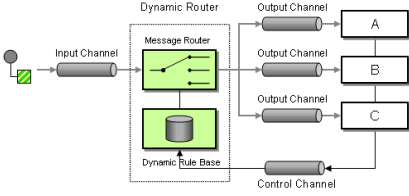
	<p>uno en su propio canal.</p> <p>Solicitud-respuesta tiene dos participantes:</p> <ol style="list-style-type: none"> <li>1.Solicitante: Permite enviar un mensaje de solicitud y espera un mensaje de respuesta.</li> <li>2. Receptor: Recibe el mensaje de solicitud y responde con un mensaje de respuesta.</li> </ol>	
<p><b>dirección del remitente</b></p>	<p>El mensaje de solicitud debe contener una dirección de retorno que indica dónde enviar la respuesta</p>	
<p><b>Identificador de correlación</b></p>	<p>Cada mensaje de respuesta debe contener un identificador de correlación, un identificador único que indica para quien va dirigido el mensaje de respuesta.</p>	
<p><b>Secuencia de Mensaje</b></p>	<p>Siempre es necesario enviar un conjunto de datos como una secuencia de mensajes y marcar cada mensaje con los campos de identificación de secuencia.</p>	

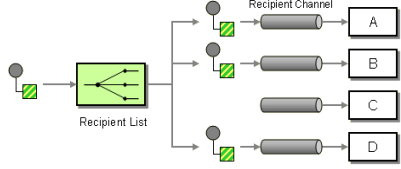
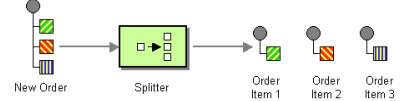
<p><b>Caducidad o del mensaje</b></p>	<p>Representa un límite de tiempo de cuánto el mensaje es viable. Dependiendo de este tiempo el mensaje se considera caducado o no y dependiendo además de si ha sido consumido.</p>	
<p><b>Indicador de Formato</b></p>	<p>Diseño de un formato de datos que incluye un Indicador de formato, de modo que el mensaje indica qué formato se está utilizando. El indicador de formato permite al remitente indicarle al receptor el formato del mensaje.</p>	

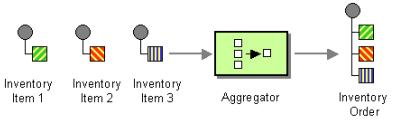
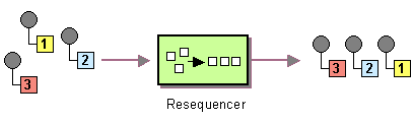
Cuadro N°3. Patrones de construcción del mensaje  
Fuente: Hohpe y Woolf (2003) p.140-168.

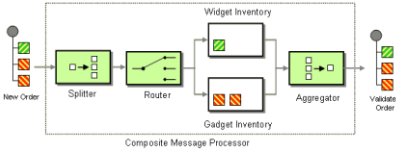
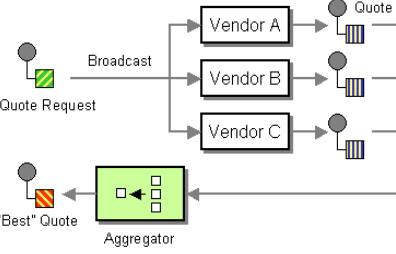
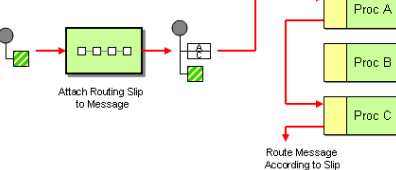
- Patrones de enrutamiento del mensaje:** Describen las distintas soluciones para proveer enrutamiento e intermediación para una solución de integración. Dichos patrones están relacionados con el ruteo de los mensajes desde una aplicación a otra. Los patrones que componen esta categoría son:

PATRONES DE ENRUTAMIENTO DEL MENSAJE	DESCRIPCION	IMAGEN
<p><b>Router Basada en contenido</b></p>	<p>El router basada en contenido examina el contenido del mensaje y</p>	

	<p>encamina el mensaje en un canal diferente, basado en los datos contenidos en el mensaje.</p>	
<p><b>Filtro de mensajes</b></p>	<p>Un filtro de mensajes, eliminar mensajes no deseados de un canal basado en un conjunto de criterios. El mensaje de filtro sólo tiene un único canal de salida. Si el contenido del mensaje coincide con los criterios especificados por el filtro de mensajes, el mensaje se en-ruta al canal de salida. Si el contenido del mensaje no coincide con los criterios, se descarta el mensaje.</p>	 <p>The diagram shows a Message Filter component, represented by a green box with a yellow 'Y' shape. It receives input from three sources: 'Widget Quote', 'Gadget Quote', and another 'Widget Quote'. The output of the filter goes to a single 'Widget Quote' destination.</p>
<p><b>Router Dinámico</b></p>	<p>Utilice un router dinámico, un router que puede auto-configurar basado en mensajes especiales de configuración de destinos de participantes. El router dinámico almacena las "preferencias" de cada participante en una base de reglas. Cuando llega un</p>	 <p>The diagram illustrates a Dynamic Router system. It features an 'Input Channel' on the left that feeds into a 'Message Router' (a green box with a switch). Below the Message Router is a 'Dynamic Rule Base' (a green box with a database icon). The Message Router is connected to three 'Output Channels' that lead to destinations labeled 'A', 'B', and 'C'. A 'Control Channel' is also shown at the bottom, connected to the Dynamic Rule Base.</p>

	<p>mensaje, el router dinámico evalúa todas las reglas y encamina el mensaje al destinatario cuyas reglas se cumplan. Esto permite un enrutamiento eficaz, predecible y sin la dependencia de mantenimiento del router dinámico en cada posible receptor.</p>	
<p><b>Lista de destinatarios</b></p>	<p>Utilice una lista de destinatarios para inspeccionar un mensaje entrante, determinar la lista de los destinatarios que desees, y reenviar el mensaje a todos los canales asociados a los destinatarios de la lista. La lista de destinatarios no suelen modificar el contenido del mensaje.</p>	
<p><b>Splitter o DIVISOR</b></p>	<p>Utilizar un divisor o splitter para dividir el mensaje compuesto en una serie de mensajes individuales, cada uno que</p>	

	contiene datos relacionados con un artículo.	
<b>Agregador</b>	El agregador es un filtro especial que recibe un flujo de mensajes e identifica los mensajes que están correlacionados. Una vez que un conjunto completo de los mensajes se ha recibido, el agregador recoge información de cada mensaje correlacionada y publica un único mensaje, agregada al canal de salida para su posterior procesamiento.	
<b>Resecunciador</b>	El Resecunciador recibe un flujo de mensajes que pueden no llegar a fin. El Resecunciador contiene en tampón interna para almacenar mensajes de fuera de la secuencia hasta que se obtiene una secuencia completa. Los mensajes en secuencia a continuación, se publican en el canal de salida. Como la mayoría de los routers,	

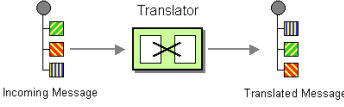
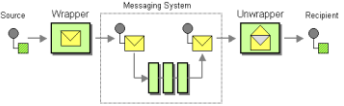
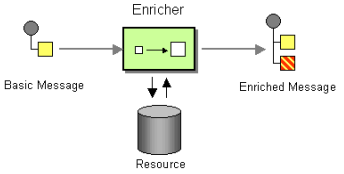
	<p>un Resecunciador no suele modificar el contenido del mensaje.</p>	
<p><b>Procesador de mensajes compuestos</b></p>	<p>El procesador de mensajes Compuesto divide el mensaje, enruta los sub-mensajes a los destinatarios apropiados y re-agrega las respuestas de nuevo en un solo mensaje.</p>	
<p><b>Scatter-Gather</b></p>	<p>Un Scatter-Gather transmite un mensaje a varios destinatarios y re-agrega las respuestas de nuevo en un solo mensaje. El Scatter-Gather enrutas un mensaje de solicitud a un número de destinatarios. A continuación, utiliza un agregador para recoger las respuestas y se destila en un único mensaje de respuesta.</p>	
<p><b>Lista de distribución</b></p>	<p>Cada mensaje se le adjunta una lista de distribución, la misma especifica la secuencia de pasos de procesamiento. Se Envuelve cada componente con un</p>	



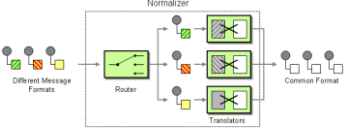
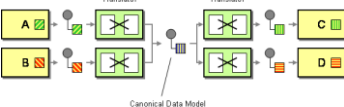
	router de mensaje especial que lee la lista de distribución y enruta el mensaje al siguiente componente de la lista.	
<b>Administrador de procesos</b>	Una unidad de procesamiento central, un administrador de procesos, permite mantener el estado de secuencia y determinar el siguiente paso de procesamiento sobre la base de los resultados intermedios.	<p>The diagram shows a central box labeled 'Process Manager' containing a tree-like structure. A 'Trigger Message' (represented by a green square with a diagonal line) enters from the left. Three arrows, labeled 1, 2, and 3, point from the Process Manager to three separate boxes labeled 'Proc A', 'Proc B', and 'Proc C' respectively.</p>
<b>Message Broker</b>	Message Broker recibe mensajes de varios destinos, determina el destino y la ruta del mensaje en el canal correcto.	<p>The diagram shows a central green box representing the Message Broker. It is connected to five yellow boxes arranged around it: one above, one below, one to the left, one to the right, and one to the right of the bottom one. Lines connect the central box to each of these surrounding boxes, indicating bidirectional communication.</p>

Cuadro N°4. Patrones de enrutamiento del mensaje  
Fuente: Hohpe y Woolf (2003) p.208-290.

- Patrones de transformación del mensaje:** Identifican y dan solución a las diferentes situaciones que se presentan cuando existe la necesidad de operar entre sistemas que usan diferentes formatos, por lo tanto estos patrones permiten definir el manejo de transformaciones que pueden realizarse sobre los mensajes que se intercambian, en lo que refiere a los diferentes formatos requeridos por cada aplicación. Los patrones que integran esta categoría son:

PATRONES DE TRANSFORMACION DEL MENSAJE	DESCRIPCION	IMAGEN
<b>Traductor de mensaje</b>	Traductor de mensaje es un filtro especial utilizado para convertir de un formato de datos a otro.	
<b>Sobre Envoltura</b>	Utilice un Envelope Envoltura para envolver los datos de aplicación dentro de un sobre que es compatible con la infraestructura de mensajería. Desenvolver el mensaje cuando llega a destino.	
<b>Enriquecedor contenido</b>	Un Enriquecedor de contenido, es utilizado para acceder a una fuente de datos externa con el fin de aumentar un mensaje con la información que falta. El Enriquecedor de contenido utiliza la información en el interior del mensaje entrante para recuperar datos de una fuente externa. Después el Enriquecedor de contenido recupera los datos necesarios de los recursos	

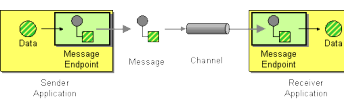
	que añaden los datos para el mensaje.	
<b>Filtro de contenido</b>	Filtro de contenido es usado para eliminar elementos de datos sin importancia de un mensaje dejando sólo los elementos importantes. El filtro de contenido también es útil para simplificar la estructura del mensaje.	
<b>Reclamación Check</b>	Almacenar datos de mensajes en un almacén persistente y pasar una verificación a los componentes siguientes. Estos componentes pueden usar la Reclamación para descubrir para recuperar la información almacenada. Reduciendo el volumen de datos de los mensajes enviados a través del sistema sin sacrificar el contenido de la información	
<b>Normalizador</b>	Un Normalizador es usado para enrutar cada tipo	

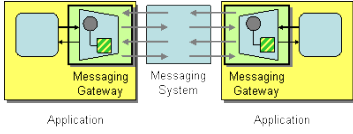
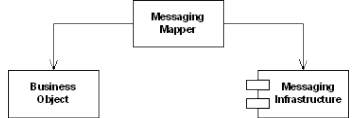
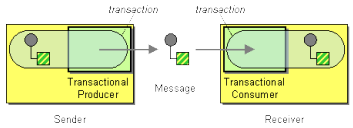
	de mensaje a través de un traductor de mensaje para que los mensajes resultantes coinciden con un formato común.	
<b>Canónica Modelo de Datos</b>	El modelo de datos de Canonical proporciona un nivel adicional de direccionamiento indirecto entre formatos de datos individuales de las aplicaciones.	

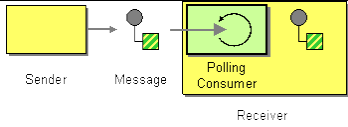
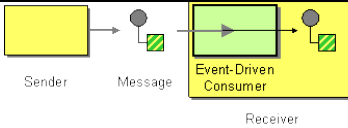
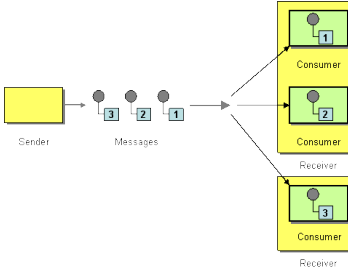
Cuadro N°5. Patrones de transformación del mensaje

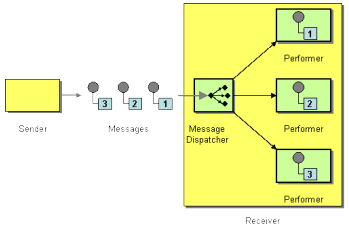
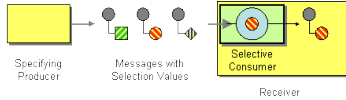
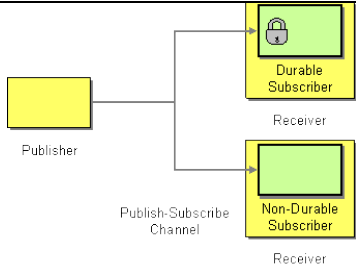
Fuente: Hohpe y Woolf (2003) p.291-316.

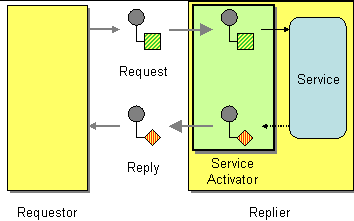
- **Patrones de puntos de acceso (EndPoint):** Describen cómo las aplicaciones pueden conectarse al sistema de mensajería, para que puedan enviar y recibir mensajes. Es decir estos patrones entregan pautas relacionadas a la generación y el consumo de mensajes, especificando por ejemplo como un productor puede interactuar con el canal para producir un mensaje o como un receptor puede comportarse ante la ocurrencia de un nuevo mensaje. Los patrones que integran la categoría son:

PATRONES DE PUNTOS DE ACCESO	DESCRIPCION	DISEÑO
<b>Message Endpoint</b>	Un mensaje de punto final es un cliente de mensajería, usado para conectar una aplicación a un canal de mensajería, la aplicación puede utilizarlo	

	para enviar o recibir mensajes	
<b>Messaging Gateway</b>	El Gateway de mensajería encapsula código de mensajería específico y lo separa del resto del código de la aplicación.	 <p>The diagram shows two 'Application' boxes on either side of a central 'Messaging System' box. Each application is connected to a 'Messaging Gateway' box, which in turn connects to the 'Messaging System'. Arrows indicate bidirectional communication between the applications and their respective gateways, and between the gateways and the messaging system.</p>
<b>Messaging Mapper</b>	Mapper mensajería contiene la lógica correspondencia entre la infraestructura de mensajería y los objetos de dominio. La Mensajería Mapper accede a uno o más objetos de dominio y las convierte en un mensaje según se requiera por el canal de mensajería. También realiza la función opuesta, crear o actualizar objetos de dominio basados en los mensajes entrantes.	 <p>The diagram shows a central 'Messaging Mapper' box. Two arrows point from it to a 'Business Object' box on the left and a 'Messaging Infrastructure' box on the right, indicating the mapping logic between the two.</p>
<b>Transactional Client.</b>	Utilizada para que un cliente transaccional puede especificar los límites de transacción. Tanto un emisor y un receptor pueden ser transaccional.	 <p>The diagram shows a 'Transactional Producer' (labeled 'Sender') on the left and a 'Transactional Consumer' (labeled 'Receiver') on the right. A 'Message' is sent from the producer to the consumer. Above the message, two 'transaction' boxes are shown, one around the producer and one around the consumer, indicating that both are part of a transactional process.</p>
<b>Polling Consumer.</b>	La aplicación debe utilizar	

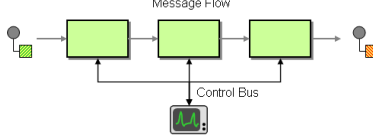
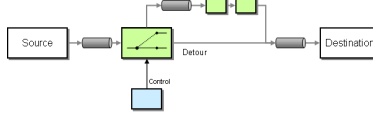
	<p>un sondeo del Consumidor, que hace explícitamente una llamada cuando se quiere recibir un mensaje.</p>	
<p><b>Event-Driven Consumer</b></p>	<p>Lo llamamos un consumidor Event-Driven porque los actos de recepción como la entrega de mensajes es un evento que activa el receptor en acción.</p>	
<p><b>Competing Consumers</b></p>	<p>Crear varios consumidores que compiten en un solo canal, de manera que los consumidores pueden procesar varios mensajes a la vez. Los consumidores en competencia son múltiples, Cuando el canal entrega un mensaje, cualquiera de los consumidores potencialmente podría recibirlo. La implementación del sistema de mensajería determina que los consumidores realmente reciben el mensaje, pero en efecto los consumidores compiten entre sí para ser el receptor.</p>	

<p><b>Message Dispatcher</b></p>	<p>Crear un mensaje de Dispatcher en un canal que va a consumir mensajes de un canal y distribuirlos.</p>	
<p><b>Selective Consumer</b></p>	<p>Consumidor un Selectivo al Consumo, que filtra los mensajes entregados por su canal de modo que sólo recibe los que coinciden con sus criterios.</p>	
<p><b>Durable Subscriber</b></p>	<p>Una suscripción duradera guarda los mensajes para un abonado inactivo y delibera estos mensajes guardados cuando el abonado se vuelve a conectar</p>	
<p><b>Idempotent Receiver</b></p>	<p>Esto significa que un mensaje con seguridad pueden reenviarse sin causar ningún problema incluso si el receptor recibe duplicados del mismo mensaje.</p>	
<p><b>Service Activator</b></p>	<p>El activador se encarga de todos los detalles de mensajería e invoca el servicio como cualquier otro</p>	

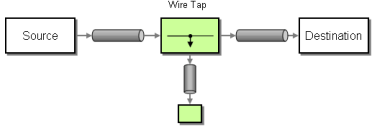
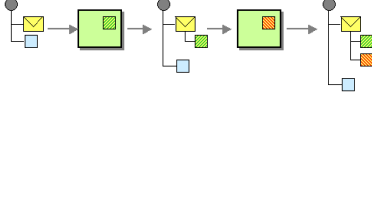
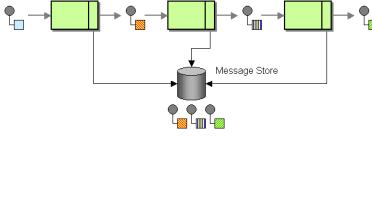
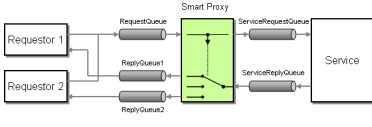
	<p>cliente, de manera que el servicio ni siquiera sabe que está siendo invocado a través de mensajería. Puede invocar siempre el mismo servicio, o puede utilizar la reflexión para invocar el servicio indicado por el mensaje.</p>	
--	--	---

Cuadro N°6. Patrones de puntos de acceso (EndPoint)  
Fuente: Hohpe y Woolf (2003) p.415-476.

- **Patrones de gestión del sistema:** Indican cómo atacar las necesidades de administración, monitoreo y testeo de los componentes del sistema y de los canales de comunicación. Los patrones que integran la categoría son:

PATRONES DE GESTIÓN DEL SISTEMA	DESCRIPCION	IMAGEN
<p><b>Bus de Control</b></p>	<p>Utilice un bus de control para gestionar un sistema de integración. El bus de control utiliza canales separados para transmitir datos que son relevantes para la gestión de los componentes que intervienen en el flujo de mensajes.</p>	
<p><b>Detour</b></p>	<p>El Detour utiliza un router basado en el contexto simple con dos canales de</p>	



	<p>salida. Detour enruta los mensajes a un canal diferente. Este canal envía el mensaje a los componentes adicionales que pueden inspeccionar y / o modificar el mensaje.</p>	
<p><b>Wire Tap</b></p>	<p>Es una Lista de destinatarios fija con dos canales de salida. Consume mensajes fuera del canal de entrada y publica el mensaje sin modificar a los dos canales de salida.</p>	
<p><b>Message History</b></p>	<p>Un historial de mensajes es una lista de todas las aplicaciones que el mensaje pasa a través desde su inicio.</p>	
<p><b>Message Store</b></p>	<p>El almacén de mensajes es un lugar donde se guardan un duplicado del mensaje enviado al canal.</p>	
<p><b>Smart Proxy</b></p>	<p>Un Proxy inteligente es usado para almacenar la dirección de retorno suministrada por el solicitante original y sustituirlo por la dirección del Proxy inteligente.</p>	

<p><b>Test Message</b></p>	<p>La prueba de mensajes es usado para asegurar el buen estado de los componentes de procesamiento de mensajes.</p>	
<p><b>Channel Purger</b></p>	<p>Usado para eliminar los mensajes no deseados de un canal</p>	

Cuadro N°7. Patrones de gestión del sistema  
Fuente: Hohpe y Woolf (2003) p.476-503.

Estos patrones describen formas para hacer de la integración una tarea más sencilla, tanto en aplicaciones externas como con procesos de la misma aplicación. Los patrones englobados en el ámbito de los sistemas de comunicación basados en mensajes están orientados a permitir la distribución de mensajes de manera sencilla, flexible, optimizada y normalizada.

**Arquitectura Orientada a Servicios (SOA)**

Alba (2008). Define a SOA como una Arquitectura Orientada a Servicios para conectar sistemas entre si permitiendo aplicar lógicas de control, negocio y procesos. Permite simplificar las relaciones entre distintos sistemas, optimizar su funcionamiento, facilitar la incorporación de nuevos elementos e incluso cambiar los existentes de una manera sencilla. En resumen, se tiende a una arquitectura de conexión de sistemas mucho más ordenada, robusta y escalable.

Existen organizaciones que disponen de gran cantidad de información, que está distribuida entre muchos sistemas. La arquitectura SOA sirve para definir cómo se deben unir los distintos sistemas para conseguir que todo sea más eficiente, escalable, operable, flexible y además se puedan realizar operaciones con los datos.

## ***Servicio Web***

El consorcio W3C define los Servicios como un conjunto de aplicaciones o de tecnologías con capacidad para añadirse a múltiples plataformas e intercambiar datos para ofrecer unos servicios usando procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web

Los servicios web proporcionan mecanismos de comunicación estándares entre aplicaciones que interactúan entre sí para presentar información dinámica al usuario.

Los servicios web son aplicaciones modulares auto descriptivas que se pueden publicar, ubicar e invocar desde cualquier punto de la web o desde una red local. Romero (2011).

Valera (2011). Define los servicios web como una herramienta que ayuda en la forma en que se comunican las aplicaciones sin importar la plataforma en la que se encuentren. Lo que hace que se estén extendiendo con gran rapidez y se estén implementando en muchas empresas.

Ramos (2014) los servicios web son independientes de la plataforma y el lenguaje, es decir, se pueden desarrollar en cualquier lenguaje e implementar en cualquier plataforma. Se definen a través de las siguientes especificaciones o protocolos:

- **XML (Extensible Markup Language):** Es el lenguaje de marcas que se utiliza para describir la información, puede describir datos y documentos.

- **SOAP (Protocolo Simple de Acceso a Objetos):** Es un protocolo basado en XML, que permite la interacción entre varios dispositivos y tiene la capacidad de transmitir información compleja. Es el protocolo que indica cómo se deben codificar los mensajes que circularán entre las dos aplicaciones, cliente y proveedor del servicio. La estructura de un mensaje SOAP está formada por un header o cabecera y un body o cuerpo. W3C.

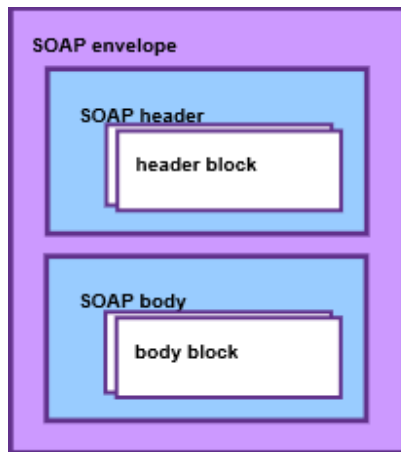


Figura 18. Componentes de un Mensaje SOAP.  
 Autor: Papazoglou, M (2008 p. 10)

**WSDL (Lenguaje de Descripción de Servicios Web):** Es un lenguaje común y extensible para describir los servicios, basado en XML. La función de WSDL, es acordar con una aplicación que formato usar para comunicarse, especificando por medio de un lenguaje estándar, tanto la dirección del servicio como la interfaz que se va a utilizar. En la figura no. 19 se muestra los niveles usados en WSDL. Jiménez (2011) define los niveles como uno abstracto y uno concreto. En el nivel abstracto, WSDL define un Servicio Web en términos del mensaje que puede enviar y recibir, normalmente utilizando un formato de datos

1. Definition: Contiene el nombre del servicio web.
2. Types: Utiliza el lenguaje XML para declarar tipos de datos complejos y otros elementos que forman parte del documento WSDL.
3. PortType: Describe la interfaz del servicio web y sus métodos, básicamente agrupan las operaciones sin ningún compromiso con un protocolo de transporte.

En el nivel concreto, WSDL define un

4. Binding: Asignar un portType y sus elementos de funcionamiento y estilo de codificación a un protocolo determinado por ejemplo a SOAP.

5. Services: Responsable de asignar una dirección de internet a un Binding específico.

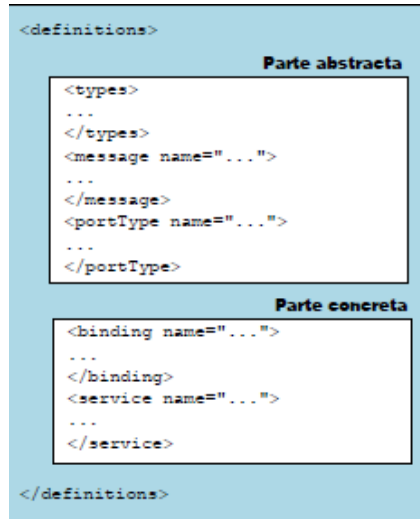


Figura 19. Estructura de un documento WSDL.  
Autor: Jiménez C. (2011 p. 190)

- **UDDI (Universal Description, Discovery and Integration):** Este protocolo proporciona un mecanismo estándar para registrar y localizar los servicios web que se pueden ofrecer a los clientes. Los directorios UDDI actúan como una guía telefónica de servicios web.

De acuerdo a las definiciones anteriormente señaladas, se puede inferir que los servicios web son el medio de comunicación que permite la interoperabilidad entre aplicaciones a través del uso de estándares abiertos. De esta manera, aplicaciones desarrolladas en diferentes lenguajes de programación y ejecutadas en diferentes plataformas pueden intercambiar datos y presentar información dinámica al usuario.

Los servicios web vienen a dar solución al problema de integración sin importar la plataformas, sin considerar la unificación de las mismas; más bien implantar interfaces que hagan posible la integración, pero conservando las peculiaridades.

### ***Bus de Integración de Servicios (ESB)***

La definición de un Bus de Servicio Empresarial o ESB de acuerdo a Caponi, et. al. (2008) (Citado por Valera 2011) es la siguiente:

Es una plataforma de integración de aplicaciones basada en estándares, que combina entre otras cosas: mensajería, servicios, ruteo y transformación de mensajes, con el objetivo de coordinar y conectar de manera confiable un número significativo de aplicaciones. Está construido sobre un canal común de mensajes (bus) altamente distribuible, multiprotocolo, basado en estándares y provee la columna vertebral para implementar una Arquitectura Orientada a Servicios (SOA). (p. 28).

Laliwala (2013) define ESB como una aplicación que permite el acceso a otras aplicaciones y servicios. Tiene como tarea principal la mensajería y la integración de la organización. Un ESB es un sistema Middleware distribuido para integrar diferentes aplicaciones. Todas estas aplicaciones se comunican a través del ESB que consta de un conjunto de servicios que integran diversos tipos de aplicaciones

Cada aplicación puede comunicarse a través de un ESB para:

- Integrar dos o tres aplicaciones y/o servicios
- Integrar mas aplicaciones y/o servicios en el futuro
- Usar diferentes protocolos de comunicación
- Publicar servicios para la composición y el consumo
- Transformación y enrutamiento de mensajes.

La necesidad de un ESB surge de la complejidad de las organizaciones que deben coordinar e integrar sus procesos de negocio, sistemas operacionales y datos sin renunciar a la innovación tecnológica imprescindible para ser competitivos. Un ESB es la implementación de SOA, una arquitectura que permite mantener integrados los sistemas, nuevos y legados, en un estilo completamente distribuido e interoperable.

## *Plataforma de Integración*

La integración de aplicaciones es de interés a las organizaciones que requieren integrar sus aplicaciones y sus procesos de negocios, lo que hace imprescindible para ser competitivos. Las plataformas de integración surgen de la necesidad de interrelacionar datos de distintas aplicaciones existentes dentro de la organización, el Éxito radica en definir una plataforma y una arquitectura de integración que sea ágil, fiable y flexible a la hora de realizar cualquier cambio o incluir nuevas aplicaciones.

Existen diferentes plataformas que dan soporte al problema de integración de aplicaciones: tales como: Oracle ESB, IBM WebSphere, así como algunas alternativas Open Source como Mule, Apache ServiceMix, WSO2. En la presente investigación se describirá Mule, por ser una plataforma de software libre y por el manejo de mensajes para su interacción.

Mule es una plataforma Software Libre que permite a los desarrolladores adaptarlo a sus necesidades, cuya arquitectura se basa en el concepto de Enterprise Service Bus (ESB), un ESB diseñado para soportar transacciones de alto rendimiento y multi-protocolo. Puede ser empleado para mensajería entre sistemas de información, como middleware transaccional, o como parte de un servidor de aplicaciones.

Mule, se diferencia de otros productos ya que permite a los desarrolladores, integrar diferentes aplicaciones de un modo incremental sin afectar la infraestructura existente en la empresa.

La arquitectura de Mule, se centra en cuatro tipos de componentes, descritos a continuación:

- **Componentes:** es una unidad de programación que ejecuta o que contiene la lógica del negocio. Por ejemplo, cualquier objeto, POJO o javabeans, servicio REST, EJBs, Bean de Spring, etc.
- **Routers:** son los encargados de asegurarse que el componente de servicio reciba el mensaje adecuado. Existen dos tipos de Routers:
  - **Inbound routers:** Se encarga de controlar el mensaje recibido para direccionarlo a un servicio.

- **Outbound routers:** Se encarga de redirigir la salida hacia otro servicio. Se puede utilizar como balanceador de carga o enrutamiento basado en reglas.
- **Endpoints:** Utilizados para conectar componentes con otros sistemas externos, es decir es una entidad de configuración que especifica cómo y dónde un mensaje puede ser enrutado. Se configura en un router de entrada (**Inbound routers**) o de salida (**Outbound routers**) y especifica donde el mensaje debe ser enviado o desde donde debe ser recibido usando un transporte.
- **Filters:** son los usados para filtrar opcionalmente los mensajes entrantes o de salida que interactúan con un conector.
- **Transformadores:** Son la clave para el intercambio de datos en Mule ESB ya que se encargan de transformar los datos de formato para redirigirlo a la entrada de un componente, es decir convierten los datos a un formato que otro componente o aplicación pueda procesar.
- **Transporte:** Se encargan de manejar la forma de conectarnos entre diferentes tecnologías utilizando diferentes protocolos de transporte: HTTP, JMS, FTP, etc.

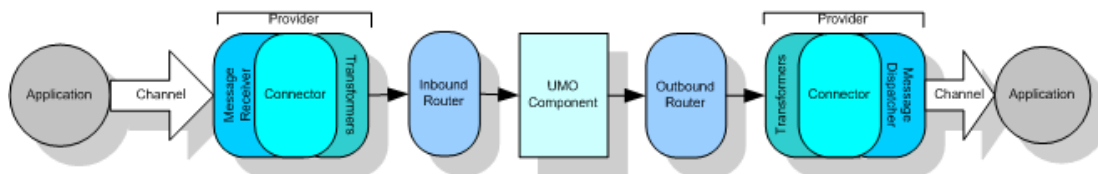


Figura 20. Arquitectura Mule.

Fuente: <http://www.mulesoft.org/documentation-3.2/display/MULEUSER/Architecture+Guide>

Los conceptos presentados anteriormente serán manejados durante la investigación y serán de uso para el lector para tener una mejor comprensión de la investigación realizada.



## **CAPITULO III**

### **MARCO METODOLÓGICO**

A lo largo de este capítulo se pretende especificar la metodología de investigación que será empleada para el desarrollo de este trabajo, y cuyo fin es responder satisfactoriamente a los objetivos planteados en el Capítulo I. En el mismo se desarrollaran los aspectos metodológicos en los cuales estará enmarcada la investigación, para lo que se requiere hacer un estudio de los temas necesarios para lograr una solución que se adapte a los objetivos que deseamos alcanzar. De esta forma dar respuesta no solo a nuestro caso de estudio sino a organizaciones que desean obtener solución al problema de integración de aplicaciones.

En el desarrollo del marco metodológico se describirán los métodos, técnicas e instrumentos utilizados para generar el conjunto de datos que se analizaran como el resultado final de la investigación.

#### **Naturaleza de la Investigación**

Por la naturaleza de esta investigación, la cual busca una solución a los problemas de integración de aplicaciones, se presenta bajo la modalidad de proyectos tal como lo especifica el manual UCLA (2002), ya que cuando se desea realizar una mejora se buscan valores reales que lleven a la realización de una propuesta, en la representa de un proyecto factible para la solución de un problema teórico-práctico, además esta se apoya en una investigación de campo descriptiva, debido a que esta no solo permite observar sino también recolectar los datos directamente de la realidad

objeto de estudio, para luego analizar los resultados de estas indagaciones.

En este sentido, según el Manual de trabajo de Grado de especialización, Maestría y tesis doctorales de la Universidad Experimental Libertador (UPEL, 2006), dispone que:

El Proyecto Factible consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El Proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades. (p.13).

Según lo planteado por el manual de la UPEL la presente investigación es un proyecto factible por tratarse de una propuesta.

Será apoyado a una investigación de campo, tipo descriptivo y documental. Que según el manual de la UPEL (2006) destaca que:

La investigación de campo el análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos, o predecir su ocurrencia, haciendo uso de métodos característicos de cualquiera de los paradigmas o enfoques de investigación conocidos o en desarrollo. (p.11).

Como se dijo anteriormente la investigación se ubica en una investigación de campo. Es de campo porque siguen las estrategias basadas en métodos aplicados a un caso de estudio permitiendo conocer los datos en forma directa de la realidad, donde se aplicaron estrategias para la recolección de información y datos necesarios para la investigación. La información está distribuida en la organización y se obtiene a través de documentación histórica, procesos existentes, entrevista informal, observación directa, esto con el fin de conocer los problemas presentados y poder diseñar la arquitectura de integración de aplicaciones.

Siguiendo las características de una investigación descriptiva, los pasos a seguir en el trabajo son: (a) Investigación teórica y documental, (b) Definición de la propuesta sobre un caso de estudio y (c) Conclusiones y recomendaciones.

## **Investigación Teórica y Documental**

Esta fase de la metodología permite construir las bases teóricas de la investigación, a través de la consulta de libros, referencias electrónicas, trabajos de grado y artículos científicos certificados. A partir de la información recolectada en estos medios, se conformó el Capítulo II de este trabajo.

### **Definición de la propuesta sobre un caso de estudio**

Siendo un proyecto factible la investigación tiene tres (3) fases de desarrollo:

#### **Fase 1: Diagnóstica**

Con el fin de realizar cualquier investigación, se debe hacer una revisión exhaustiva de la documentación formal sobre el tema, realizando un análisis de la problemática y de la situación actual, adquiriendo los conocimientos necesarios que permitan delimitar el alcance del aporte al área en estudio.

La Ejecución de esta fase y del diagnóstico generado por medio de la aplicación de instrumentos como la entrevista y la observación directa, aplicada al área en estudio, en este sentido aplicada sobre el personal que elabora en la organización, específicamente en el departamento de Informática de la Alcaldía de Iribarren y el SEMAT permitió conocer las principales tareas que serán objeto de estudio de esta investigación, y que permitirán identificar la problemática presentada.

#### **Fase 2: Estudio de Factibilidad**

Durante esta fase se determina la posibilidad de construir la propuesta siguiendo un conjunto de criterios descritos a continuación:

## **Social**

Esta propuesta presenta una solución que será de gran provecho, ya que puede ser usada en todo tipo de organización que desee mejorar la gestión de sus procesos, logrando la automatización y optimización de los mismos.

Adicionalmente, la puesta en marcha de este tipo de propuestas ayuda a mejorar la toma de decisiones y permite que sus procesos se realicen de manera transparente sin la intervención del personal, minimizando la realización de actividades redundantes, permitiendo aprovechar el tiempo en otro tipo de actividades de carácter más importante, con lo que la probabilidad de que su rendimiento aumente es alta al disminuir el número de actividades a realizar en un día laboral.

Por lo expuesto anteriormente se considera esta solución factible desde el punto de vista social.

## **Tecnológica**

La factibilidad tecnológica se refiere a la disponibilidad de recursos tecnológicos para llevar a cabo la propuesta, dichos recursos están disponibles dentro de la organización objeto de estudio.

A nivel de software la organización cuenta con herramientas tecnológicas adecuada para lograr la integración de aplicaciones. Actualmente en el mercado existen sistemas ERP que están diseñados para incrementar la eficiencia en las operaciones de la compañía que lo utilice, además tiene la capacidad de adaptarse a las necesidades particulares de cada negocio. El ERP con el que cuenta es: SAP.

Para los casos de las organizaciones donde utilizan sistemas aislados existen tecnología que permite la integración de estas aplicaciones y patrones de integración que ya han sido aplicadas en organizaciones, dichos patrones pueden ser utilizados para el desarrollo de esta propuesta.

A nivel de hardware la organización cuenta con herramientas actuales y robustas, compuestas por Servidores, equipos de escritorio, así como la plataforma de red y comunicaciones que soporta las operaciones actuales.

Por lo expuesto anteriormente se considera esta solución factible desde el punto de vista tecnológico.

### **Económica**

El desarrollo de la propuesta no implica costo alguno ya que se cuenta con las herramientas necesarias para su puesta en marcha, razón por la cual se considera factible desde el punto de vista económico.

### **Operativa**

La factibilidad Operativa se refiere a si la propuesta una vez desarrollada, representara beneficios prácticos para la organización, en tal sentido, son muchos los beneficios que genera la propuesta a las organizaciones, entre ellos brinda la capacidad para mejorar la evaluación de ejecución de sus procesos de negocio en forma rápida y generando información de suma importancia para la toma de decisiones, logrando establecer un proceso de mejora continua dentro de la organización.

## **Fase 3: Diseño de Propuesta**

De acuerdo a los resultados obtenidos en el diagnóstico y en el estudio de factibilidad, se procede al diseño de la propuesta la cual está basada en el Diseño de una arquitectura de integración de aplicaciones sobre la base de intercambio de documentos.

La investigación estará dividida en tres (3) etapas que son descritas a continuación.

### **Etapas:**

**Etapas I: Realizar un diagnóstico sobre la manera en que se produce el intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.**

Durante esta etapa se realizará un estudio por medio de instrumentos como la entrevista y la observación directa para conocer la forma como los sistemas

involucrados intercambian información.

**Etapa II: Realizar un análisis orientado a servicios para conocer los requisitos funcionales y no funcionales presentes para el buen intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.**

Durante esta etapa se aplicara la estrategia usada por Julio Torres en su investigación titulada “Modelo de Integración de Aplicaciones Orientado a Objetivos e Ingeniería de Métodos” en la que utilizo el framework i\* y los Diagramas de caso de uso para especificar los requisitos en función de las necesidades de los usuarios, con la finalidad de clarificar el panorama que permite definir las interacciones entre las aplicaciones que se desean integrar, pudiendo definir sus interacciones en función a servicios a ser posteriormente consumidos bajo un enfoque arquitectural propio de este paradigma.

**Etapa III: Diseñar una arquitectura de integración de aplicaciones que satisfaga los requerimientos presentes en el caso de estudio, específicamente en la Alcaldía de Iribarren del Estado Lara**

Tomando en cuenta el objetivo alcanzado en la fase anterior, se plantea el diseño de una arquitectura de integración de aplicaciones, aplicando como metodología de desarrollo una metodología ágil como la programación Extrema (XP), se plantea la elaboración iterativa de artefactos entregables, en cuyo proceso de estructuración incremental se garantice el logro de los objetivos. Los entregables que se elaboraran en esta fase son:

Análisis Orientado a Patrones de Integración basado en Mensajes.

El modelado de los servicios alcanzados en la fase anterior utilizando el diagrama de actividad el cual muestra el proceso como un flujo de trabajo a través de una serie de acciones.

El diseño de los servicios haciendo uso de la tecnología Mule que es un ESB ligero que permite la integración de aplicaciones de forma rápida y sencilla.

El diseño de una arquitectura SOA por un ESB que administrara los servicios modelados en los entregables anteriores.

## CAPITULO IV

### PROPUESTA DEL ESTUDIO

Como se menciona en el desarrollo de este proyecto el objetivo a alcanzar es desarrollar una arquitectura que permita aumentar la interoperabilidad entre los sistemas utilizados dentro de la organización. Para ello en el capítulo anterior se definieron las fases del estudio, que serán desarrolladas en el presente capítulo.

Se utilizará la arquitectura orientada a servicios (SOA) que tiene como función enfocarse en los problemas de desarrollo de software, de una manera sencilla, rápida y ágil consiguiendo así una fácil adaptación a los cambios organizacionales y estructurales de las organizaciones. Para el cumplimiento del mismo es necesario adoptar una metodología que permita llevar un orden y ejecutar una serie de actividades. Para ello, se plantea usar la metodología de desarrollo ágil XP con entregables incrementales.

Esta investigación está dividida en tres (3) etapas de trabajo, a continuación se presentan los resultados de ejecución durante la aplicación de la primera etapa del proyecto.

**Etapas I: Realizar un diagnóstico sobre la manera en que se produce el intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.**

En el planteamiento del problema, se mencionó la necesidad una arquitectura de integración de aplicaciones que permita resolver los problemas e inconvenientes



presentados por la existencia de sistemas heterogéneos dentro de las organizaciones, específicamente en la Alcaldía de Iribarren del Estado Lara, objeto de estudio.

Primordialmente se requiere analizar el dominio al cual se le realizara el diagnóstico, Alcaldía de Iribarren del estado Lara específicamente al manejo de la información entre los sistemas involucrados en la Recaudación Municipal.

El Sistema de Recaudación de Impuesto consiste en la recepción de los pagos que deben realizar los Contribuyentes a la Alcaldía, por las diferentes obligaciones tributarias. La entidad encargada de velar por el cumplimiento de estas obligaciones tributarias es el Servicio Municipal de Administración Tributaria (SEMAT), el cual es un ente descentralizado que se encarga de desarrollar políticas y planes tributarios que permitan asumir el proceso de recaudación de los ingresos municipales, que contribuye a fortalecer la capacidad fiscal, y la eficiencia de los programas de administración de los impuestos, tasas y contribuciones, todo ello con la finalidad de mejorar la gestión y el autofinanciamiento del gobierno local.

Esta entidad, la cual funciona de la mano de la Dirección de Hacienda, organiza y mantiene al día los registros fiscales y de contribuyentes, previstos en las ordenanzas y reglamentos, para facilitar la administración de los tributos municipales Asimismo, orientan al contribuyente e inversionista acerca de los sistemas de información y múltiples Servicios que presta, a fin de administrar eficientemente los tributos municipales.

Para llevar a cabo estas funciones la Alcaldía de Iribarren de la mano con el SEMAT actualmente cuenta con aplicaciones que permiten realizar tareas que ayudan a controlar y gestionar lo referido a la recaudación, estas tareas son las siguientes:

- Declaración de Impuestos Municipales: son prestaciones en dinero que fija la ley con carácter general y obligatorio a cargo de personas para contribuir con el gasto público municipal
- Liquidación de Impuestos Municipales: consiste en la generación y recepción de los pagos que deben realizar los contribuyentes al Órgano o Ente encargado por las diferentes obligaciones tributarias.

- Solicitud y Aprobación de trámites. Se considera todo tipo de documento que solicita el contribuyente que este regido bajo una ordenanza o ley municipal.
- Cuadre de Movimientos: es la verificación de los cierres diarios realizados por todos los operadores en las diferentes entidades bancarias.

Los sistemas usados para el logro de estas tareas son:

SAP ERP Este sistema abarca prácticamente todos los aspectos de la administración, es decir, es usado dentro de la organización para realizar las funciones a nivel administrativo, entre las que se pueden nombrar, la declaración de impuesto, las facturas y pedidos generados de la liquidación de impuesto, generar solicitudes de trámites, realizar cuadro de los movimientos realizados, entre otras tareas.

La Alcaldía y el SEMAT para mejorar sus funcionalidades y prestar un mejor servicio a sus clientes dispuso un mecanismo que permitió al contribuyente realizar sus pagos en la comodidad de sus casas través de una aplicación por internet de pago de impuestos y de solicitud de tramites como por ejemplo constancia de no poseer vivienda o no contribuyente, entre otros trámites.

La aplicación web de pago de impuestos para su funcionamiento requiere de información contenida en SAP para el cálculo del impuesto, así como de datos de contribuyentes, pero no solo la aplicación web necesita de información contenida en SAP para su funcionamiento, SAP también requiere de información generada en el sistema de pago de impuesto por internet para generar las facturas de pagos de los contribuyentes y realizar el cuadro de los movimientos.

El intercambio de información entre los dos sistemas involucrados se realiza de la siguiente manera:

1. El sistema SAP contiene almacenada información relacionada con los contribuyentes e información necesaria para el cálculo de los tributos municipales como son: valor de la unidad tributaria, fecha de vencimiento de periodo entre otros, esta información se extrae mediante un Batch que son ejecutados en horas no pico, específicamente en horas de la noche, para no tener un impacto en el resto de procesos que

se ejecutan durante el día. Estos procesos Batch crean archivos de textos (.txt) separados por tabulación que representan información necesaria para el funcionamiento del sistema de pago de impuesto por internet, y colocadas en un directorio.

2. Los archivos de textos generados son consumidos por otra aplicación puente realizada en Visual Basic que toma estos archivos y actualiza la base de datos (MySQL) del sistema de pago de impuesto por internet. Esta actualización se realiza en horas de la madrugada.

3. La aplicación web de pago de impuesto por internet, permite a los contribuyentes generar los pagos de sus tributos y realizar solicitud de trámites, esta información generada es almacenada en la base de datos y debe ser transferida a SAP, para ello, el sistema realizado en Visual Basic consume esta información desde la base de datos y genera archivos de textos (.txt) separados por tabulación y colocadas en un directorio que luego serán consumidos por SAP. Estos archivos de textos se generan tres (3) veces al día por la gran cantidad de demanda que el sistema web ha logrado, de modo que al ser procesados en SAP no perjudique el funcionamiento de las operaciones realizadas en SAP a nivel administrativo.

4. Los archivos son procesados en SAP para generar una solicitud de pedido, que registra el pago generado por el contribuyente.

5. Las entidades bancarias inscritas para el manejo de los tributos municipales, generan archivos de textos (.txt) que contienen todos los pagos generados en el día.

6. Estos archivos una vez que son enviados al SEMAT son procesados en SAP para compensar los pagos, actualizar el periodo de pago de los contribuyentes y actualizar el cuadro de los movimientos.

Este proceso se puede visualizar en la figura no. 21

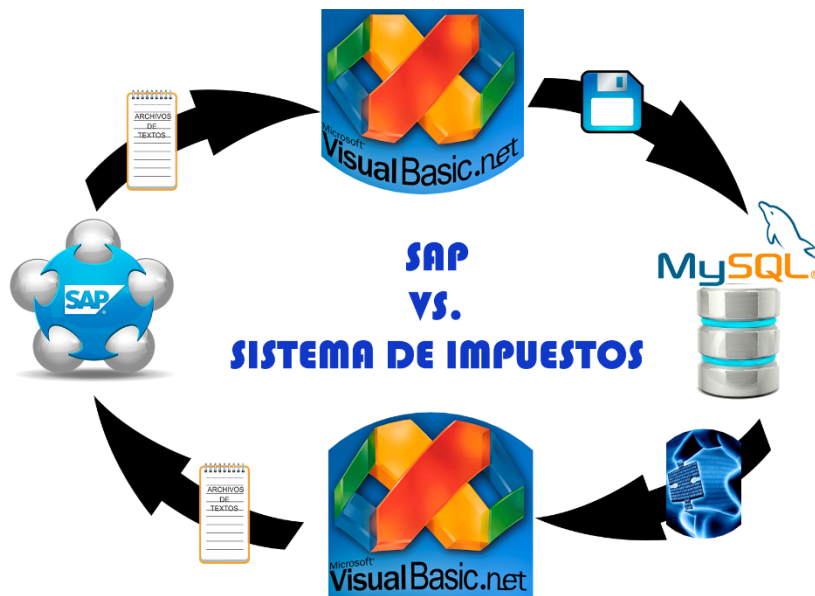


Figura 21. Proceso de Integración SAP y Aplicación Web de Pago de Impuesto.  
 Autor: el investigador

Estos procesos se realizan diariamente y a pesar de que benefician al contribuyente porque pueden realizar sus pagos desde la comodidad de sus casas, y disminuyen la cantidad de personas que se dirigen al SEMAT a realizar los pagos, también presenta sus desventajas o dificultades tales como:

- La actualización puede fallar, cuando alguno de los sistemas involucrados no se encuentra disponible. El manejo de la información entre las distintas aplicaciones en ocasiones por un retraso en la transferencia de archivos da como resultado del uso de información obsoleta.
- La respuesta a los contribuyente no es inmediata.
- Debe existir un sistema que sirva de puente entre las dos aplicaciones a integrar, que permita adaptar los formatos para que sea procesada por los sistemas involucrados.
- Al momento de generar y consumir los archivos se lleva a la decisión de producirlos o ejecutarlos ya sea en el horario nocturno, semanal, trimestral, etc. Las Aplicaciones solo pueden usar los archivos cuando esté disponible.

Las deducciones obtenidas por medio del diagnóstico realizado según la observación directa de los procesos y una entrevista informal realizada al coordinador de Informática del SEMAT apoyan la elección de un enfoque que permita abordar las dificultades encontradas. Luego de indagar sobre enfoques de integración de aplicaciones se propone una arquitectura de integración orientada a servicios (SOA) usando patrones de integración basadas en mensajes y lograr la integración de la información de los procesos de pago de impuestos por internet y la facturación en SAP. Dicha arquitectura será diseñada con el fin de solucionar el problema de integración que se presenta en la Alcaldía de Iribarren como también para otras organizaciones que tengan este mismo tipo de problemas independientemente del sistema que usen, y permitir que las aplicaciones trabajen en conjunto en forma electrónica en lugar del intercambio o contacto físico directo, con la finalidad de disminuir lo más posible este último. Bajo este particular lograr características como: transparencia, velocidad de respuesta, reducción de costos de la transacción y la comunicación "online" con los participantes, tratamiento y transmisión electrónica de datos, entre otros, siempre y cuando cumpla con las principales tareas del dominio.

**Etapas II: Realizar un análisis orientado a servicios para conocer los requisitos funcionales y no funcionales presentes para el buen intercambio de información entre los sistemas SAP y la Aplicación web de pago de impuestos.**

Al comenzar un proyecto de software, el analista conoce muy poco acerca del problema a resolver. La elicitación de requerimientos es el proceso que consiste en adquirir todo lo relevante, el conocimiento necesario para producir un modelo de requerimientos, es decir, es descubrir cuál es el problema que se debe resolver y clarificar todos los elementos que se requiere para cualquier solución de software a implementar.

Los requisitos de software se categorizan en funcionales y no funcionales, los funcionales definen las funciones del sistema, y los no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema. Describen una

restricción sobre el sistema. La elicitación de requisitos es considerada como uno de los procesos más importantes dentro de la Ingeniería de Requisitos.

Para la elicitación de estos requisitos se usara una secuencia de actividades propuesta por Julio Torres en su investigación titulada “Modelo de Integración de Aplicaciones Orientado a Objetivos e Ingeniería de Métodos” en la que utilizo el framework i\* y los Diagramas de caso de uso para especificar los requisitos en función de las necesidades de los usuarios, con la finalidad de clarificar el panorama que permite definir las interacciones entre las aplicaciones que se desean integrar, pudiendo definir sus interacciones en función a servicios a ser posteriormente consumidos bajo un enfoque arquitectural propio de este paradigma. Estas actividades se muestran en la figura no. 22

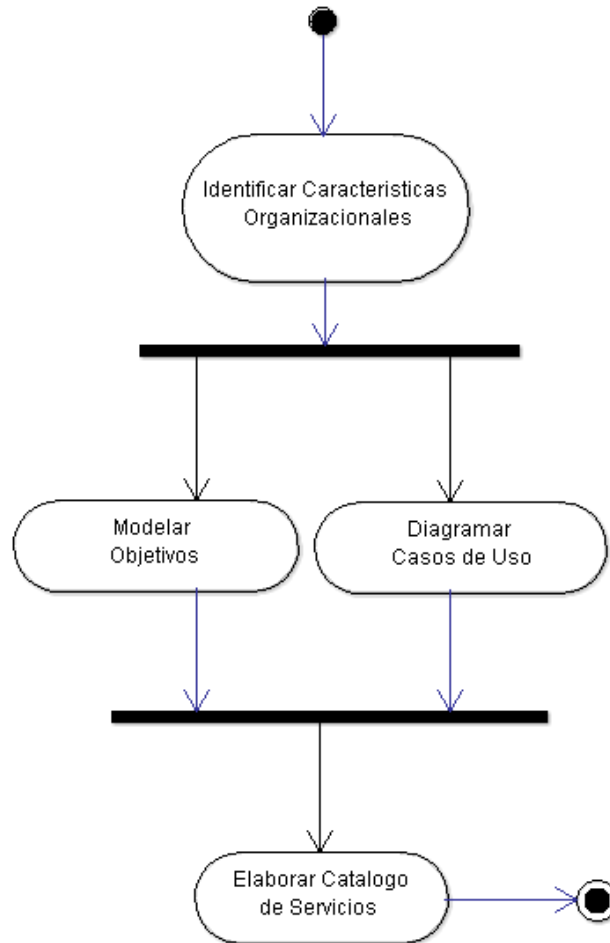


Figura N°22. Secuencia de actividades Elicitación de servicios.  
Fuente: Julio (2013 p. 105)

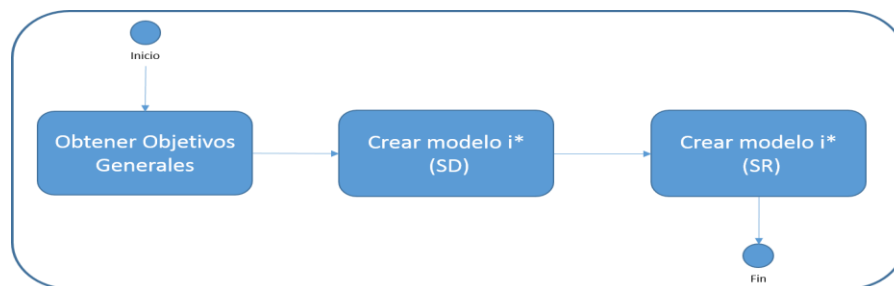
De lo anterior se desprende que luego del diagnostico realizado a la organización se deben realizar dos (2) etapas que son el modelo de objetivos con el uso del framework i\* y el diagrama de casos de usos usando la notación UML, con los resultados obtenidos en estas etapas se identificaran los servicios candidatos y se elaborara el catalogo de servicios. Estas etapas se definen y se detallan a continuación:

### ***Modelar Objetivos:***

Para nuestro enfoque utilizamos el framework i\* para obtener los requerimientos dependiendo de las necesidades obtenidas en el etapa anterior donde se realizo un diagnostico de las ncesidades en la organización en relación a la integración de aplicaciones. Con i\* se representan de manera gráfica los objetivos estratégicos, decisionales e informacionales de cada actor involucrado en el proceso y las relaciones entre los mismos.

Esta técnica consiste en dos modelos: el modelo de dependencia estratégica (SD) para describir las relaciones de dependencia entre los distintos actores en su contexto organizacional, y el modelo de motivos estratégicos (SR), que se utiliza para describir los intereses y preocupaciones de cada actor y de cómo podrían abordarse.

El concepto central en los modelos de i\* son las intenciones de los actores, ya que son los agentes de la organización, los que dependen de los objetivos a alcanzar, las tareas a realizar y de los recursos que se deben suministrar. La secuencia de actividades para el modelo de objetivos con i\* se muestra en la figura no 23.



*Figura N°23. Secuencia de actividades para modelar Objetivos usando Framework i\*.*

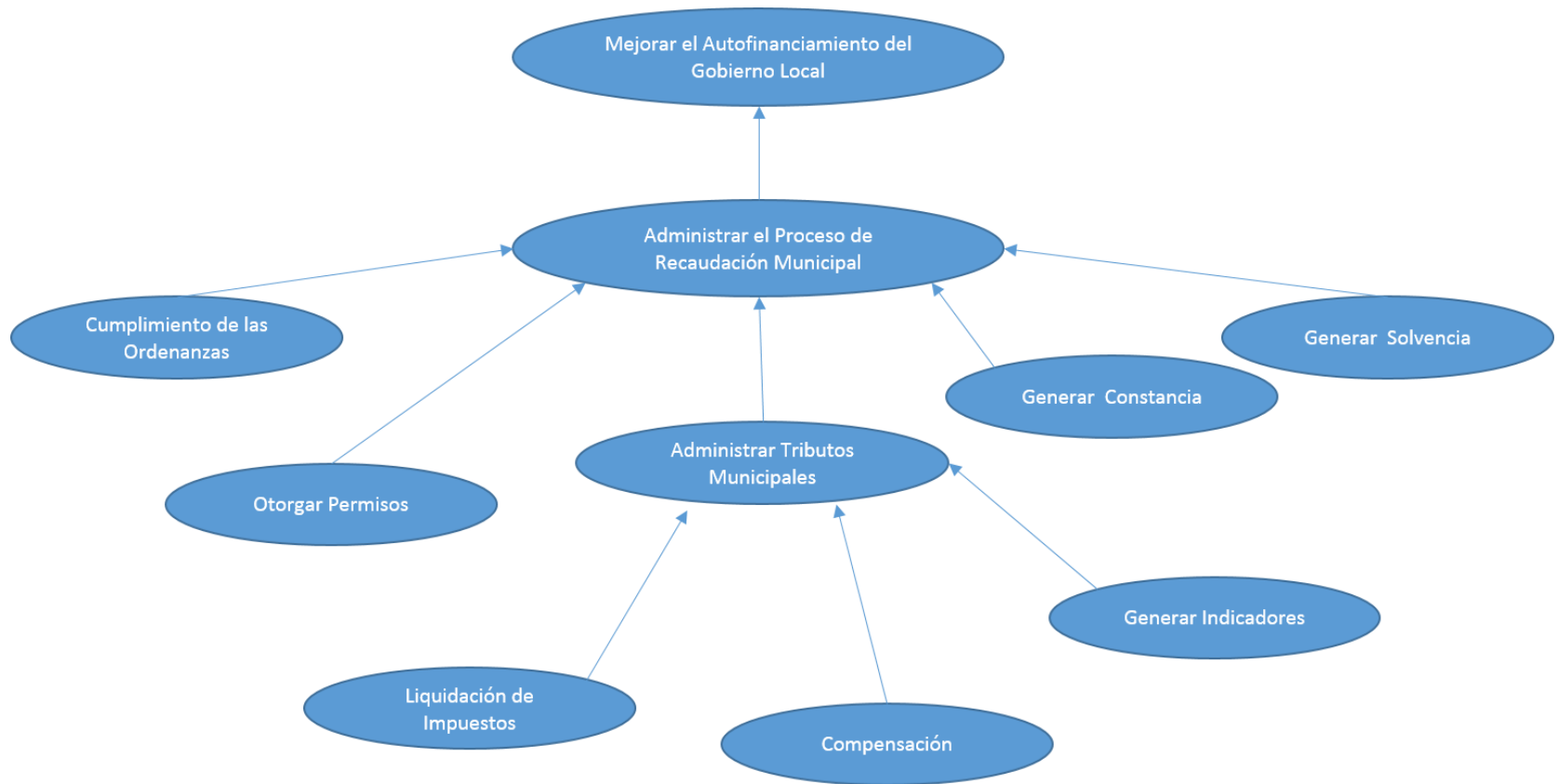
*Fuente: Autor de la Investigación*

Para iniciar el proceso de modelado de objetivos, se deben identificar los usuarios o actores involucrados en los procesos. Para el caso de estudio: existe un usuario con privilegios, un usuario contribuyente y un usuario no contribuyente. El usuario con privilegio que determinay registra los valores necesarios para la declaración, así como también de realizar tareas administrativas necesarias en la declaración municipal entre otras tareas. Los usuarios contribuyentes se encargan de realizar sus pagos para cumplir con el municipio a nivel tributario y también realizar otros trámites como solvencias y constancias. Finalmente, los usuarios no contribuyentes quienes podrán realizar solicitud de constancias de no contribuyen. Una vez identificado los actores se procede a obtener los objetivos generales.

### ***Iteración 1***

Para nuestro caso de estudio los objetivos que tiene el SEMAT en relación a la declaración de impuestos municipales. Este proceso se ilustra en la figura no. 24





*Figura N°24. Diagrama de Objetivo*  
*Fuente: Autor de la Investigación*

De lo anterior, es destacable aclarar que en función del alcance de la presente investigación se tomarán exclusivamente los procesos orientados a satisfacer la Administración de Tributos Municipales, los procesos para generar trámites (solvencia, generar constancia) entre otros, en este particular en la siguiente iteración se analizarán los procesos en función de sus objetivos, tareas y usuarios involucrados.

### ***Iteración 2***

En esta etapa se realizó el diagrama SD, en la cual se utilizarán los resultados obtenidos de la iteración anterior.

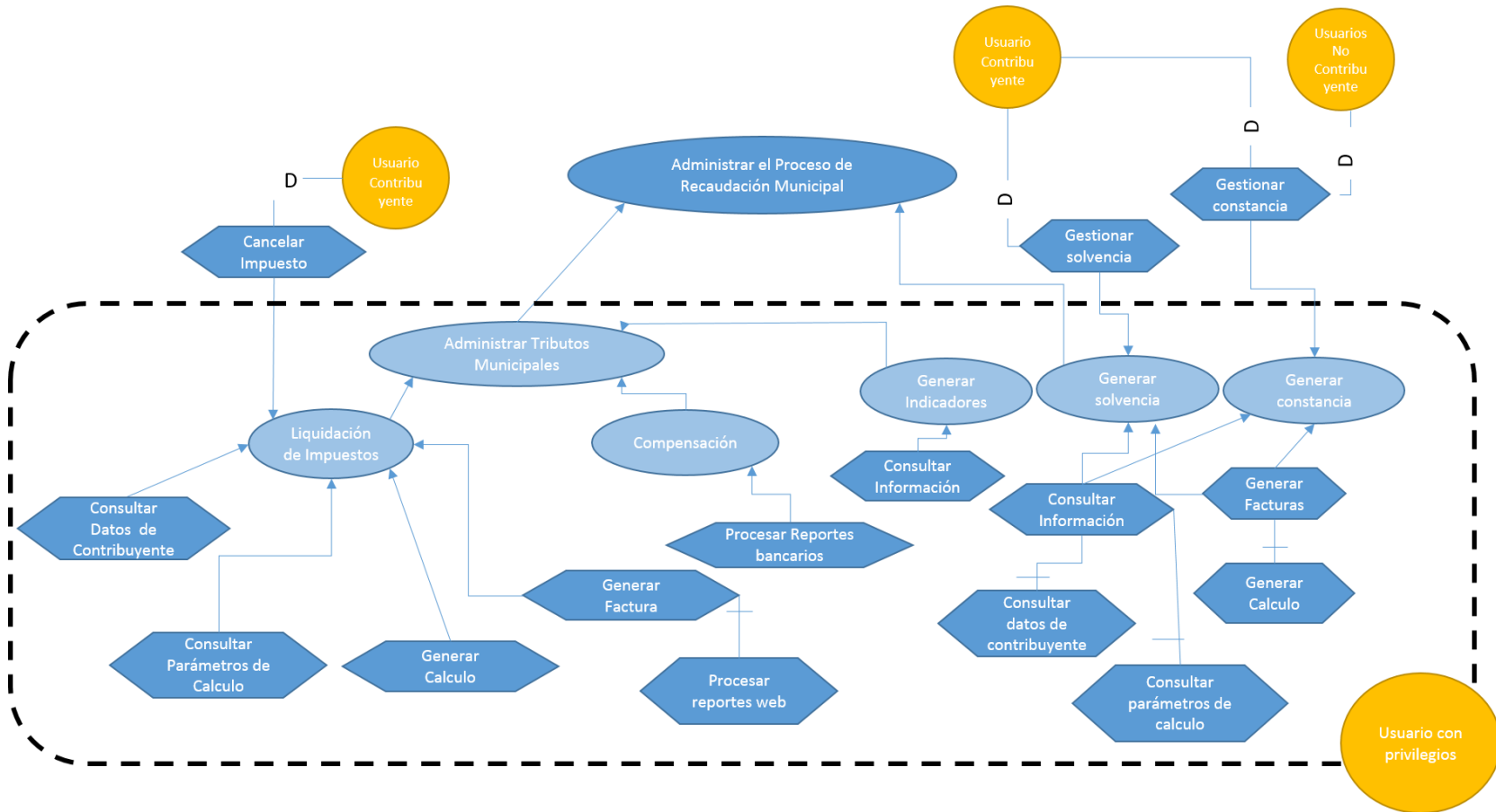


Figura N°25. Modelo SD con I\*  
Fuente: Autor de la Investigación

En el modelo SD anterior observamos lo siguiente:

- Para cumplir con el objetivo de la liquidación de impuesto por parte del usuario contribuyente o registrado por el usuario con privilegios, es posible mediante la incorporación de la tarea o procesos internos “calcular información”, “calcular monto a pagar”, “generar factura”
- Se debe realizar las compensaciones por parte del usuario con privilegios, por lo que se ha considerado un requerimiento, este es posible mediante la realización de la tarea procesar reportes bancarios.
- Se debe realizar la tarea “generar indicadores” realizada por el usuario con privilegios, es posible mediante la tarea “consultar información”.
- Se debe cumplir con el objetivo generar tramite, activado por el usuario contribuyente o por el usuario no contribuyente, por lo que se deben realizar las tareas “consultar información” y “generar factura” con sus procesos internos.

### ***Iteración 3***

Una vez creado el modelo SD en la etapa anterior, en esta etapa se realiza el diagrama SR, en el que se identificaran actores, tareas y recursos.

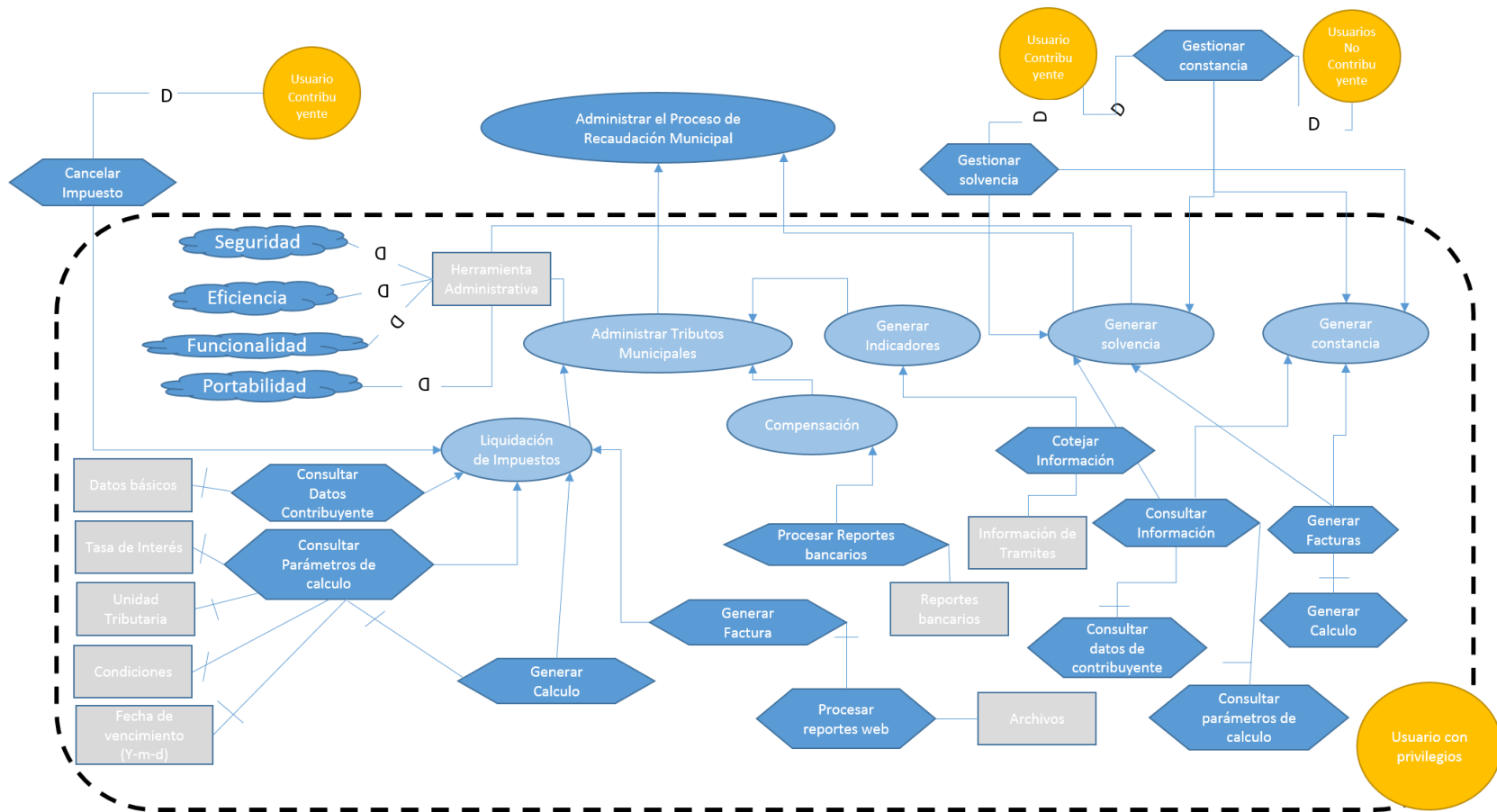
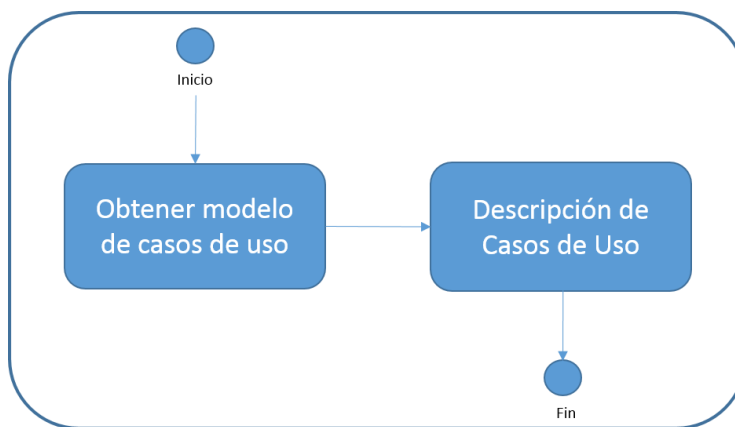


Figura N°26. Modelo SR con I\*  
Fuente: Autor de la Investigación

### ***Generar Casos de Uso:***

Para generar los casos de uso se definieron un conjunto de pasos que establecen la relación entre los elementos especificados en las iteraciones para crear modelos con el framework i\* y un modelo de casos de uso y sus correspondiente descripción. Este proceso se representa en la figura no. 27



*Figura N°27. Secuencia de actividades para modelar casos de uso UML.*

*Fuente: Autor de la Investigación*

Para iniciar el proceso de caso de uso usando la notación UML, se utilizaran un conjunto de metas u objetivos descritos en los modelos resultates en las iteraciones anteriores, en los que se obtubieron un conjunto de objetivos y que en forma posterior fueron deribados en relaciones de dependencia de recursos o tareas entre actores organizaciones (Usuarios con privilegios, usuario contribuyente, usuario no contribuyente).

### ***Iteración 1***

El diagrama de Casos de Uso Asociado a los objetivos organizacionales descritos en la sección anterior, se ilustra en la figura no. 28

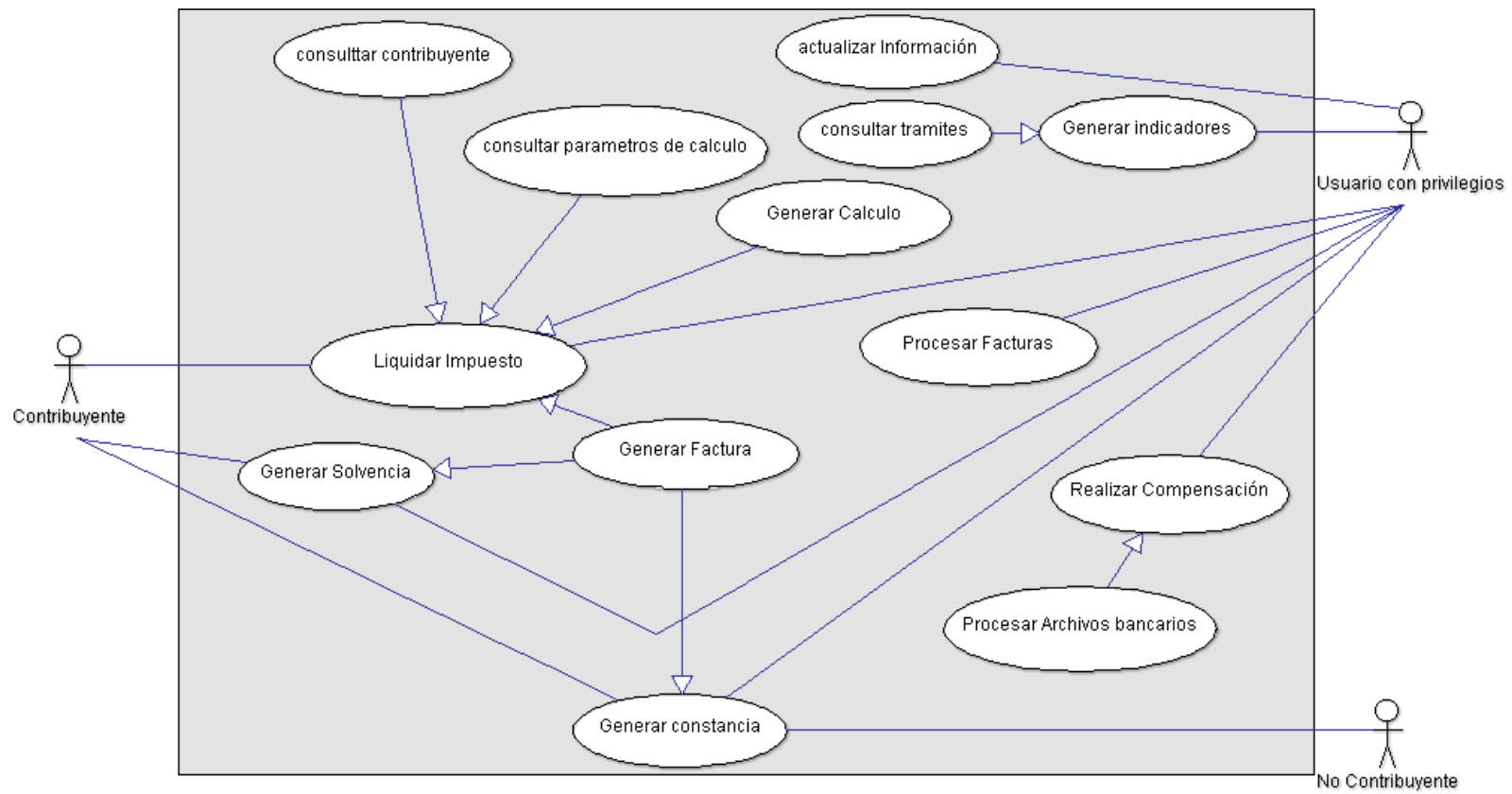


Figura N°28. Diagrama de casos de uso UML.

Fuente: Autor de la Investigación

El modelo de casos de uso representa el conjunto de funcionalidades que debe ser satisfecho por la arquitectura de solución propuesta. El método a aplicar es en base a una traza que viene de los objetivos, casos de uso y los servicios que servirán para satisfacerlos.

### ***Iteración 2***

Para cada caso de uso representado en la iteración anterior es necesario crear un modelo de descripción, el cual utiliza plantillas para mostrar la secuencia de eventos que intercambian los actores y el sistema. A continuación se detallan:

<b>Ref.:</b>	<b>UC-1</b>
<b>Caso de Uso:</b>	Liquidar impuesto
<b>Descripción :</b>	Permite al contribuyente declarar sus impuestos
<b>Pre-condición:</b>	Usuario contribuyente, usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción que permite declarar sus impuestos.</li> <li>2. El usuario introduce el código de la licencia a declarar.</li> <li>3. El sistema despliega la información referente a la licencia.</li> <li>4. El usuario introduce los datos necesarios para calcular el impuesto a cancelar y seleccionar generar cálculo.</li> <li>5. El Sistema realiza la transacción generar calculo y muestra el total a pagar.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El Sistema No encuentra el código de licencia y envía un mensaje indicando licencia no encontrada.</li> <li>4. El Sistema No realiza el cálculo y envía un mensaje indicando lo sucedido.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de liquidación de impuesto.

*Cuadro N° 8 Caso de uso: Liquidar Impuesto.*

*Fuente: autor de la investigación.*



<b>Ref.: UC-1-1</b>	
<b>Caso de Uso:</b>	Consultar Contribuyente.
<b>Descripción :</b>	Obtención de un conjunto de información referente del contribuyente necesario para la liquidación de impuesto.
<b>Pre-condición:</b>	UC-1
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario activa la opción correspondiente a “liquidar impuesto”.</li> <li>2. El buscador localiza información relacionada con el contribuyente.</li> <li>3. El sistema devuelve el conjunto de información.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El buscador NO localiza información referente al contribuyente.</li> <li>3. El sistema devuelve mensaje correspondiente al resultado anterior.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de consulta de datos de contribuyente.

*Cuadro N° 9. Caso de uso: Consultar Datos de Contribuyente.*

*Fuente: autor de la investigación.*

<b>Ref.: UC-1-2</b>	
<b>Caso de Uso:</b>	Consultar Parámetros de Calculo.
<b>Descripción :</b>	Obtención de un conjunto de información de parámetros de cálculo.
<b>Pre-condición:</b>	UC-1
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario activa la opción correspondiente a “liquidar impuesto”.</li> <li>2. El buscador localiza información relacionada a parámetros de cálculo.</li> <li>3. El sistema devuelve el conjunto de información.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El buscador NO localiza información referente a parámetros de cálculo.</li> <li>3. El sistema devuelve mensaje correspondiente al resultado anterior.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de consulta parámetros de cálculo.

*Cuadro N° 10. Caso de uso: Consultar parámetros de cálculo.*

*Fuente: autor de la investigación.*

<b>Ref.:</b>	<b>UC-1-3</b>
<b>Caso de Uso:</b>	Generar Calculo
<b>Descripción :</b>	Permite calcular el monto a pagar por el contribuyentes
<b>Pre-condición:</b>	UC-1
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso generar calculo</li> <li>2. El sistema emite el resultado del monto a pagar por el contribuyente.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El Sistema No emite monto a pagar porque el contribuyente está al día con sus impuestos municipales.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de generar cálculo.

*Cuadro N° 11 Caso de uso: Generar Cálculo.*

*Fuente: autor de la investigación.*

<b>Ref.:</b>	<b>UC-1-3</b>
<b>Caso de Uso:</b>	Generar Factura
<b>Descripción :</b>	Permite generar la factura de la liquidación municipal
<b>Pre-condición:</b>	UC-1
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso generar factura</li> <li>2. El sistema emite la factura de la liquidación municipal.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El Sistema No emite la factura de pago.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo la emisión de la factura.

*Cuadro N° 12 Caso de uso: Generar Factura.*

*Fuente: autor de la investigación.*

<b>Ref.:</b>	<b>UC-2</b>
<b>Caso de Uso:</b>	Realizar Compensación
<b>Descripción :</b>	Permite compensar todos los pagos realizados por los usuarios
<b>Pre-condición:</b>	usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso realizar compensación</li> <li>2. El sistema modifica el último pago de los contribuyentes por el periodo cancelado en la factura compensada.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El Sistema No modifica el último pago del contribuyente.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de compensar pago.

*Cuadro N° 13 Caso de uso: Realizar Compensación.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-2-1</b>
<b>Caso de Uso:</b>	Procesar Archivos Bancarios
<b>Descripción :</b>	Permite cargar la información proveniente de la entidad bancaria sobre los pagos realizados en el día
<b>Pre-condición:</b>	UC-2
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso procesar archivos bancarios</li> <li>2. El sistema emite un mensaje de archivos cargados con éxito.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El Sistema No emite mensaje de éxito.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de generar archivos bancarios.

*Cuadro N° 14 Caso de uso: Procesar Archivos Bancarios.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-3</b>
<b>Caso de Uso:</b>	Generar Solvencia
<b>Descripción :</b>	Solicitud de Solvencia
<b>Pre-condición:</b>	Usuario Contribuyente, usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario activa la opción correspondiente a “Solicitar Solvencia”.</li> <li>2. El sistema muestra los tipos de solvencia que un usuario pueden realizar</li> <li>3. El usuario selecciona la solvencia a solicitar.</li> <li>4. El sistema genera el monto a pagar.</li> <li>5. El usuario selecciona generar factura.</li> <li>6. El sistema muestra la factura.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. No existen solvencias disponibles para realizar en el sistema.</li> <li>3. El usuario no selecciona solvencia.</li> <li>4. El sistema no genera el monto que el usuario debe pagar</li> <li>5. El usuario selecciona cancelar y no genera la factura</li> </ol>	
<b>Post-condición:</b>	El trámite queda registrado, junto con el código de la factura que está asociado a la solvencia.

*Cuadro N° 15 Caso de uso: Generar Solvencia.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-4</b>
<b>Caso de Uso:</b>	Generar Constancia
<b>Descripción :</b>	Solicitud de Constancia
<b>Pre-condición:</b>	Usuario Contribuyente, Usuario no contribuyente, usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario activa la opción correspondiente a “Solicitar Constancia”.</li> <li>2. El sistema muestra los tipos de constancia que un usuario pueden realizar</li> <li>3. El usuario selecciona la constancia a solicitar.</li> <li>4. El sistema genera el monto a pagar.</li> <li>5. El usuario selecciona generar factura.</li> <li>6. El sistema muestra la factura.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. No existen constancias disponibles para realizar en el sistema.</li> <li>3. El usuario no selecciona constancia.</li> <li>4. El sistema no genera el monto que el usuario debe pagar</li> <li>5. El usuario selecciona cancelar y no genera la factura</li> </ol>	
<b>Post-condición:</b>	El trámite queda registrado, junto con el código de la factura que está asociado a la constancia.

*Cuadro N° 16 Caso de uso: Generar Constancia.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-5</b>
<b>Caso de Uso:</b>	Generar Indicadores
<b>Descripción :</b>	Obtención de indicadores que permitan hacer seguimiento y evaluaciones
<b>Pre-condición:</b>	UC-1 o UC-2
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción que permite la generación de indicadores</li> <li>2. El sistema genera el reporte solicitado.</li> <li>3. El sistema permite la visualización del reporte solicitado en una nueva ventana.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El sistema presenta problemas al momento de generar el reporte.</li> <li>3. El sistema envía un mensaje al usuario indicando la imposibilidad de mostrar el reporte solicitado.</li> </ol>	
<b>Post-condición:</b>	El reporte de indicadores es generado y visualizado satisfactoriamente.

*Cuadro N° 17 Caso de uso: Generar Indicadores.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-5-1</b>
<b>Caso de Uso:</b>	Consultar Tramites
<b>Descripción :</b>	Permite consultar la información de tramites generados
<b>Pre-condición:</b>	UC-5
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso consultar tramites</li> <li>2. El sistema muestra información de los trámites realizados.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El sistema No muestra información porque no existen trámites generados.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de consultar trámites.

*Cuadro N° 18 Caso de uso: Consultar Trámites.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-6</b>
<b>Caso de Uso:</b>	Procesar Facturación
<b>Descripción :</b>	Permite procesar las facturas generadas por los contribuyentes por la web
<b>Pre-condición:</b>	Usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso procesar facturación</li> <li>2. El sistema busca las facturas generadas en la web</li> <li>3. El sistema procesa todas las facturas generadas</li> <li>4. El sistema envía un mensaje de facturación procesadas satisfactoriamente</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>2. El sistema No encuentra facturas de trámites generados.</li> <li>3. El sistema No puede procesar la facturación</li> <li>4. El sistema No muestra información porque no existen trámites generados.</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de procesar facturación.

*Cuadro N° 19 Caso de uso: Procesar Facturación.*

Fuente: autor de la investigación.

<b>Ref.:</b>	<b>UC-7</b>
<b>Caso de Uso:</b>	Actualizar Información
<b>Descripción :</b>	Permite actualizar información que se encuentra en SAP a la WEB
<b>Pre-condición:</b>	Usuario con privilegios
<b>Curso Normal</b>	
<ol style="list-style-type: none"> <li>1. El sistema activa el caso de uso actualizar información</li> <li>2. El sistema ejecuta transacción para buscar información</li> <li>3. El sistema exporta información a archivo</li> <li>4. El sistema actualiza la información a partir de la información exportada.</li> </ol>	
<b>Curso Alterno</b>	
<ol style="list-style-type: none"> <li>3. El sistema No encuentra información para procesar</li> <li>4. El sistema No puede actualizar la información</li> </ol>	
<b>Post-condición:</b>	El Sistema ha llevado a cabo las operaciones de actualizar Información

*Cuadro N° 20 Caso de uso: Actualizar Información*

Fuente: autor de la investigación.

Los diagramas de casos y usos en conjunto con su descripción son muy importantes desde el punto de vista de análisis de requerimientos, los casos de uso se pueden utilizar como una herramienta de comunicación hacia los clientes, ya que en ella quedan perfectamente plasmadas todas las funcionalidades que va a tener el software en un lenguaje natural para el cliente.

***Catalogo de Servicios:***

Tomando en cuenta los modelos generados con el framework i\* (ver Figuras N° 24-26), y los casos de usos con sus respectivas descripciones (ver Figuras N° 27) y (ver Cuadros N° 8-20) se identifican los objetivos principales de la organización, se elaboró un catálogo de servicios, dicho formato se muestra en la tabla no. 21. Según formato definido por Julio T. (ob. cit), adaptado a las necesidades relacionadas al presente estudio.

Objetivos	Iteraciones		Servicios
	Actores	Casos de Uso	
Liquidación de impuestos	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> <li>• Usuario Contribuyente</li> </ul>	<ul style="list-style-type: none"> <li>• Consultar Datos contribuyente</li> <li>• Consultar parámetros de calculo</li> <li>• Generar Calculo</li> <li>• Generar Factura</li> </ul>	<ul style="list-style-type: none"> <li>• S-1: Consultar Contribuyente</li> <li>• S-2: Consultar parámetros de calculo</li> <li>• S-3: Generar Calculo</li> <li>• S-4: Generar factura</li> </ul>
Generar Compensación	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> </ul>	<ul style="list-style-type: none"> <li>• Generar compensación</li> <li>• Generar archivos bancarios</li> </ul>	<ul style="list-style-type: none"> <li>• S-5: Compensar pagos</li> <li>• S-6: Procesar reportes bancarios</li> </ul>

Generar Solvencia	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> <li>• Usuario Contribuyente</li> </ul>	<ul style="list-style-type: none"> <li>• Generar Factura</li> <li>• Generar Solvencia</li> </ul>	<ul style="list-style-type: none"> <li>• S-7: Generar solvencia</li> </ul>
Generar Constancia	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> <li>• Usuario Contribuyente</li> <li>• Usuario NO Contribuyente</li> </ul>	<ul style="list-style-type: none"> <li>• Generar Factura</li> <li>• Generar Constancia</li> </ul>	<ul style="list-style-type: none"> <li>• S-8: Generar Constancia</li> </ul>
Generar Indicadores	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> </ul>	<ul style="list-style-type: none"> <li>• Generar Indicadores</li> <li>• Consultar tramites</li> </ul>	<ul style="list-style-type: none"> <li>• S-9: Generar indicadores</li> <li>• S-10: Consultar tramites</li> </ul>
Procesar Facturación	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> </ul>	<ul style="list-style-type: none"> <li>• Procesar facturación</li> </ul>	<ul style="list-style-type: none"> <li>• S-11: Procesar Facturación</li> </ul>
Actualizar Información	<ul style="list-style-type: none"> <li>• Usuario con privilegios</li> </ul>	<ul style="list-style-type: none"> <li>• Actualizar Información</li> </ul>	<ul style="list-style-type: none"> <li>• S-12: Actualizar Información</li> </ul>

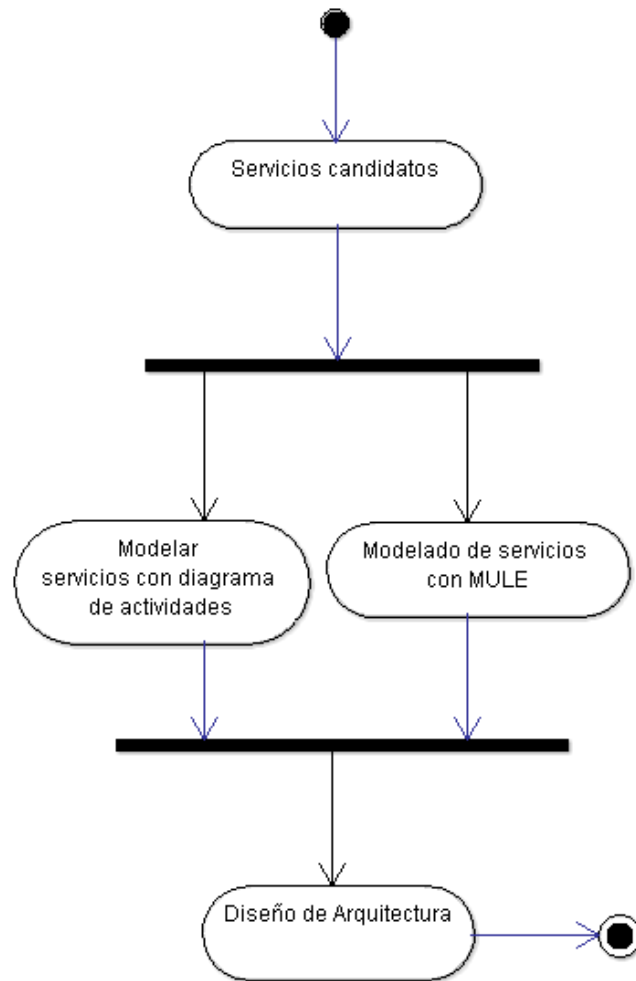
*Cuadro N° 21 Caso de uso: Catálogo de Servicios.*

Fuente: autor de la investigación.

### **Etapa III: Diseñar una arquitectura de integración de aplicaciones que satisfaga los requerimientos presentes en el caso de estudio, específicamente en la Alcaldía de Iribarren del Estado Lara**

Una vez descrita y culminada la etapa anterior, en la que se describieron modelos que dieron como resultados requerimientos y servicios candidatos que serán usados para el diseño de una arquitectura que satisfaga las necesidades de integración entre aplicaciones, se procede a aplicar estos servicios en una secuencia de actividades que al finalizar den como resultado una arquitectura que satisfaga estas necesidades organizacionales. Esta secuencia se describe a continuación en la figura no. 29





*Figura N°29. Secuencia de actividades para crear arquitectura*  
 Fuente: Autor de la Investigación

En la figura no. 29 muestra la secuencia de actividades para crear la arquitectura de integración de aplicaciones. En la que se parte de los servicios resultantes de la etapa anterior.

Para el diseño de la arquitectura y cumplir con la secuencia de actividades se usara la metodología ágil XP. Partiendo por una etapa en la que se analizarán los patrones que serán usados en la arquitectura que se desea diseñar.

### ***Iteración 1***

#### **Análisis orientado a patrones de integración basado en mensajes:**


Los ambientes de aplicaciones empresariales están caracterizados por su variabilidad y necesidad de facilitar el cambio de las mismas y mantener los sistemas conectados para compartir datos con aplicaciones probablemente desarrolladas en la misma empresa o adquiridas de otras empresas, estas necesidades conllevan a crear como solución una arquitectura usando estilo de integración basado en Mensajería que permita integrar múltiples aplicaciones intercambiando paquetes de datos de manera frecuente, inmediata, confiable y asincrónicamente usando formatos personalizables.




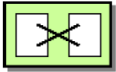
El uso de mensajería se ha considerado como un mecanismo maduro para la integración de aplicaciones con patrones que explican cómo hacer un buen uso de ella.

La Mensajería utilizando patrones de integración brinda confiabilidad, asíncronía y alto rendimiento. A través de la utilización de una Arquitectura de integración de aplicaciones que hará posible que las aplicaciones actuales y futuras se integren para lograr los objetivos planteados anteriormente.

Al hablar de patrones se habla de solución, una solución para problemas recurrentes donde se fomente su reutilización y la reutilización de arquitecturas de software que han tenido éxito y que facilitan la reestructuración de sistemas que inicialmente no fueron desarrollados para el uso de patrones.

Por lo tanto, se desea crear una arquitectura de integración de aplicaciones haciendo uso de los patrones basados en mensajes sugeridos por Enterprise Integration Patterns de Hohpe y Woolf (2003) la cual proporciona una guía para implementar los patrones de integración de aplicaciones basados en mensajes. Alguno de estos patrones se muestra en la siguiente tabla:

Patrón	Descripción
 <p data-bbox="462 1753 771 1795">Punto Final de Mensaje</p>	<p data-bbox="857 1711 1396 1795">Define una conexión desde una aplicación hasta un canal de mensaje.</p>

 <p>Canal de Mensaje</p>	<p>Permite a las aplicaciones conectarse unas con otras</p>
 <p>Adaptador de Mensaje</p>	<p>Define como los sistema integrados se pueden conectar al ESB para poder recibir o enviar mensajes</p>
 <p>Ruteo Basado en Contenido</p>	<p>Usado para enrutar los mensajes en base al contenido del mensaje.</p>
 <p>Transformador</p>	<p>Es usado cuando el contenido del mensaje necesita ser cambiado antes de ser enviado</p>

Cuadro N° 22 patrones de integración de aplicaciones basados en mensajes

Fuente: Hohpe y Woolf (2003)

Los patrones de integración pueden ayudar a completar la amplia brecha entre la visión de alto nivel de integración y la implementación real del sistema. Con el fin de conectar dos sistemas a través de una solución de integración, una serie de cosas tiene que suceder. Invariablemente, algunos datos tienen que ser transportados de una aplicación a la siguiente.

Los datos que son transportados de una aplicación a otra y deben ser entendidos por ambos extremos. Dos elementos proporcionan esta función básica. *El canal del Mensaje* donde se coloca el mensaje, un fragmento de datos que tiene un significado acordado por ambas aplicaciones a integrar. Debido a que el formato de datos interna de una aplicación no puede ser cambiado a menudo el middleware debe proporcionar algún mecanismo para convertir el formato de datos de una aplicación en la del otro. Llamamos a este paso *traducción o transformador*.

Hasta el momento podemos enviar datos desde un sistema a otro y dar cabida a las diferencias en los formatos de datos. Es momento de encaminar el mensaje al sistema destino, es allí donde *el patrón de Ruteo* inspecciona el contenido de un mensaje y lo enruta a otro canal basado en el contenido del mensaje

Existen situaciones en las que se necesita enviar un mensaje en las que se necesita tomar ciertas decisiones, Por ejemplo consumir un mensaje a la vez, pero publicando varios mensajes en orden secuencial. Para este tipo de casos se hace uso de un diagrama mostrado en la figura 30 que facilite esta decisión.

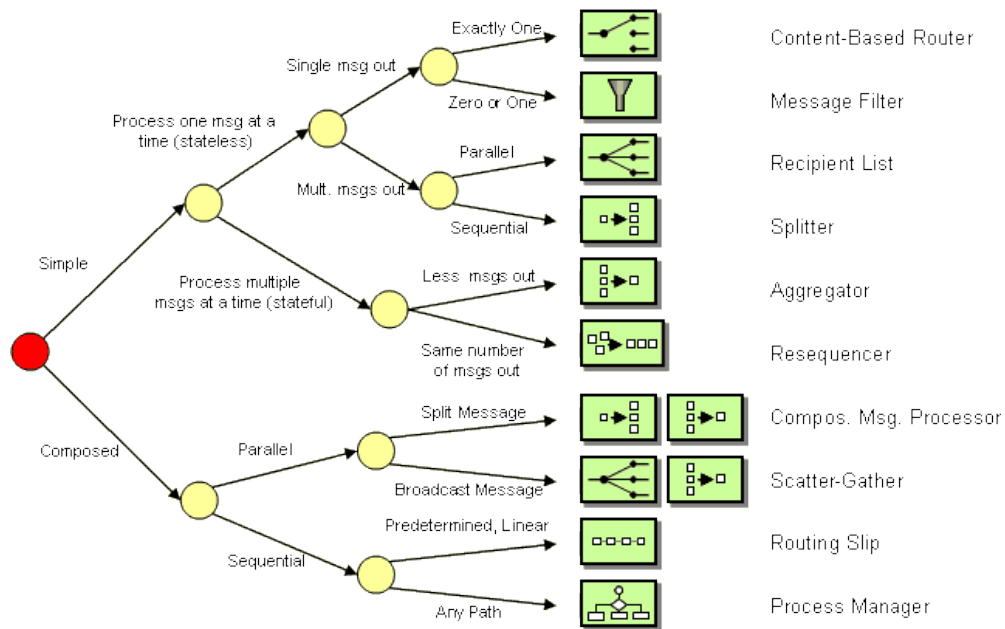


Figura N°30. Diagrama de decisión para seleccionar el patrón correcto

Fuente: <http://www.eaipatterns.com/MessageRoutingIntro.html>

El diagrama de decisión permite encontrar el modelo adecuado, ayuda a seleccionar el patrón correcto dependiendo del propósito. Por ejemplo si se desea consumir un mensaje a la vez, pero publica varios mensajes en orden secuencial, se debe utilizar un *Splitter*. De igual forma sucede con el resto de los patrones.

Una vez que podemos mover datos de un sistema a otro por medio de un canal, adaptarse a las diferencias en el formato de datos usando un transformador, pasar los datos a los sistemas necesarios y supervisar el rendimiento de la solución, ahora solo hace falta conectar los sistemas a una solución de integración haciendo uso de un patrón de *mensaje de punto final* para conectar el sistema de forma explícita a la solución de integración. El punto final puede ser una pieza especial de código o

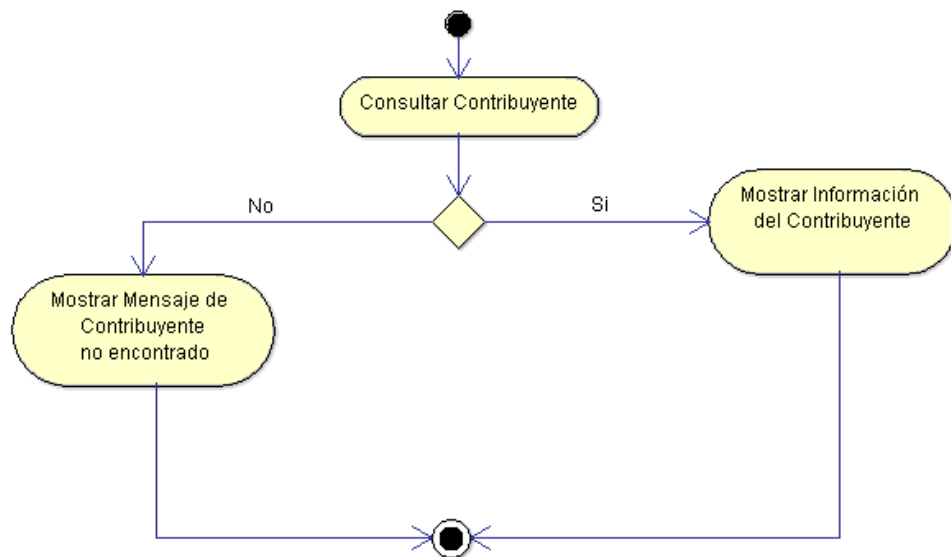
un adaptador de canal proporcionado por un proveedor de software de integración.

Para la utilización de estos patrones en la arquitectura a diseñar se utilizara Mule ESB como software para desplegar el componente de integración, Mule es un Bus de servicio empresarial (ESB) de licencia libre basado en Java y una plataforma de integración que permite a los desarrolladores conectar aplicaciones de forma rápida y sencilla intercambiando datos entre ellos.

### *Iteración 2*

#### **Modelado de Servicios usando diagrama de actividades:**

Con el uso de los diagramas de actividades se modela el comportamiento de un sistema, muestra una visión simplificada de lo que ocurre durante una operación o proceso. El modelado de los servicios candidatos mostrados en el cuadro no. 21 se muestran a continuación:



*Figura N°31. Diagrama de Actividad Consultar Contribuyente*

Fuente: Autor de la Investigación

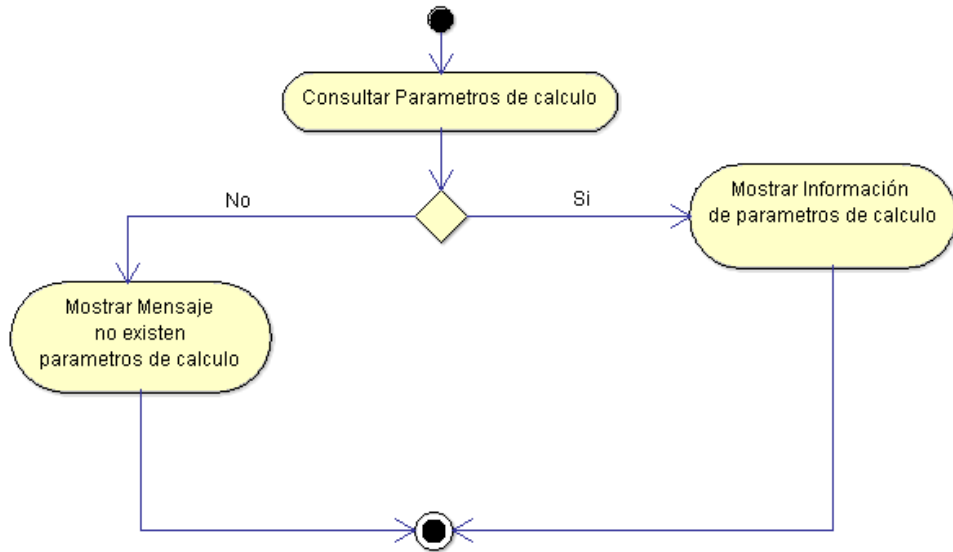


Figura N°32. Diagrama de Actividad Consultar Parámetros de Cálculo  
Fuente: Autor de la Investigación

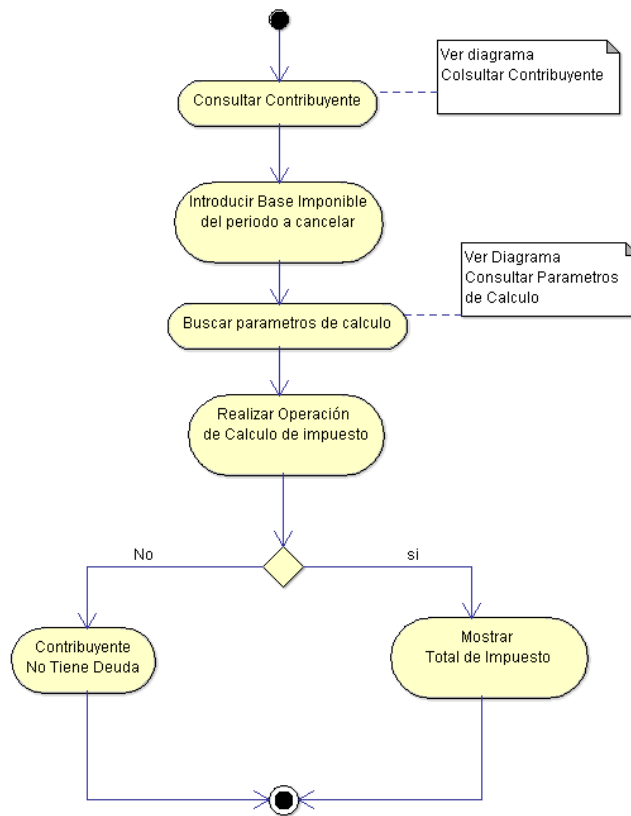
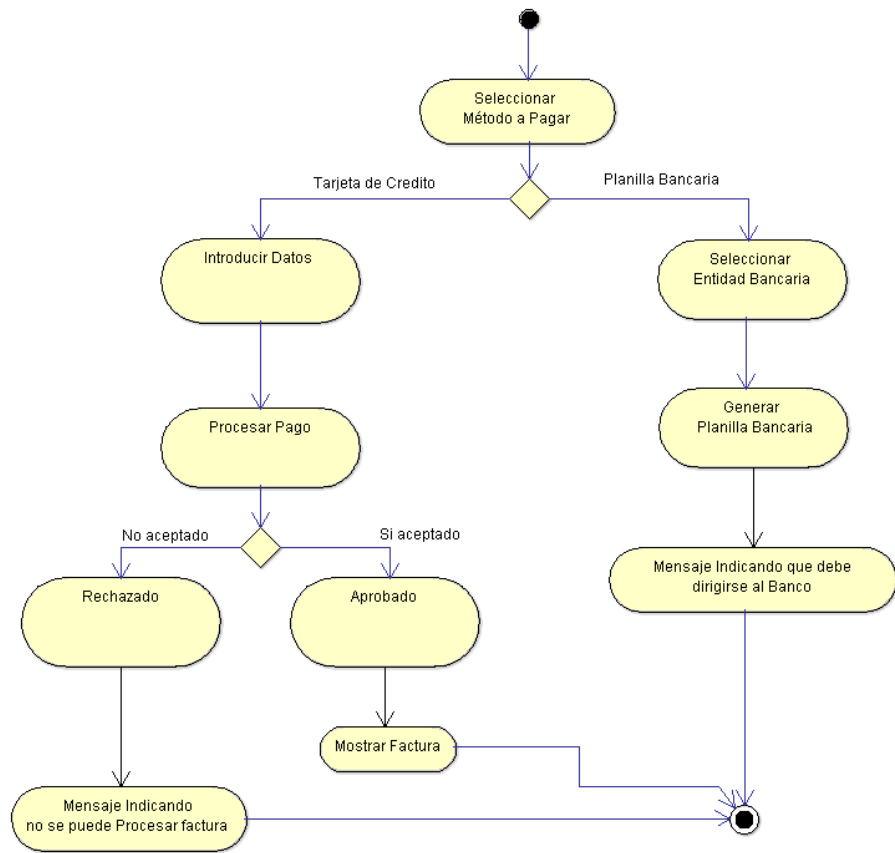


Figura N°33. Diagrama de Actividad Generar Calculo  
 Fuente: Autor de la Investigación



*Figura N°34. Diagrama de Actividad Generar Factura*  
 Fuente: Autor de la Investigación



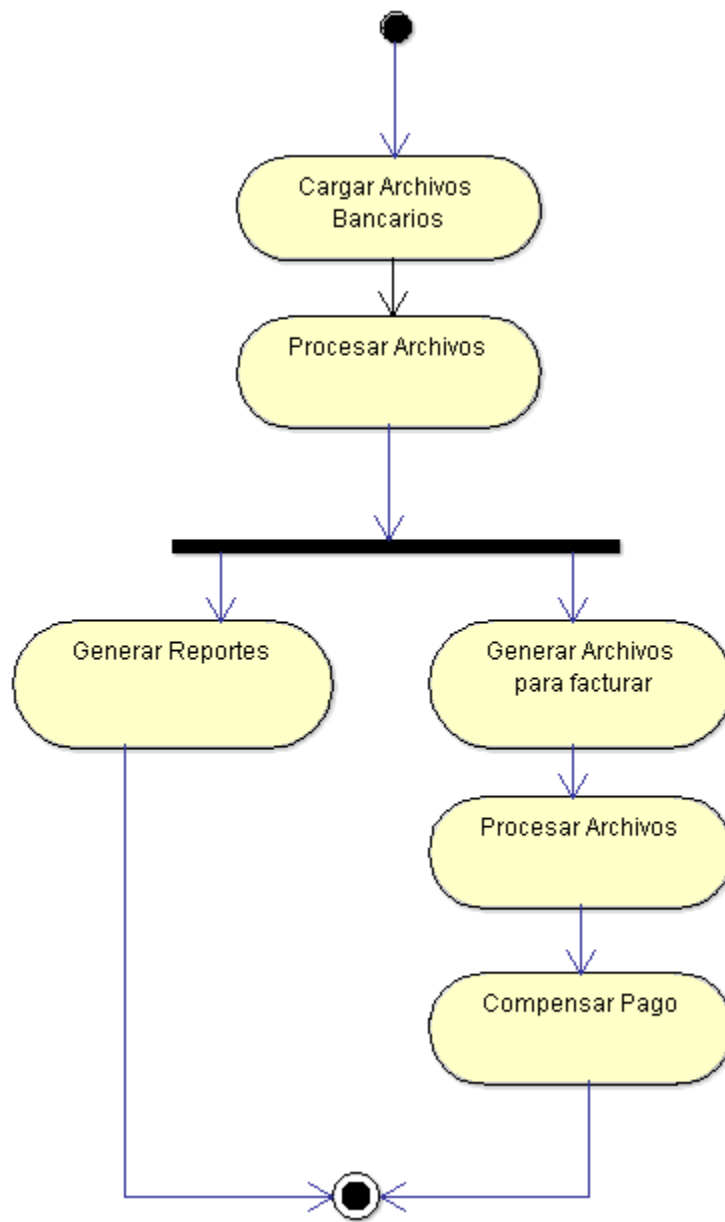
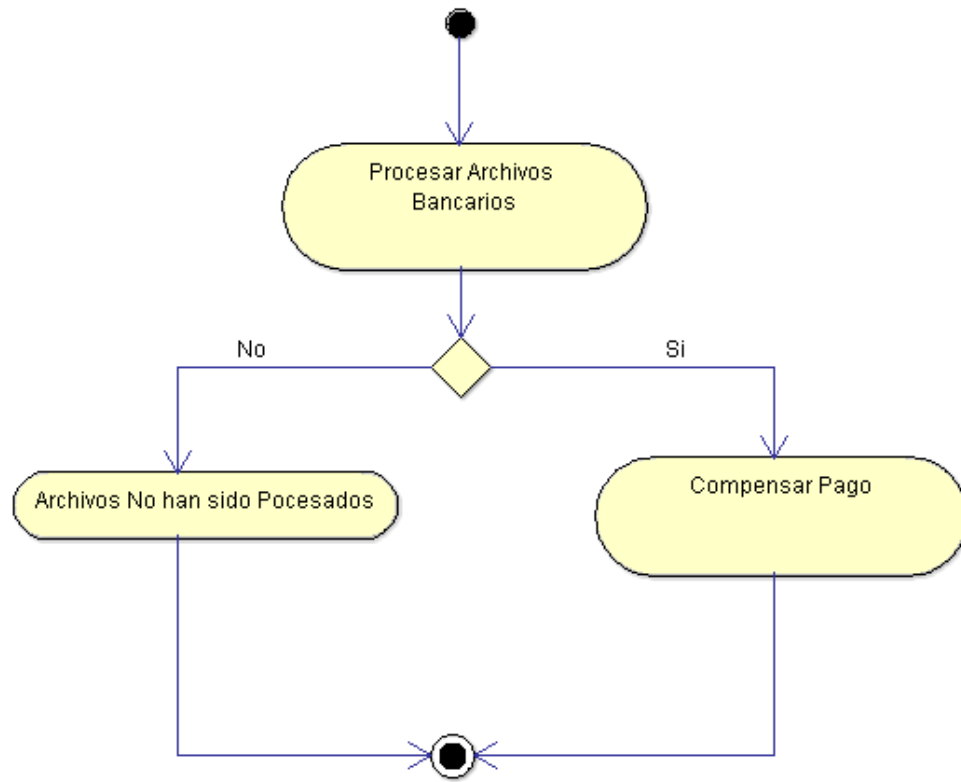


Figura N°35 Diagrama de Actividad Compensar Pago  
Fuente: Autor de la Investigación



*Figura N°36 Diagrama de Actividad Procesar Archivos Bancarios*  
Fuente: Autor de la Investigación

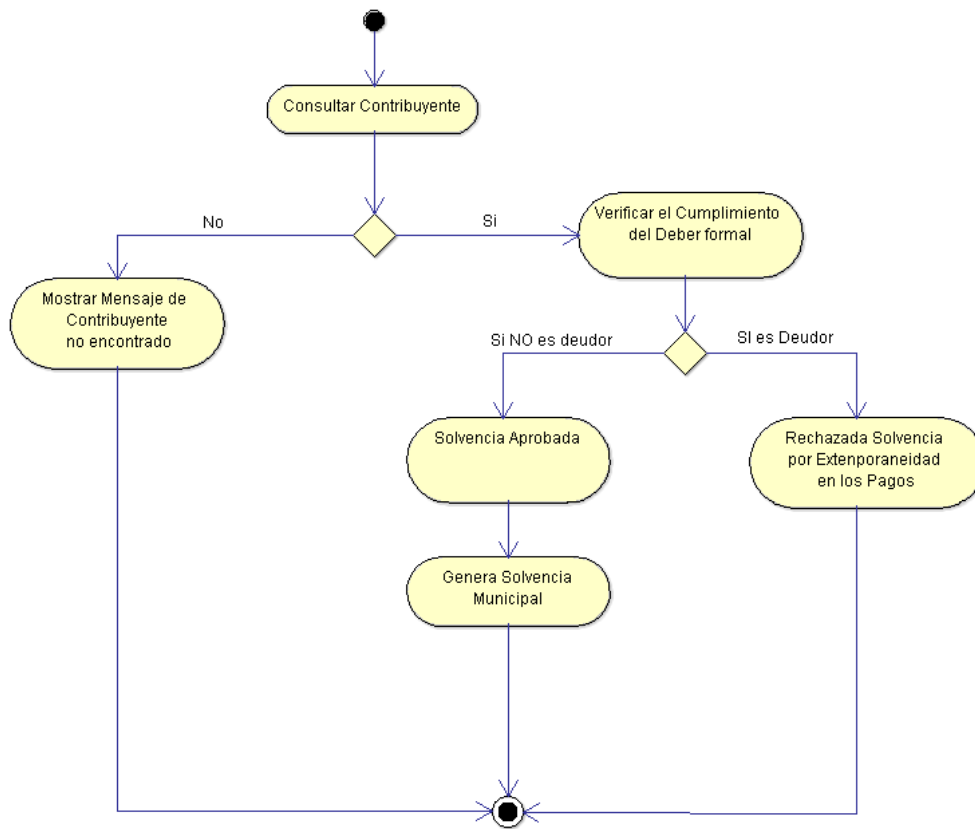


Figura N°37 Diagrama de Actividad Generar Solvencia  
 Fuente: Autor de la Investigación

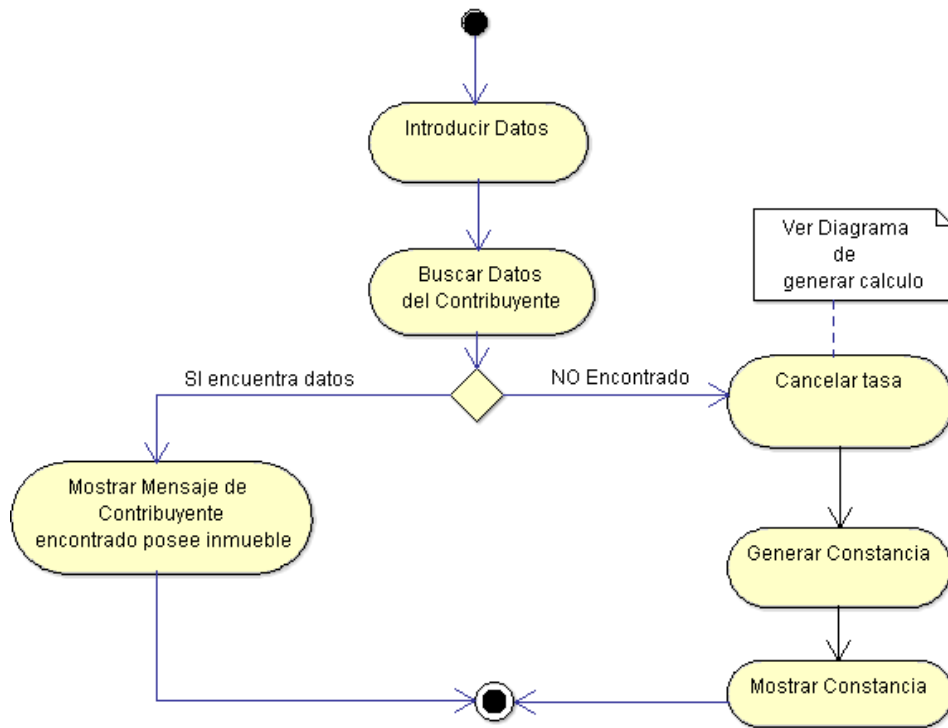


Figura N°38. Diagrama de Actividad Generar Constancia  
Fuente: Autor de la Investigación

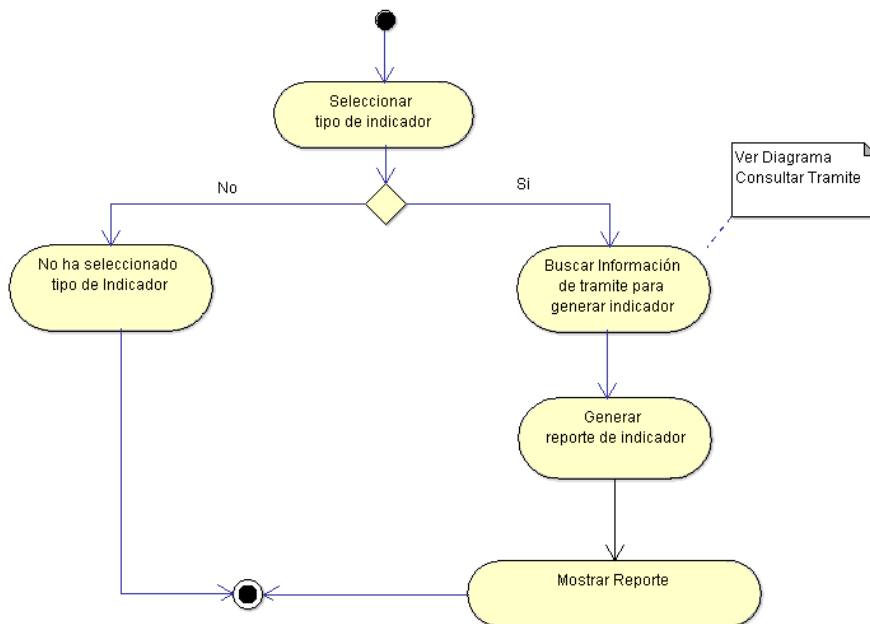


Figura N°39. Diagrama de Actividad Generar Indicadores  
Fuente: Autor de la Investigación

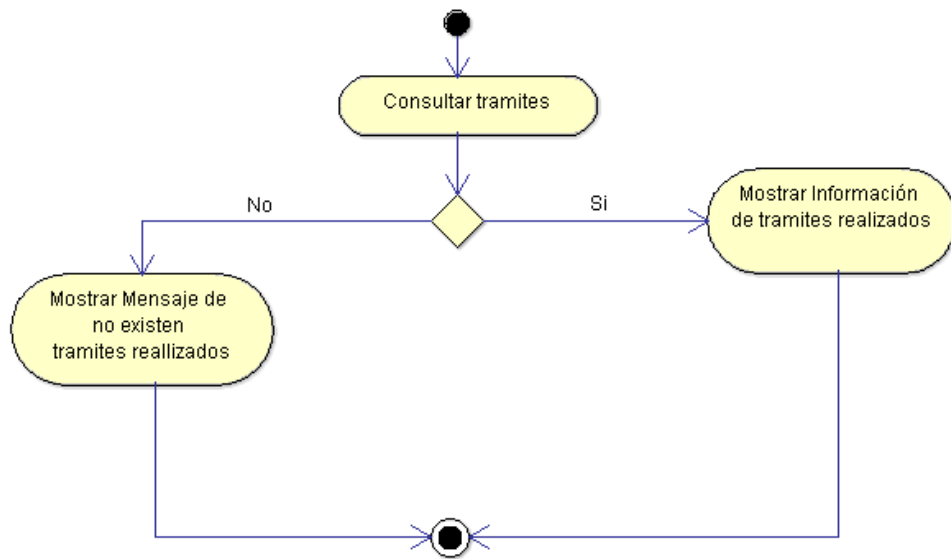


Figura N°40 Diagrama de Actividad Consultar Tramite  
 Fuente: Autor de la Investigación

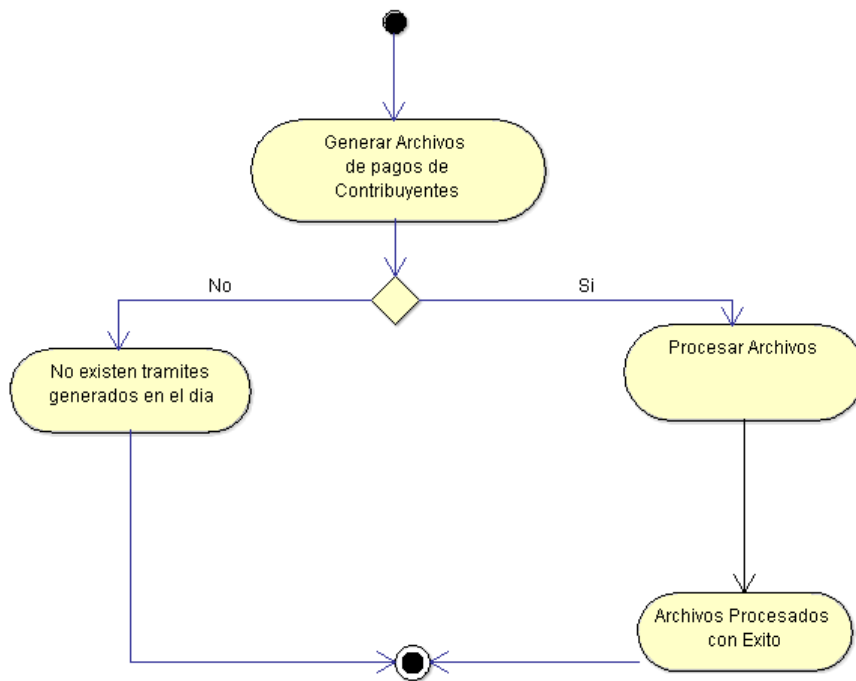


Figura N°41. Diagrama de Actividad Procesar Facturación  
 Fuente: Autor de la Investigación

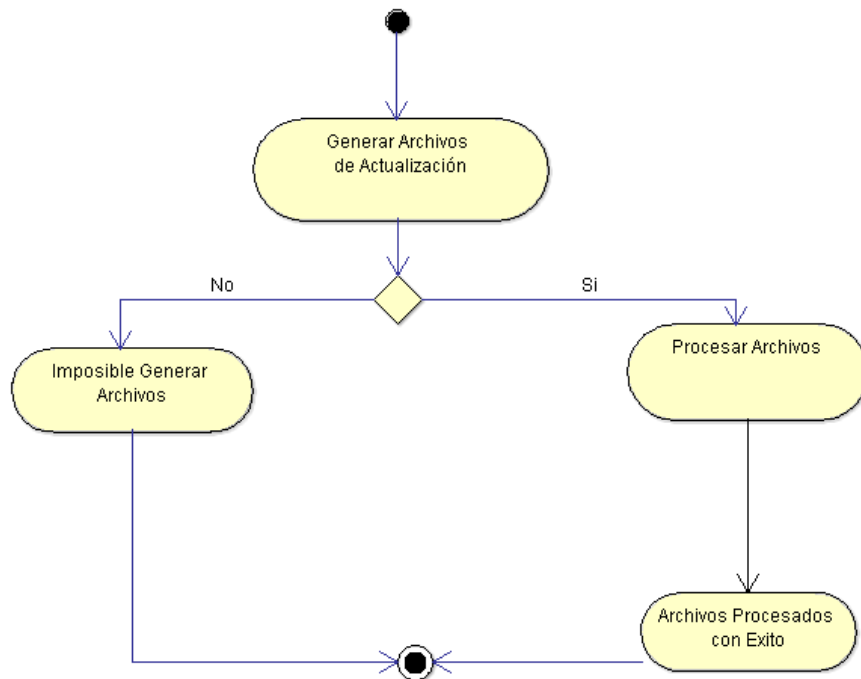


Figura N°42 Diagrama de Actividad Actualizar Información

Fuente: Autor de la Investigación

Con el uso de estos diagramas de actividad se muestra el comportamiento y de la forma como deben interactuar los servicios.

### ***Iteración 3***

#### **Modelado de Servicios usando MULE:**

A esta altura ya se tiene especificado el comportamiento de los servicios y la forma cómo interactúan. Sin embargo, a un no se han definido aspectos necesarios para la aplicación de mensajería en el desarrollo de la solución, como por ejemplo tipos de mensajes a intercambiar, ruteo de estos mensajes, transformación de los mismos si corresponde, canales sobre los que se enviarán, etc. Para de unir estos aspectos se utiliza la plataforma de desarrollo MULE que permite mostrar mediante un lenguaje visual los aspectos anteriormente mencionados utilizando los patrones de

mensajería introducidos en el capítulo II.

Los patrones de mensajería que se utilizarán para el diseño de las comunicaciones entre los sistemas que interactúan en el proceso de recaudación Municipal de la Alcaldía de Iribarren son los siguientes:

**Request/Reply (Solicitud-respuesta):** donde una aplicación solicitante envía una solicitud, una aplicación replier recibe la solicitud y devuelve una respuesta, y el solicitante recibe la respuesta.

**Message Endpoint:** permite conectar una aplicación a un canal de mensajería, la aplicación puede utilizarlo para enviar o recibir mensajes.

**Router Basada en contenido:** examina el contenido del mensaje y encamina el mensaje en un canal diferente, basado en los datos contenidos en el mensaje.

**Message Translator:** Permite transformar un mensaje desde el formato con el cual fue recibido al esperado por cada una de las entidades receptoras.

**Message Filter:** Utilizado para eliminar mensajes no deseados de un canal basado en un conjunto de criterios.

A continuación se detallan los servicios modelados mediante la plataforma de desarrollo MULE y haciendo uso de patrones de Mensajería descritos anteriormente.

Se modela el servicio de consultar contribuyente, el cual utiliza JMS para enviar peticiones SOAP

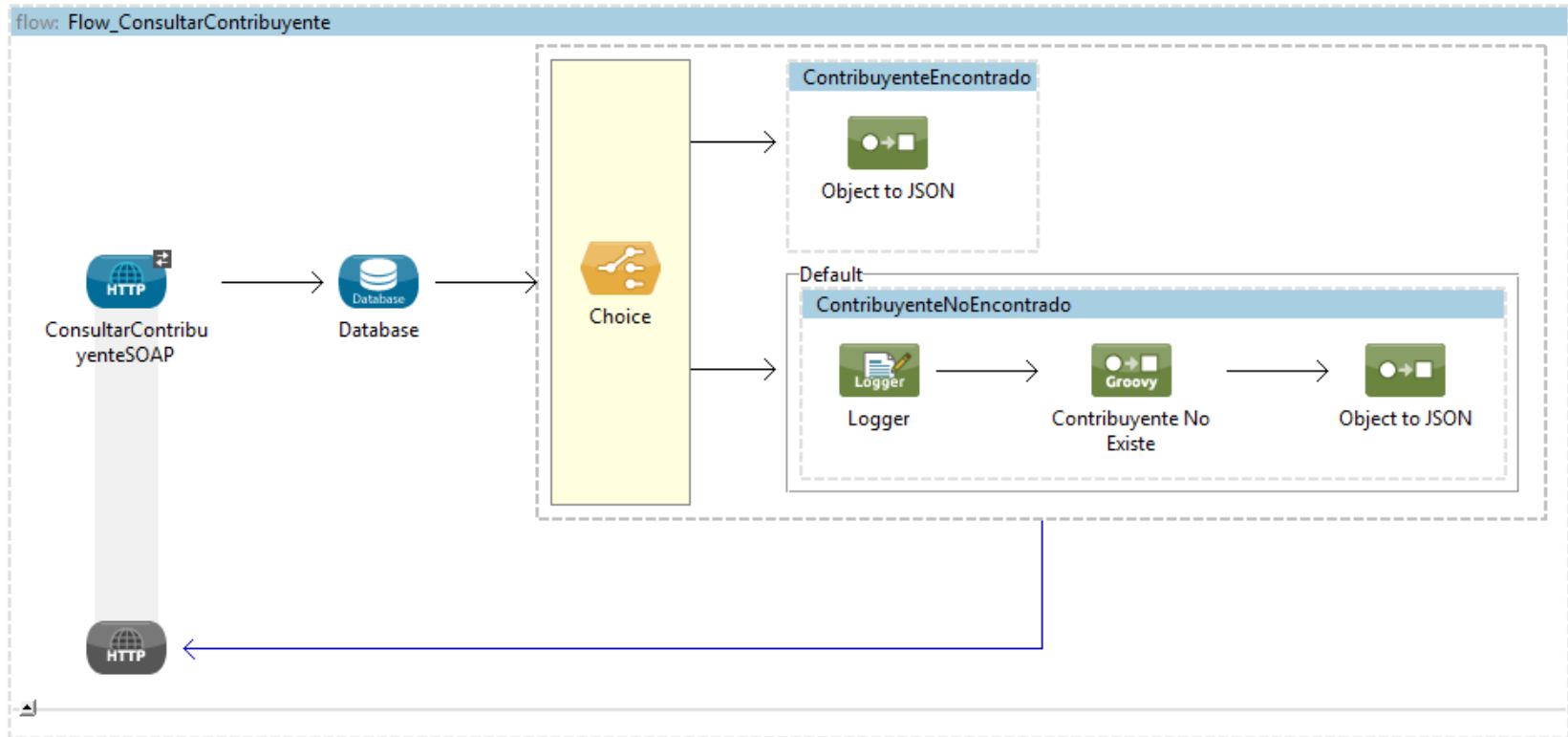


Figura N°43 Flujo con Mule de Consultar Contribuyente  
Fuente: Autor de la Investigación



El flujo diseñado en la figura No. 43 contiene un servicio web de SOAP que recibirá peticiones para consultar información del contribuyente en una base de datos, el resultado obtenido de la consulta sql es pasada por un flujo de control Choice que valida el resultado, en el caso que el contribuyente existe obtiene la información en formato JSON. En el caso que el contribuyente no exista, se registra el resultado usando un componente registrador Logger para registrar la carga útil del mensaje en los archivos de registro de la aplicación. Finalmente, Mule mueve el mensaje a un componente groovy y se muestra la salida en formato JSON.

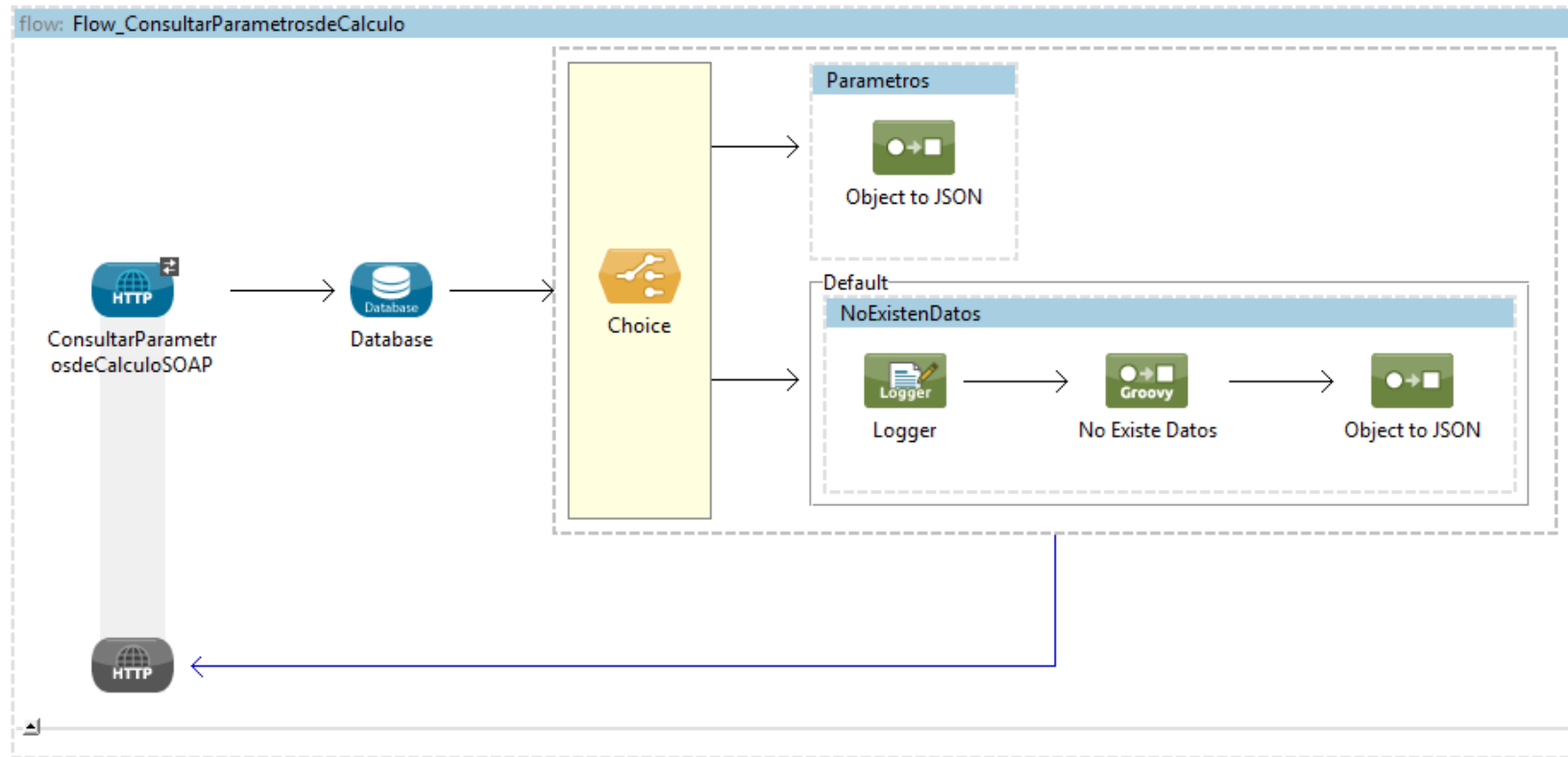


Figura N°44 Flujo con Mule de Consultar Parámetros de Calculo  
Fuente: Autor de la Investigación

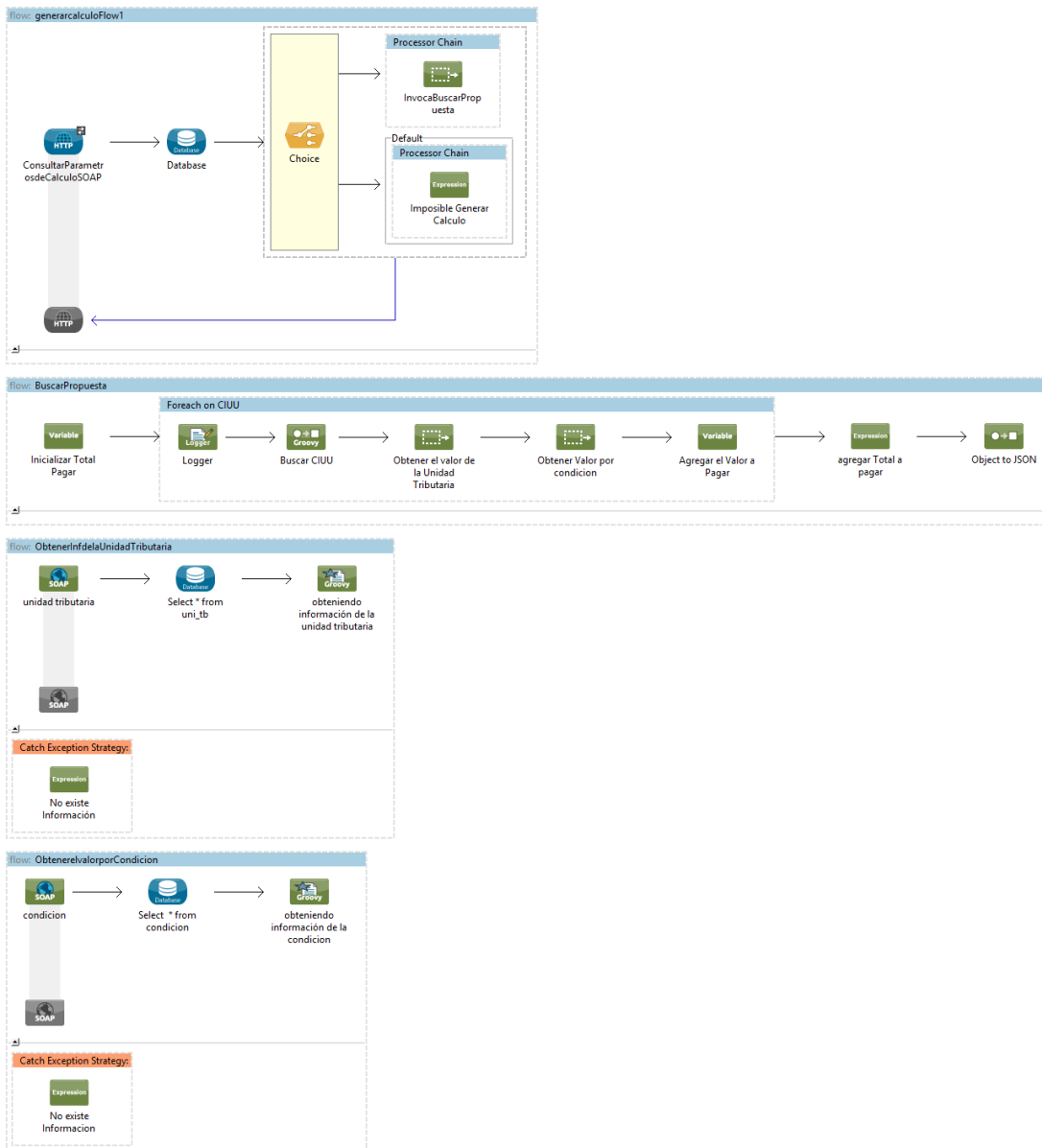


Figura N°45 Flujo con Mule de Generar Cálculo  
Fuente: Autor de la Investigación



*Figura N°46 Flujo con Mule de Generar Factura*  
Fuente: Autor de la Investigación

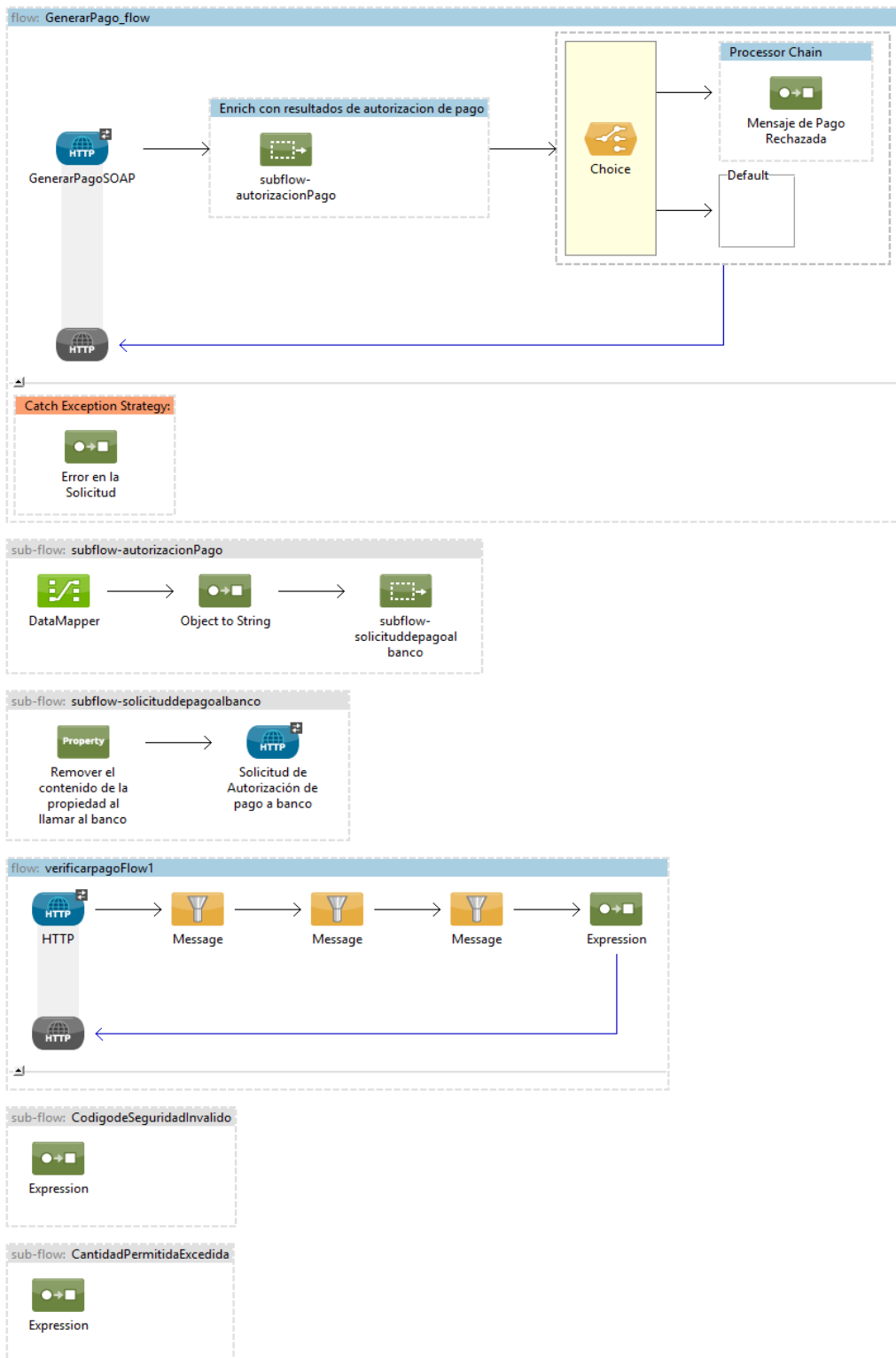


Figura N°47 Flujo con Mule de Generar Pago  
Fuente: Autor de la Investigación

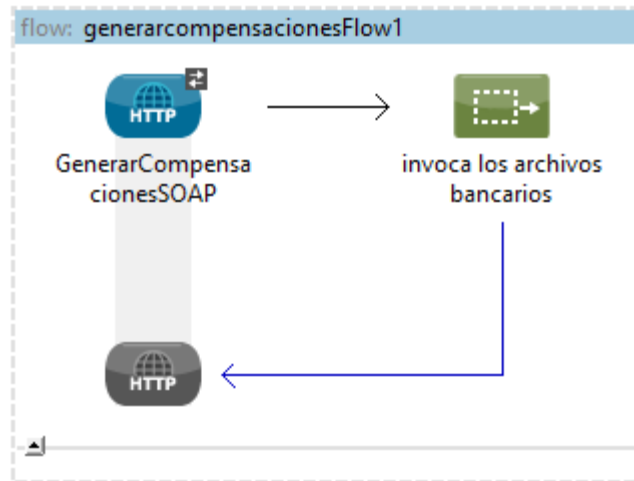
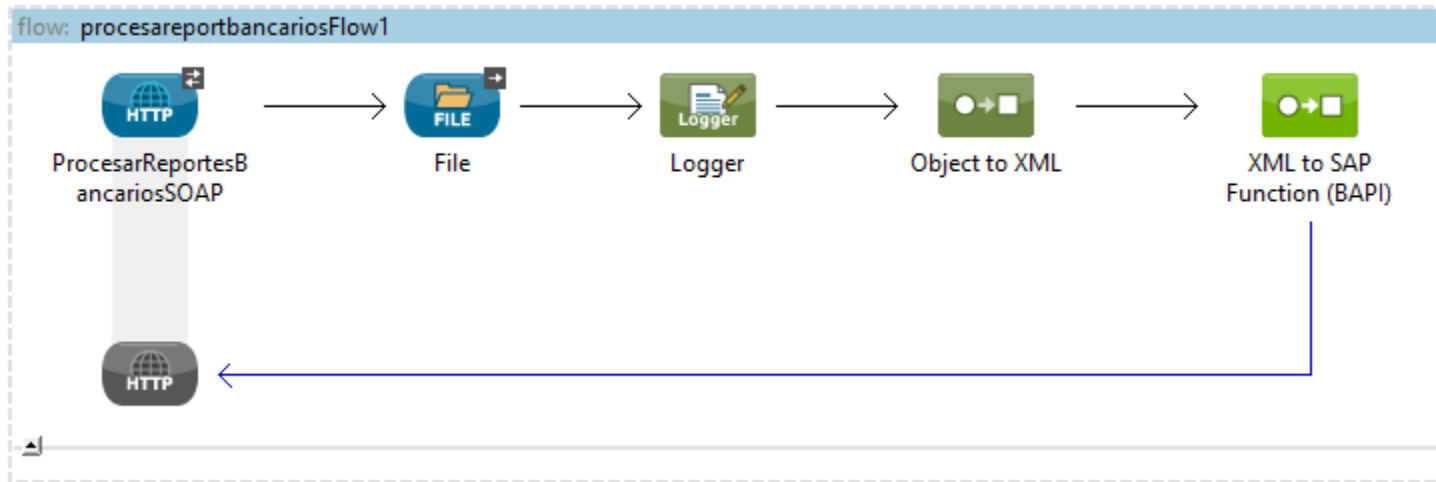


Figura N°48 Flujo con Mule de Generar Compensaciones  
Fuente: Autor de la Investigación



*Figura N°49 Flujo con Mule de Procesar Reportes Bancarios*  
Fuente: Autor de la Investigación

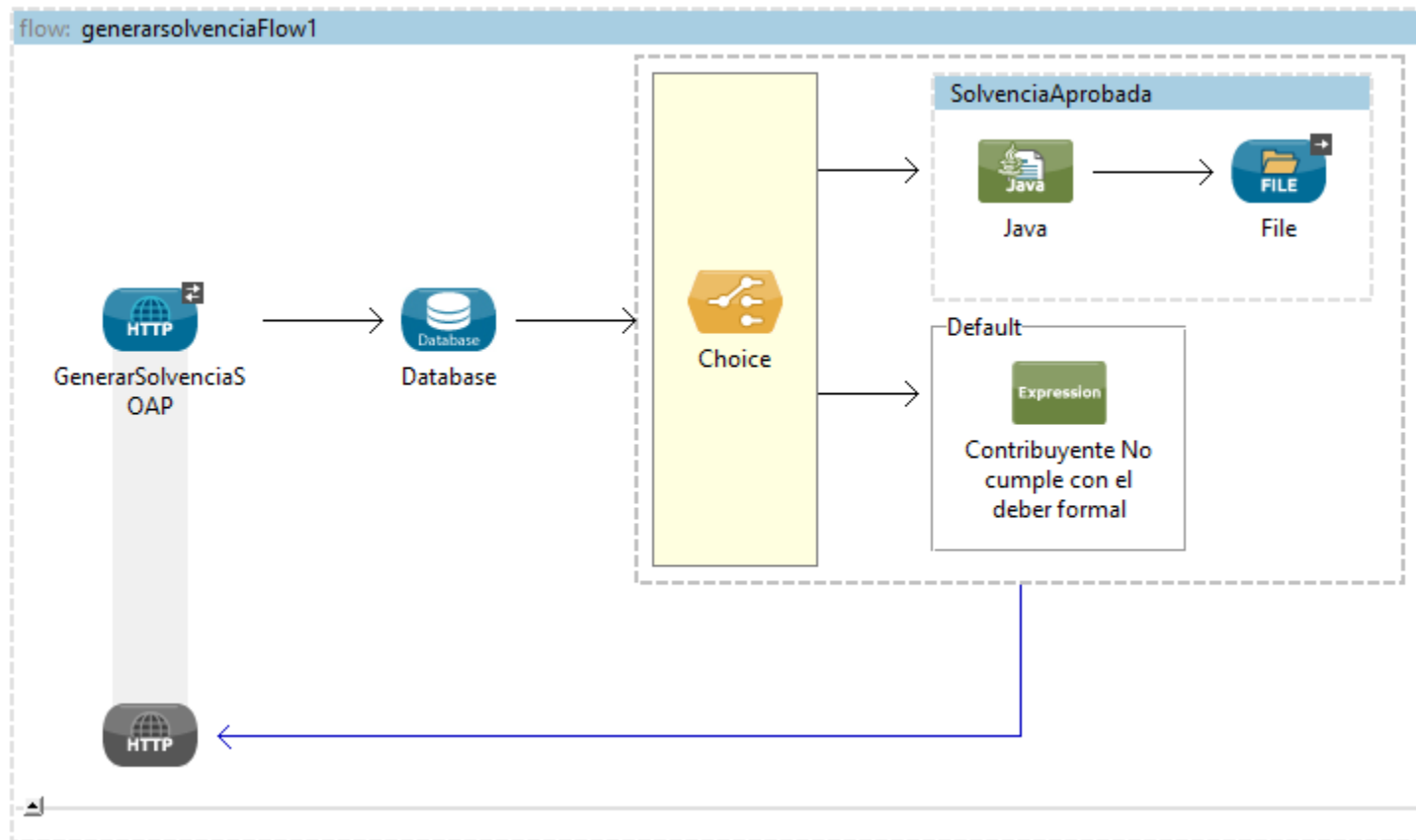


Figura N°50 Flujo con Mule de Generar Solvencia  
 Fuente: Autor de la Investigación



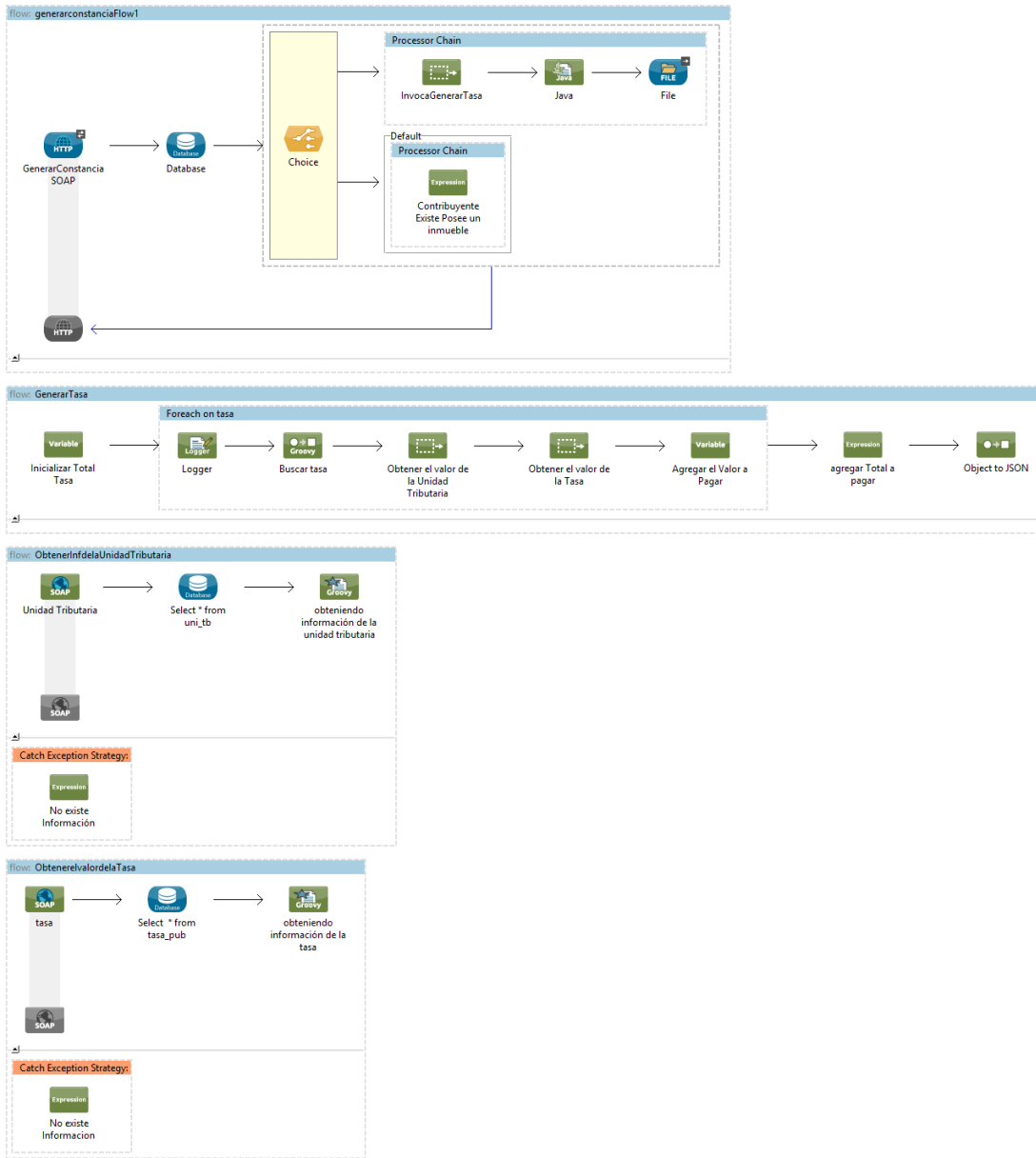


Figura N°51 Flujo con Mule de Generar Constancia  
Fuente: Autor de la Investigación

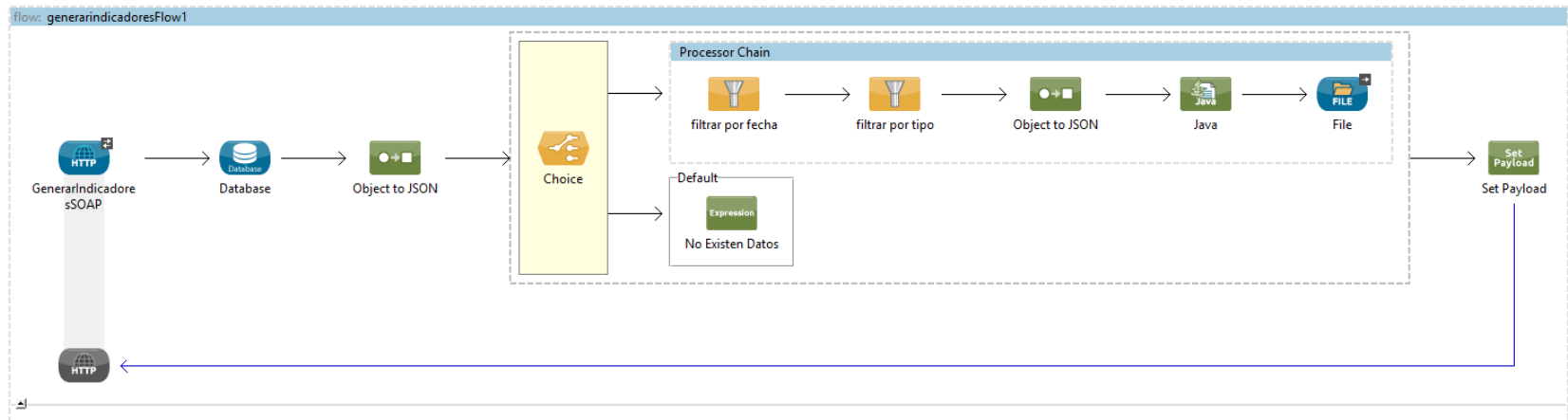


Figura N°52 Flujo con Mule de Generar Indicadores  
 Fuente: Autor de la Investigación

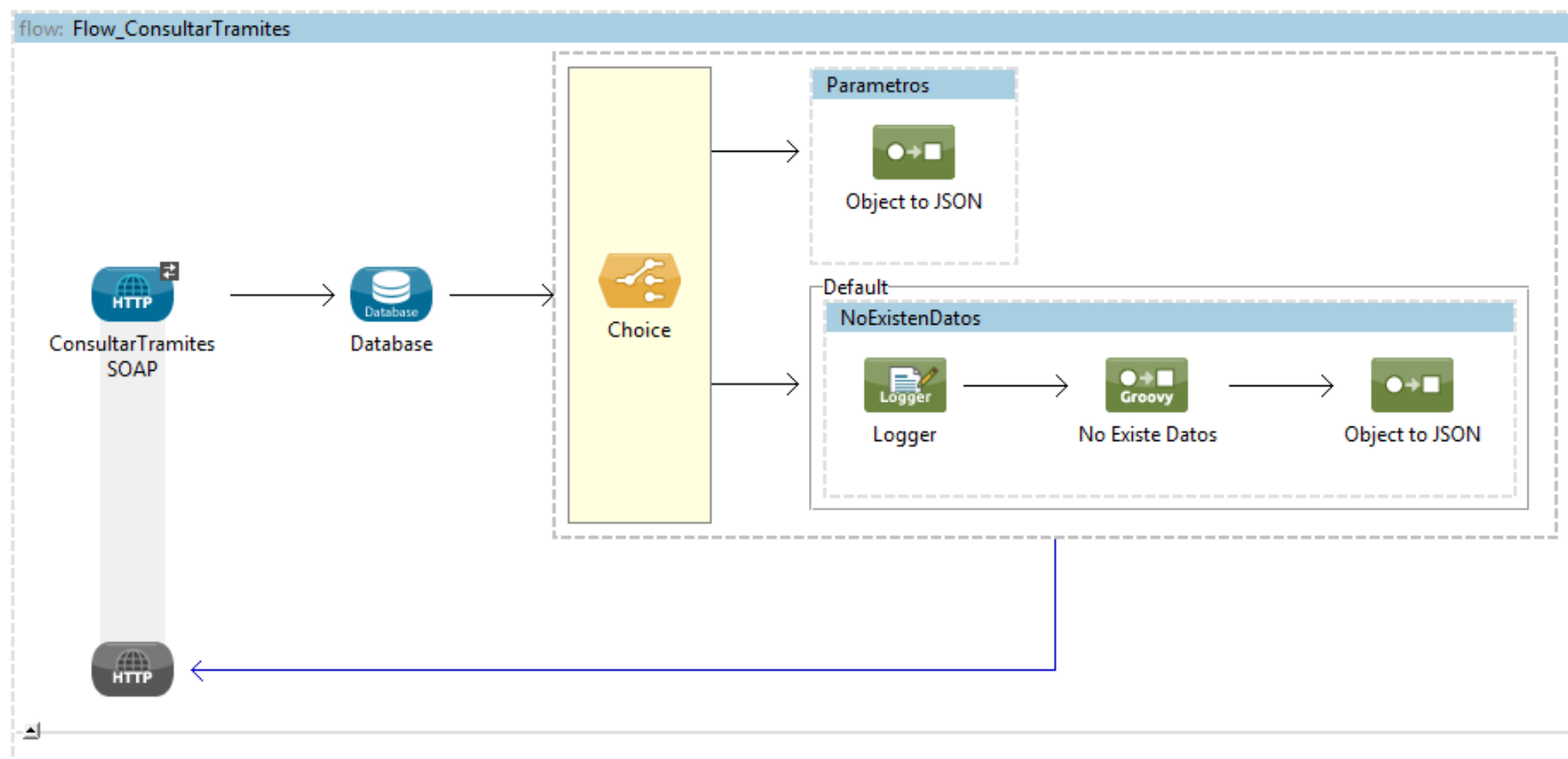


Figura N°53 Flujo con Mule de Consultar Trámite  
Fuente: Autor de la Investigación

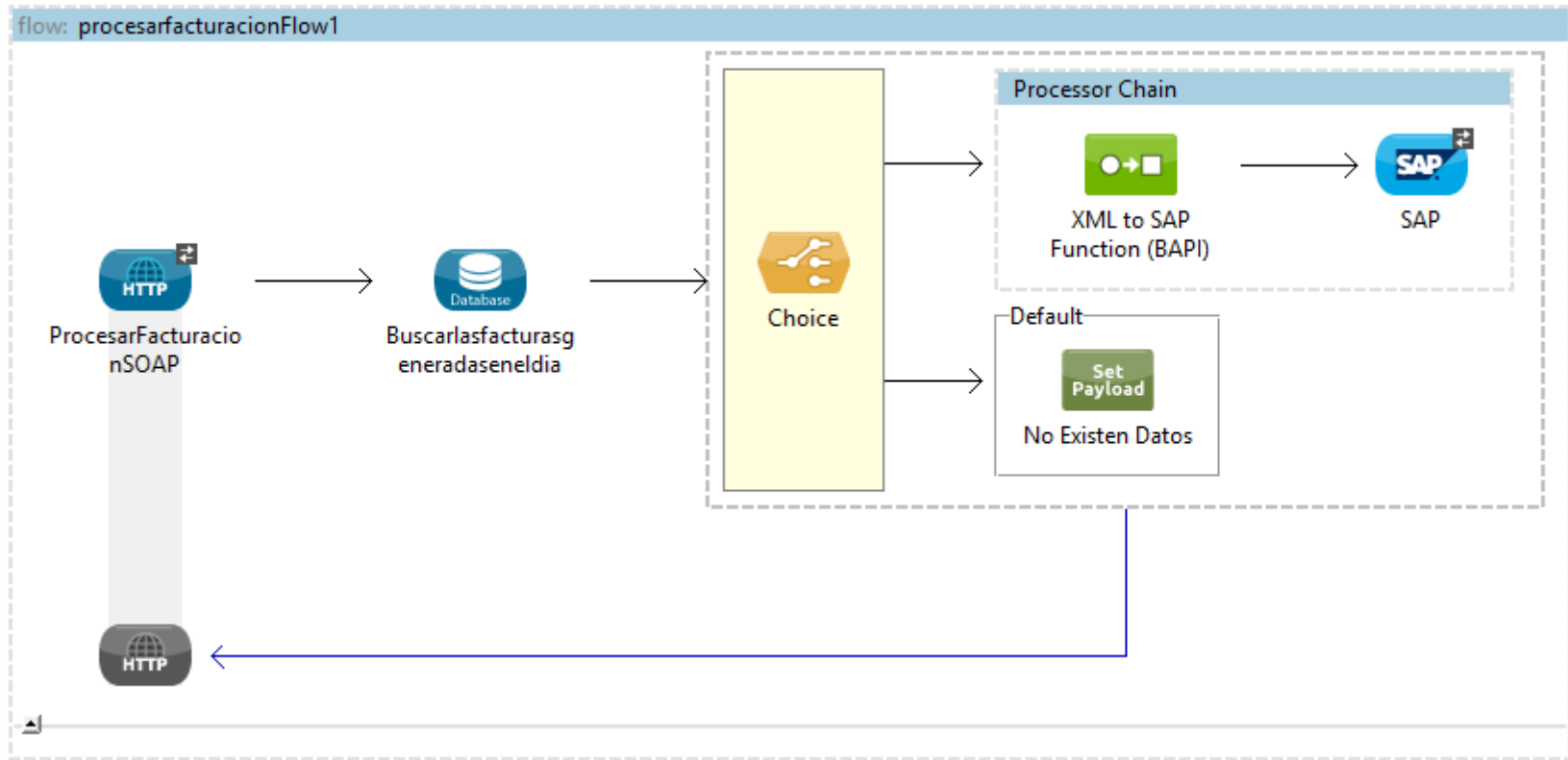
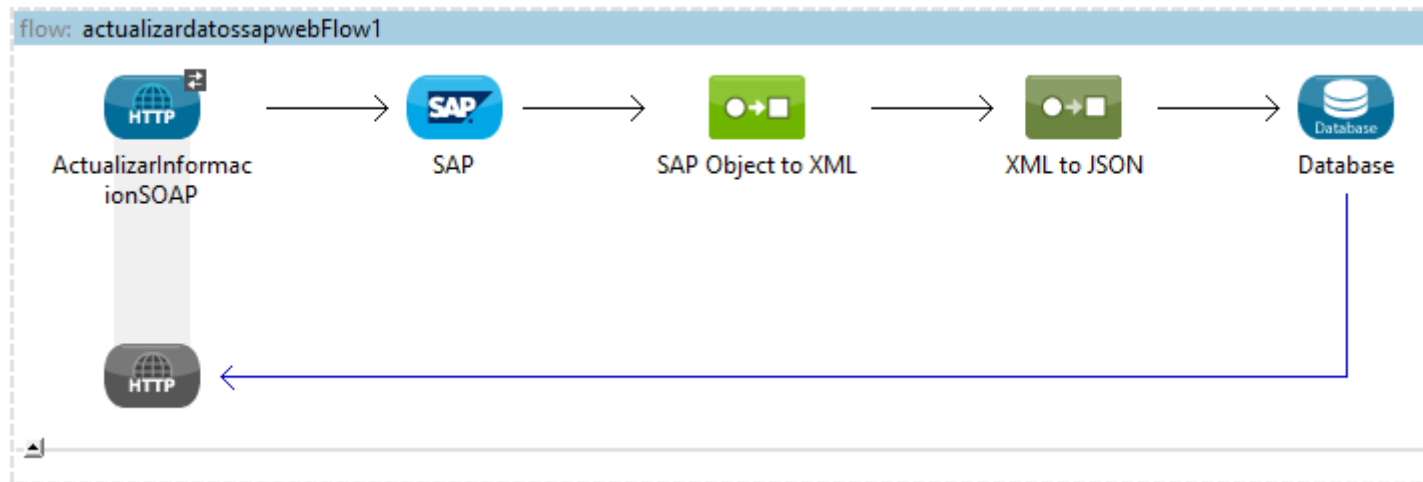


Figura N°54 Flujo con Mule de Procesar Facturación  
Fuente: Autor de la Investigación



*Figura N°55 Flujo con Mule de Actualizar Información*  
Fuente: Autor de la Investigación

Una vez definido los diagramas usando la plataforma MULE representando el comportamiento para el intercambio de mensajes entre las aplicaciones involucradas en el proceso, se procede a establecer la arquitectura de integración de aplicaciones que permite cumplir con el objetivo planteado.

La arquitectura planteada es una arquitectura orientada a servicios (SOA) implementando un ESB que plantea usar las propuestas de implementación de patrones basandose fuertemente en el uso de ESBs como plataforma de base. Dicha arquitectura planteada se ilustra en la figura No. 56.

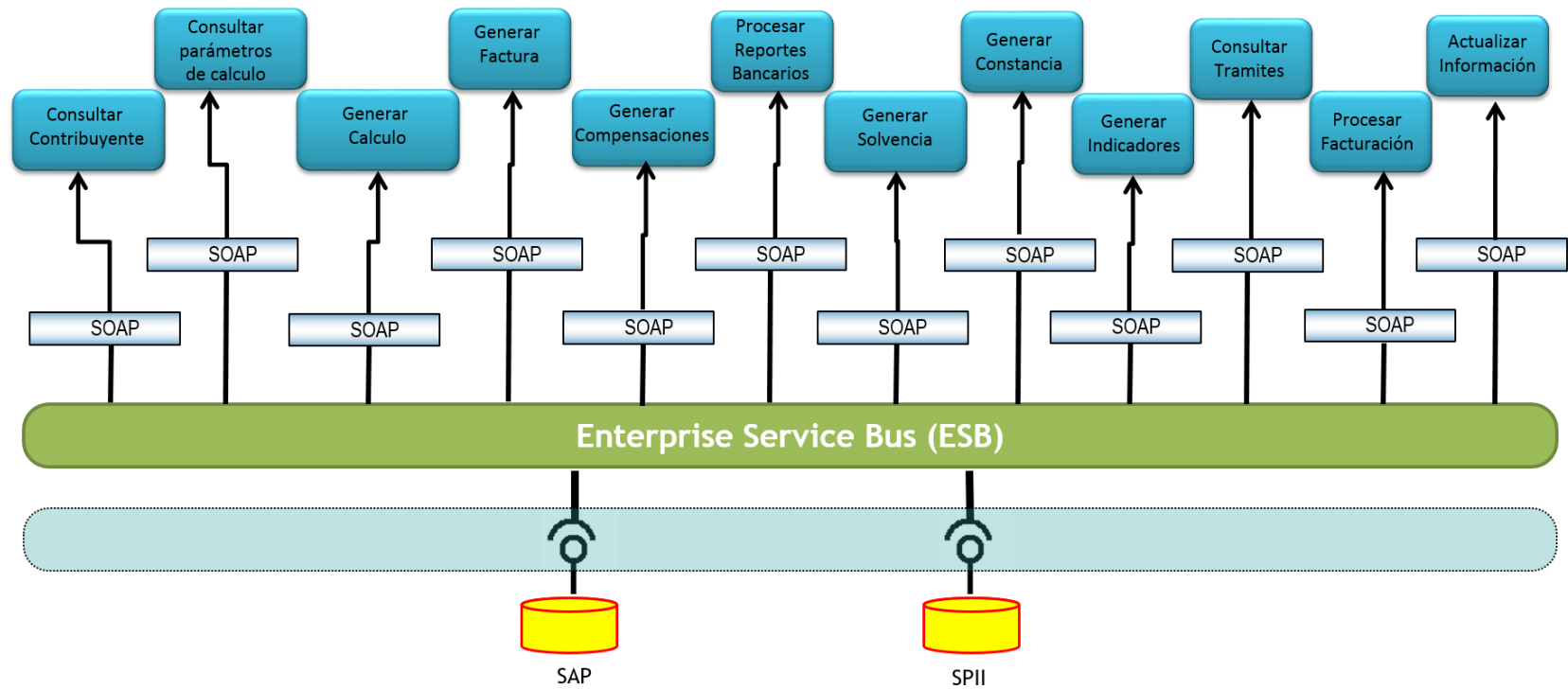


Figura N°56 Arquitectura de Integración de Aplicaciones  
 Fuente: Autor de la Investigación

## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

El presente trabajo de investigación, cuyo desarrollo se fundamenta en el diseño de una arquitectura de integración de aplicaciones con el fin de mejorar los procesos y la toma de decisiones dentro de las organizaciones, haciendo uso de patrones de integración basados en mensajes y un bus de integración usando la plataforma MULE ESB, con el fin de modelar los servicios encontrados en la etapa de análisis de la presente investigación.

Por tal motivo, se establecieron cuatro fases para alcanzar los objetivos propuestos usando metodología XP; donde a través de una investigación bibliográfica se estudiaron los patrones de integración basados en Mensajes, tecnología SOA, los servicios web, las plataformas de integración como Mule ESB, entre otros.

En este sentido, el desarrollo del presente trabajo de investigación, permite establecer una serie de conclusiones a ser descritas a continuación.

- La Elección y la aplicabilidad de la metodología de desarrollo XP en esta investigación resultó satisfactoria en términos como la manera incremental y ordenada de conocer las funcionalidades del negocio, aplicar las herramientas necesarias para lograr la arquitectura deseada y permitir fácilmente adaptabilidad a los cambios.
- El Uso de la Ingeniería de Método como disciplina, propuesta por Torres (2013),



aportó rigurosidad en la obtención de los productos resultantes de los procesos efectuados en cada etapa, añadiendo valor agregado al resultado obtenido. De igual forma se usó la técnica de descubrimiento de la ingeniería de requisitos para lograr visualizar lo que se pretende integrar, realizando un estudio de los objetivos organizacionales utilizando el enfoque Orientado a Objetivos, destacado en la obtención, estructuración, análisis y documentación de requisitos.

- Para apoyar el análisis, descubrimiento y diagramado de los objetivos, se aplicó la herramienta propuesta por Torres (2013) luego de una investigación de otras herramientas existentes. Dicha herramienta usada es el Framework i\* que permitió el modelado de los objetivos, utilizando técnicas con un proceso formal destacado por su riqueza gráfica y adaptabilidad. También se usó UML para el diagramado de casos de uso con su respectivo modelo de descripción, utilizando plantillas que permitieron mostrar la secuencia de eventos que intercambian los actores y el sistema.
- Para el diagramado de los servicios y de las funcionalidades que involucran la integración de aplicaciones y para ver el comportamiento de las mismas se usó Mule ESB con la herramienta Mule Studio, que permitió crear flujos de mensajes y servicios de Mule de forma gráfica y conectar las aplicaciones administrando inteligentemente el enrutamiento de mensajes usando los patrones de mensajería.
- En función de cumplir con lo planteado durante la presente investigación y lograr cubrir una mejor comunicación entre aplicaciones heterogéneas que inicialmente no fueron creadas pensando en integración, se puso en práctica el descubrimiento de un conjunto de funcionalidades traducibles a servicios obtenido luego del diagramado de los objetivos usando i\* y UML. Dichos servicios serán consumidos por las aplicaciones en una arquitectura orientada a servicios (SOA).
- Finalmente se elaboró una arquitectura usando estilo de integración basado en Mensajería que permita integrar aplicaciones intercambiando paquetes de datos de manera frecuente, inmediata, confiable y asincrónicamente usando formatos personalizables. Se utilizó la mensajería por ser considerado un mecanismo maduro para la integración de aplicaciones trabajando conjuntamente con patrones de integración que explican cómo hacer un buen uso de ella y que le brindan agilidad y

eficiencia. Dicha arquitectura hará posible que los futuros sistemas se integren con el sistema actual, con poca o ningún cambio en la infraestructura de mensajería. De esta manera, la funcionalidad de negocio de la aplicación puede ser fácilmente utilizado por otra aplicación.

- Todo lo anterior, se aplicó a un caso de estudio en el que se ejemplificó el modelo planteado obteniendo los resultados esperados.

### **Recomendaciones**

- Aplicar los resultados de la presente investigación y su aporte en el caso de estudio en su contexto operativo.
- La arquitectura de integración resultante para el caso de estudio Alcaldía de Iribarren, pudiese abarcar un ámbito nacional, es decir puede ser usado por las alcaldía que cuenta con sistemas heterogéneos.
- Se aconseja el uso de Ontología que son componentes que permiten hacer explícito y formal el significado de la información contenida en la Web.
- Se aconseja el uso de la Web Semántica, para dar a la información un significado bien definido, que permita a los ordenadores y a las personas trabajar en cooperación. La Web Semántica dotará de estructura al contenido significativo de las páginas web creando un ambiente donde los agentes software puedan navegar por las páginas y puedan llevar a cabo fácilmente sofisticadas tareas en beneficio de los usuarios.

## REFERENCIAS BIBLIOGRÁFICAS

- ✚ Aguilar O. (2010). Integración de aplicaciones empresariales mediante el Enterprise service bus: caso de uso: Programa Nacional de Apoyo Directo a los Más Pobres-PCM.
- ✚ Alba J. (2008) SOA Arquitectura Orientada al Servicio
- ✚ Balado E. (2005). Estrategias para la implantación de nuevas tecnologías en PYMES, obtenga el máximo resultado aplicando las TIC en el ámbito empresarial.
- ✚ Barchini G. et al. (2007). El Rol de las Ontologías en los SI.
- ✚ Bastidas (2008). El Comercio Electrónico y la protección al consumidor en Venezuela.
- ✚ Bazán (2009), un Modelo de integrabilidad con SOA y BPM
- ✚ Bravo C. (2010). Inteligencia Artificial Distribuida como estrategia para la Gerencia Integrada de Producción Industrial
- ✚ Barry D. (2013). Web Services Description Language (WSDL) [Documento en Línea]. Autor. Disponible: [http://www.service-architecture.com/articles/web-services/web\\_services\\_description\\_language\\_wsdl.html](http://www.service-architecture.com/articles/web-services/web_services_description_language_wsdl.html) [Consultada: 2014, Agosto 13].
- ✚ Baixauli B. (s.f.). Análisis de Variabilidad con Modelos de Objetivos
- ✚ Fernández V. (2006) Desarrollo de sistemas de información: Una metodología basada en el modelado
- ✚ Frantz (2008), Integración de Aplicaciones, un Lenguaje específico de dominio para el diseño de soluciones de Integración
- ✚ Gallardo (2003). Sistemas ERP: Importancia de sus aplicaciones en la gestión empresarial.
- ✚ García T. (2011). Opciones de Personalización de su ERP: Integraciones de Datos. [Documento en Línea]. Autor. Disponible: <http://www.erpsoftwareblog.com/2011/12/opciones-de-personalizacion-de-su-erp-integraciones-de-datos/> [Consulta: 2012, Agosto 20].
- ✚ García V. et. al. (2012). Diseño de un método para determinar un conjunto de recomendaciones para realizar la integración de aplicaciones empresariales.

- ✚González L. (2004). ERP: Guía práctica para la selección e implantación.
- ✚Hohpe, G. y Woolf, B. (2003). Enterprise Integrations Patterns. Boston, Estados Unidos: Addison-Wesley.
- ✚Jiménez, C (2011). Analizador de Servicios Web Basados en WSDL 1.1 Para Pruebas Paramétricas
- ✚Laliwala Z et al. (2013). Mule Esb Cookbook
- ✚Lamarca M. (2013). Servicios Web. [Documento en Línea]. Autor. Disponible: [http://www.hipertexto.info/documentos/serv\\_web.htm](http://www.hipertexto.info/documentos/serv_web.htm) [Consulta: 2014, Enero 25].
- ✚López V. (2011) Propuesta de Integración de los sistemas de información para la UNEXPO basados en ROA.
- ✚Manouvrier B. et al. (2010) Application Integration: EAI B2B BPM and SOA
- ✚Martínez E. (2014) Proyecto y viabilidad del negocio o microempresa. ADGD0210
- ✚Moraleda A. (2004). La innovación, clave para la competitividad empresarial.
- ✚MuleSoft (2012). Introduction[Documento en Línea]. Autor. Disponible: <http://www.mulesoft.org/documentation-3.2/display/MULEUSER/Architecture+Guide> [Consultada: 2015, Enero 10].
- ✚Pablos C. et al. (2004). Informática y comunicaciones en la empresa.
- ✚Papazoglou M. (2008) Web Services: Principles & Technology, 1st Edition, © Pearson Education Limited
- ✚Parra E. (2011). Revista Virtual Universidad Católica del Norte”.No. 34, Propuesta de metodología de desarrollo de software para objetos virtuales de aprendizaje - MESOVA-1. [Documento en Línea]. Autor. Disponible: <http://revistavirtual.ucn.edu.co/> [Consulta: 2014, Agosto 21].
- ✚Pérez D. et al (2011). Extreme Programming (XP)
- ✚Ramírez Z. (s.f.). Las ontologías como herramienta en la Gestión del Conocimiento Ontologies as a Knowledge Management Tool
- ✚Ramos A. et al. (2014) Aplicaciones Web Segunda Edición.
- ✚Risi (2007). Enterprise Application Integration. Mucho más que productos de moda. [Documento en Línea]. Autor. Disponible: <http://www.sg.com.mx/content/view/422> [Consulta: 2012, Marzo 12].
- ✚ R.F OLTRA-BADENES et. al (2013). Análisis de requerimientos y diseño de modelo de datos para el desarrollo de una solución ERP vertical adaptada al sector cerámico.

- ✚Romero F. (s.f.). Las Nuevas Tecnologías y la Toma de Decisiones Éticas en la Empresa Privada
- ✚Romero M. et al. (2011). Servicios web SOA utilizando agentes inteligentes para la integración de aplicaciones y servicios de las instituciones de educación superior
- ✚Safelayer.com (2012). Opciones de Personalización de su ERP: Integraciones de Datos. [Documento en Línea]. Autor. Disponible: <http://labs.safelayer.com/es/tecnologia/articulos/361-la-integracion-de-aplicaciones-mediante-enterprise-service-bus> [Consulta: 2012, Diciembre 02].
- ✚Sanz I et. al. (2007), Ontologías en Informática.
- ✚Sevillano F. (2010). redindustria, Red de conocimiento en Informática Industrial, Gestión en Tiempo Real y Agilidad Empresarial. [Documento en Línea]. Autor. Disponible: <http://redindustria.blogspot.com/2010/04/estilos-de-integracion-de-sistemas-i.html> [Consulta: 2012, noviembre 13].
- ✚Sommerville I. (2005). Ingeniería del software.
- ✚Suárez C. (2010). Sistemas Integrados de Gestión (ERP).
- ✚Suárez R. (2010). Tecnologías de la Información y la comunicación.
- ✚Teruel M. (2011). Un Lenguaje para la Especificación de Requisitos en Sistemas Colaborativos
- ✚Torres J. (2013) Modelo de integración de aplicaciones orientado a objetivos e ingeniería de métodos.
- ✚Ulric G. et. al. (2014) Accounting Information Systems
- ✚Universidad Centro Occidental “Lisandro Alvarado” (UCLA). 2002. Manual para la Presentación del Trabajo Conducente al Grado Académico de: Especialización, Maestría, Doctorado. Barquisimeto. Venezuela
- ✚Valera E. (2011) Construcción de un Motor de Búsqueda de Contenidos en Repositorios Confiables, Basado en Crawlers, Enmarcado en una Arquitectura Orientada a Servicio.
- ✚Vázquez C. (2012) Ingeniería de Requisitos Orientado a Objetivos
- ✚Vera M. (2007) Implantación y mantenimiento de aplicaciones ofimáticas y corporativas
- ✚W3C (2012). Guía Breve de Servicios Web[Documento en Línea]. Autor. Disponible: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb> [Consultada: 2012, Diciembre 02].

## **ANEXOS**

## *ANEXOS*

### **Currículo Vitae del Autor**

**Ambar Arelys Mendoza Colmenarez**, portadora de la C.I: N° V-15.960.266, nace en Barquisimeto, edo. Lara el 20 de Octubre de 1982. Realiza estudios de Primaria en la Escuela Lara. Los estudios de secundaria los realiza en el Liceo Lisandro Alvarado obteniendo el título de Bachiller en Ciencias. Posteriormente ingresa a la Universidad Centroccidental “Lisandro Alvarado” de Barquisimeto en el año 2001, para iniciar estudios de pregrado en la carrera de Ingeniería en Informática, egresando en el año 2006 obteniendo el título de Ingeniero en Informática. En el último semestre de la carrera decide realizar pasantías en la Alcaldía de Iribarren y finalizadas las mismas comienza su actividad profesional en el área de Diseño y Desarrollo de Software dentro de la Alcaldía de Iribarren cargo que desempeña en la actualidad. Su trayectoria en dicha organización le ha permitido ganar conocimiento en áreas relacionadas al desarrollo de Software y desea plenamente seguir aprendiendo cada día.