



**Universidad Nacional Mayor de San Marcos**  
Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Escuela Académico Profesional de Ingeniería de Sistemas

**Arquitectura orientada a servicios aplicada al  
factoring electrónico en una entidad financiera no  
bancaria**

**TESINA**

Para optar el Título Profesional de Ingeniero de Sistemas

**AUTORES**

Erik Jorge COZ GONZALES

Carlos Alberto MEDINA ALBÁN

**ASESOR**

Cayo Víctor LEÓN FERNÁNDEZ

Lima, Perú

2007



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Coz, E. & Medina, C. (2007). *Arquitectura orientada a servicios aplicada al factoring electrónico en una entidad financiera no bancaria*. Tesina para optar el título de Ingeniería de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

---

## Contenido

<b>RESUMEN</b> .....	5
<b>ABSTRACT</b> .....	7
<b>INTRODUCCION</b> .....	9
<b>1. PLANTEAMIENTO DEL PROBLEMA</b> .....	15
<b>1.1. Fundamentos del problema</b> .....	15
<b>1.2. Descripción de la realidad</b> .....	17
1.2.1. Aspectos generales de la organización.....	17
1.2.2. Aspectos comerciales de la organización .....	21
1.2.3. La realidad de la arquitectura tecnológica en la empresa .....	34
<b>1.3. Antecedentes del problema</b> .....	40
1.3.1. Caso de Éxito: Caixa Galicia.....	40
1.3.2. Otros Casos de Éxito: .....	50
<b>1.4. Justificación e importancia de la investigación</b> .....	56
1.4.1. Justificación de carácter practico .....	56
1.4.2. Justificación de carácter metodológico .....	57
1.4.3. Justificación de carácter teórico.....	60
1.4.4. Importancia de la investigación.....	60
<b>1.5. Delimitaciones del problema</b> .....	61
1.5.1. Limitaciones temporales .....	61
1.5.2. Limitaciones espaciales.....	62
1.5.3. Otras Limitaciones .....	62
<b>2. FORMULACION DEL PROBLEMA</b> .....	63
<b>2.1. Objetivos</b> .....	63
2.1.1. Objetivos Generales .....	63
2.1.2. Objetivos especificos.....	63
<b>2.2. Definición del problema</b> .....	64
<b>3. MARCO TEORICO CONCEPTUAL</b> .....	67

<b>3.1. Bases Teóricas</b> .....	67
3.1.1. El Financiamiento de la PYMES en el Perú .....	67
3.1.2. El Factoring como fuente de Financiamiento para las empresas .....	74
3.1.3. Aspectos operativos del Factoring .....	87
3.1.4. El Factoring internacional .....	90
3.1.5. Introducción a SOA.....	98
3.1.6. Definición de SOA .....	102
3.1.7. Mitos y Verdades sobre SOA.....	105
3.1.8. Servicios Web.....	106
3.1.9. Tecnologías componentes de SOA.....	108
3.1.10. Relación de SOA con otras tecnologías.....	110
3.1.11. Principios de la orientación a servicios .....	112
3.1.12. Elementos esenciales de SOA.....	114
3.1.13. Tipos de arquitectura SOA.....	119
3.1.14. Capas de la Arquitectura SOA .....	124
3.1.15. Quien define las pautas de SOA.....	127
3.1.16. Ventajas y Desventajas de SOA.....	130
3.1.17. Elementos de SOA que son importantes para su éxito .....	132
3.1.18. Casos de Negocio para SOA.....	134
3.1.19. Barreras a vencer para obtener el éxito de SOA.....	136
3.1.20. Lo que puede lograrse con una Arquitectura SOA.....	136
3.1.21. Como puede beneficiar SOA a los negocios.....	139
3.1.22. Cómo SOA puede afectar los resultados de los negocios.....	140
3.1.23. Modelado de Procesos de Negocio .....	142
3.1.24. Metodología para elaboración de Arquitecturas de Sistemas Microsoft	153
<b>3.2. Definición de Términos Básicos</b> .....	154
3.2.1. Servicio Web:.....	154
3.2.2. UDDI:.....	155

3.2.3.	HTTP: (HyperText Transfer Protocol) .....	155
3.2.4.	WSDL: .....	155
3.2.5.	XML: .....	156
3.2.6.	SOAP: .....	156
3.2.7.	Granularidad (Objetos de Grano Fino y Grueso):.....	156
3.2.8.	.NET: .....	157
3.2.9.	DCOM: .....	157
3.2.10.	Middleware: .....	158
3.2.11.	CORBA:.....	158
<b>4.</b>	<b>METODOLOGIA DE LA INVESTIGACION.....</b>	<b>159</b>
<b>4.1.</b>	<b>Estado del Arte: Análisis de Arquitecturas y Soluciones Existentes ..</b>	<b>159</b>
4.1.1.	Comparativa entre arquitecturas (BenchMarking).....	159
4.1.2.	Soluciones existentes de Interoperabilidad en la Caja Metropolitana ....	187
<b>4.2.</b>	<b>Arquitectura Orientada a Servicios Propuesta .....</b>	<b>195</b>
4.2.1.	Objetivos de la Arquitectura .....	195
4.2.2.	Criterio de elección entre arquitecturas SOA .....	196
4.2.3.	Vista Lógica de la Arquitectura propuesta.....	210
4.2.4.	Capas de la Arquitectura .....	214
4.2.5.	Vista Física de la Arquitectura propuesta.....	233
<b>4.3.</b>	<b>Metodología de Generación de Servicios .....</b>	<b>237</b>
<b>4.4.</b>	<b>Lineamientos del proceso de Generación de Servicios.....</b>	<b>249</b>
<b>4.5.</b>	<b>Aplicación de la Arquitectura.....</b>	<b>257</b>
4.5.1.	Análisis de la propuesta de solución de Factoring Electrónico .....	257
4.5.2.	Procesos de negocio candidatos a convertirse en Servicios Web.....	278
<b>5.</b>	<b>CONCLUSIONES .....</b>	<b>285</b>
<b>6.</b>	<b>RECOMENDACIONES .....</b>	<b>288</b>
<b>7.</b>	<b>REFERENCIAS BIBLIOGRAFICAS .....</b>	<b>289</b>
<b>8.</b>	<b>ANEXOS .....</b>	<b>300</b>

## **RESUMEN**

### **ARQUITECTURA ORIENTADA A SERVICIOS APLICADA AL FACTORING ELECTRÓNICO EN UNA ENTIDAD FINANCIERA NO BANCARIA**

**COZ GONZALES, Erik Jorge**

**MEDINA ALBÁN, Carlos Alberto**

**Septiembre – 2007**

**Asesor: LEON FERNANDEZ, Cayo**

**Grado: Tesina para optar por el grado de Ingeniero de Sistemas e  
Informática**

---

La presente tesina aplicada, se refiere a la elaboración de una propuesta de arquitectura tecnológica empresarial, aplicada a una entidad financiera en particular, basándose en el análisis de la problemática actual de la

organización, y particularmente en el concepto de servicios Web para proveer una herramienta tecnológica capaz de brindar solución a sus necesidades de negocio. Como caso de aplicación se estudia el producto financiero denominado Factoring. La tesina aborda los principales modelos de arquitecturas orientadas a servicios disponibles y se ocupa adicionalmente de la interoperabilidad de las aplicaciones informáticas existentes. La metodología en la que se enmarca el desarrollo de la arquitectura propuesta es la metodología Microsoft para arquitecturas de aplicaciones empresariales.

Asimismo se propone una metodología para la generación de servicios, que se encuentra basada en un enfoque iterativo, esto con la finalidad de contar con un adecuado marco para la automatización de procesos de negocio y una serie de lineamientos generales a tomarse en cuenta durante su construcción y que son factibles de aplicarse en otros proyectos de comportamiento similar.

**Palabras clave:**

Arquitectura Orientada a Servicios

Servicio Web

Interoperabilidad

Factoring electrónico

**ABSTRACT**

**ARQUITECTURA ORIENTADA A SERVICIOS APLICADA AL FACTORING  
ELECTRÓNICO EN UNA ENTIDAD FINANCIERA NO BANCARIA**

**COZ GONZALES, Erik Jorge**

**MEDINA ALBÁN, Carlos Alberto**

**September – 2007**

**Adviser: LEON FERNANDEZ, Cayo**

**Degree: Tesina para optar por el grado de Ingeniero de Sistemas e  
Informática**

---

The present applied tesina, talks about the elaboration of a proposal of enterprise technological architecture, applied to a financial organization in individual, being based on the problematic analysis of the present one of the organization, and particularly in the concept of Web Services, to provide a

technological tool able to offer solution to its necessities of business. As case of application we studies the financial product denominated Factoring. The tesina approaches the main models of Services-oriented architectures available and it takes care additionally of the interoperability of the existing computer science applications. The methodology in which the development of the propose architecture is framed is the Microsoft methodology for architectures of enterprise applications.

Also a methodology for the generation of Services is sets out, that is cradle in an iterative approach, this with the purpose of counting on a suitable frame for the automatization of business processes and a series of guidelines to be taken into account during its construction and that are feasible to be applied in other projects of similar behavior.

**Keywords:**

Services – Oriented Architecture

Web Services

interoperability

e - Factoring

## **INTRODUCCION**

Cada vez más las organizaciones dependen de su infraestructura de tecnologías de información para alcanzar sus objetivos. Pero en un entorno competitivo como el actual, aprovechar las oportunidades de negocio exige moverse cada vez con mayor rapidez. Sin embargo, con frecuencia las Tecnologías de Información no permiten estas respuestas rápidas o disponen de la flexibilidad necesaria para competir de forma efectiva. Un alto porcentaje de las ineficiencias organizativas tienen un mismo origen: el predominio de procesos manuales con un nivel de error elevado, sistemas ineficaces para compartir la información en el núcleo de la organización; la incapacidad de hacer un correcto seguimiento de los procesos de negocio de principio a fin, el cumplimiento con las normativas legales aplicables, etc. Todo esto obliga a manejar grandes cantidades de información y en formatos complicados de utilizar. Allí donde se produce cualquiera de estas situaciones, el impacto sobre la productividad de la empresa es negativo, y se pone en riesgo la capacidad de crecimiento y competencia de la misma.

En la raíz de todas estas deficiencias está la información. No es un problema de escasez de información –de hecho, la información es el activo de más rápido crecimiento dentro de las organizaciones en la actualidad- sino de

la imposibilidad de presentar la información de forma sencilla y útil a los usuarios y directivos de una manera coherente y sistemática. En última instancia, esto se debe a que las aplicaciones de línea de negocio y otras aplicaciones antiguas normalmente no pueden aportar una visión general de los procesos de negocio cuando éstos abarcan varias áreas funcionales a su paso. Para lograr que la información se mueva a través de sistemas distintos (y generalmente incompatibles), dentro de las fronteras de la organización o a través de ellas, siempre es necesaria la intervención humana. Hasta ahora esta intervención se ha hecho de dos formas. La primera es la de volver a introducir a mano los datos entregados por un sistema en otro distinto e incompatible. La segunda ha sido programar una interfaz especialmente diseñada para permitir la transferencia de información entre dos aplicaciones incompatibles.

Ambas soluciones son costosas e ineficientes debido en el primer caso a que se trata de un esfuerzo de duplicación de datos donde el error humano es frecuente, y en el segundo, porque conseguir un buen nivel de interoperabilidad e integración no solo es muy difícil a la hora de desarrollar una solución, sino porque su mantenimiento posterior es una verdadera pesadilla.

Cualquier cambio en cualquier aplicación puede hacer fallar la interfaz de transferencia impidiendo la comunicación entre ellas. Lo que se necesita es una herramienta basada en estándares para integrar sistemas y aplicaciones heterogéneos sobre una serie de plataformas y protocolos de comunicación heterogéneos, así como una metodología bien establecida para lograr el nivel óptimo de integración, de manera que la nueva arquitectura facilite –en lugar de impedir- los cambios posteriores que puedan surgir como respuesta a la evolución en las necesidades de la empresa.

Es así que esta problemática se presenta en diferentes tipos de organizaciones de diversa índole, una de ellas es el ámbito financiero. Hoy en día se evidencia un incremento en la demanda por parte de la población de diversos servicios y productos de este tipo. El prometedor momento que atraviesa la economía nacional influye en las empresas de nuestro entorno que cada día requieren responder con mayor prontitud a los requerimientos de sus clientes y por supuesto a la creciente competencia, ofreciendo nuevos productos y nuevas formas de interacción, empleando para ello diversos mecanismos (banca por Internet, servicios móviles, etc.). Pero, ¿están las Tecnologías de información en nuestro país - y sobre todo en nuestras instituciones financieras, una de las cuales es sujeto de estudio en la presente tesina – preparadas para el desafío que implica un menor tiempo de reacción, para aprovechar las nuevas oportunidades de negocio, integrando sus sistemas y su plataforma tecnológica de manera flexible y a tiempo?

En esta tesina nos ocupamos de una institución financiera específica, es el caso de la Caja Metropolitana de Lima, la cual presenta una problemática muy similar a la descrita pero bajo ciertas particularidades propias de la empresa. El objetivo de esta investigación es proponer una arquitectura que le permita a esta organización, integrar sus aplicaciones existentes y poder responder con rapidez a los cambios del mercado y poder soportar sus requerimientos de negocio a través de una nueva, esto mediante la aplicación de la denominada Arquitectura Orientada a Servicios, ejemplificado el impacto de esta propuesta con la construcción el modelado de proceso de negocio existente en la organización como el Factoring electrónico.

El capítulo 1, se centra en el planteamiento del problema, a través de este capítulo presentamos los fundamentos que dan sustento al problema, se menciona la necesidad de una arquitectura orientada a servicios por medio de una sucinta comparación con modelos existentes, los aspectos de descripción de la realidad de la organización y damos una visión general a manera de marco referencial para conocer la organización en la que se aplica este estudio, que es la Caja Metropolitana de Lima, aquí mostramos a través de una serie de información estadística el comportamiento y los principales indicadores de esta organización en su parte comercial y hacemos una descripción de la realidad de su arquitectura tecnología actual. Si bien obtener información sobre los detalles de las implementaciones de arquitecturas orientadas a servicios resulta complicado ya que resulta ser un material la mayoría de veces confidencial de las empresas y sobre todo al no contar con casos aplicados en nuestro país, brindamos algunos antecedentes del problema por medio de una serie de casos de implementaciones orientados a servicios en el exterior. En el caso de instituciones similares a la de nuestra tesina encontramos el de la Caja Galicia de España, una institución financiera que por medio de la orientación a servicios consiguió implementar una solución que integra sus sistemas para dar soporte a una serie de procesos de negocio para un fin específico como es una extranet hipotecaria, así como otros casos de éxito. Por último se presentan las limitaciones de la investigación por medio de la delimitación del problema.

En el capítulo 2, presentamos la formulación del problema por medio de los objetivos, tanto generales como específicos de esta investigación, la cual se centra en una propuesta de arquitectura tecnológica orientada a servicios que soporten el desarrollo e implementación de soluciones de negocio como el

Factoring electrónico en la Caja Metropolitana. Se define el problema, describiéndolo en el ámbito de la organización y señalando las necesidades que la arquitectura actual no permite satisfacer tanto en forma adecuada como en los tiempos adecuados.

En el capítulo 3, se toca el marco teórico conceptual a través del cual presentamos los diferentes conceptos que serán utilizados a lo largo de nuestra investigación, es importante conocer detalladamente estos conceptos para así tener una clara comprensión del problema. Presentamos el marco tanto en el aspecto tecnológico como en el aspecto de negocios, - el cual se apoyara en la arquitectura propuesta -. En la parte de negocio se toca la problemática de las pequeñas y mediana empresa en nuestro país, las fuentes de financiamiento a las que tienen acceso en la actualidad. Describimos la teoría relacionada al Factoring como alternativa de financiamiento. En la parte tecnológica - la cual es la más extensa – presentamos el marco teórico de la Arquitectura orientada a servicios, y los conceptos de las tecnologías relacionadas a esta arquitectura como son los Servicios Web, los protocolos de comunicación, el lenguaje de descripción de interfaces, el lenguaje XML, los principios de la orientación a servicios y sus elementos esenciales. También se mencionan los cinco tipos de arquitectura SOA, dependiendo estas de la problemática a la que atacan en las organizaciones. Se describen las ventajas y desventajas de una arquitectura SOA a grandes rasgos así como la descripción genérica de las capas de una arquitectura orientada a servicios.

En el capítulo 4, se presenta el cuerpo principal de la tesina , es decir la investigación propiamente dicha para lo cual se ha preparado un análisis propio de los tipos más conocidos de arquitecturas de tecnología de información, a

saber los de dos capas, de n capas basados en componentes y las arquitectura orientada a servicios. Presentando una comparativa entre sus indicadores principales. También se analiza como estado del arte las soluciones existentes en la organización y que atacan el problema de la interoperabilidad entre empresas que desean establecer procesos y servicios con la Caja Metropolitana de Lima. Asimismo se desarrolla la metodología para la confección de arquitecturas, sobre la cual se basa la investigación. En este caso partimos de la aplicación de la metodología Microsoft que involucra vistas de arquitectura tanto Conceptual, Lógica y Física, de la cual se tomara los elementos necesarios para nuestro caso de estudio. Se presenta la propuesta de arquitectura orientada a servicios para este caso particular, describiendo las capas que lo componen. Asimismo y tomando siempre la orientación de servicios como tronco principal se presenta una metodología para la generación de servicios en la organización y los lineamientos generales que deben tenerse en cuenta al momento de su construcción. Por último se muestra el caso práctico de aplicación, el cual trata de la identificación y modelado de los procesos de negocios para el caso del Factoring electrónico que se brinda como producto financiero en la empresa, bajo la metodología BAM (Modelamiento de Actividades de Negocio).

En el capítulo 5 y 6, se presentan las conclusiones y recomendaciones respectivamente de nuestra investigación.

## **1. PLANTEAMIENTO DEL PROBLEMA**

### **1.1. Fundamentos del problema**

Desde siempre, desarrollar aplicaciones empresariales no ha sido una tarea fácil considerando que los avances de la tecnología han ido simplificando el manejo de varios aspectos de este problema. Sin embargo, los desarrolladores de este tipo de aplicaciones continúan enfrentándose a desafíos tales como datos complejos; la mayor parte del tiempo los requerimientos no son explícitos, con usuarios simultáneos y múltiples, los requerimientos del negocio cambian con frecuencia, plataformas heterogéneas, e interdependencias complejas entre aplicaciones distribuidas. Además, el hecho de la poca difusión de implementaciones basadas en los pocos estándares existentes de la industria para el desarrollo de este tipo de aplicaciones ha llevado a muchas organizaciones a desarrollar sus propios modelos generando así un alto costo de desarrollo y mantenimiento de las aplicaciones.

La arquitectura orientada a servicios surge de la necesidad de establecer un sistema de intercambio de datos normalizado entre aplicaciones. Es decir de la necesidad de integrar datos y procesos entre distintas aplicaciones que se están programadas con lenguajes diversos (Cobol, C, Java, VB) que se ejecutan en equipos diferentes (Mainframe, servidor, PC) con

sistemas operativos distintos (UNIX, AIX, Windows, MacOS). A este reto hay que añadir una característica más, el aprovechamiento de Internet para conseguir dicha integración.

Existen varios protocolos orientados a permitir la computación distribuida entre distintos equipos, por ejemplo encontramos la arquitectura CORBA (Common Object Request Broker Architecture, *Arquitectura común de intermediarios en peticiones a objetos*), el modelo DCOM (Distributed Component Object Model, *Modelo de Objetos de Componentes Distribuidos*) y el modelo RMI (Remote Method Invocation, *Método de Invocación Remota*). Entre estos tres ejemplos es CORBA el que se presenta más flexible, puesto que permite la comunicación entre aplicaciones programadas en lenguajes distintos y ejecutados en plataformas diferentes, mientras que el modelo DCOM es la solución de integración propietaria de Microsoft y sólo adaptable a sus lenguajes y sistemas operativos y RMI por su parte exige que los componentes estén programados en Java, es decir que también es un modelo basado en tecnología propietaria. Sin embargo existen varios problemas que son aplicables a los tres, como son la complejidad que reviste su implementación y, lo que no es menos importante, su poco o nula capacidad de usar el protocolo http para sus comunicaciones, lo cual descarta el aprovechamiento de Internet.

“Uno de los niveles en los que hoy en día, no se ha solucionado el problema de la heterogeneidad es el del *middleware*, el cual se presenta como un método de comunicación entre procesos distribuidos. Inicialmente los sistemas *middleware* se desarrollaron para resolver el problema de heterogeneidad entre comunicación de aplicaciones, a nivel de lenguajes y plataformas y en este sentido, tuvieron éxito. No obstante la proliferación de

este tipo de sistemas ha transferido este problema al de la heterogeneidad entre los propios *middleware*. De tal forma que, dos *middleware* diferentes trabajando por ejemplo en CORBA y Java RMI no pueden interoperar”<sup>1</sup>.

## **1.2. Descripción de la realidad**

### 1.2.1. Aspectos generales de la organización

La Caja Municipal de Crédito Popular de Lima (CMCPL, que de ahora en adelante llamamos Caja Metropolitana), es una entidad financiera que pertenece al Sistema de Cajas Municipales del Perú, la cual está regulada por la Federación Peruana de Cajas Municipales del Perú (FEPCMAC) y supervisada por la Superintendencia de Banca y Seguros (SBS).

La Caja Municipal de Crédito Popular de Lima se creó por Ley N° 10769 del 20 de enero de 1947 y fue reglamentada mediante Acuerdo de Concejo N° 0062 de 1971, de propiedad de la Municipalidad Metropolitana de Lima, con autonomía económica y financiera, e inició sus operaciones el 8 de septiembre de 1949.

La Caja Municipal de Crédito Popular de Lima es una sociedad anónima integrada al sistema financiero nacional que se rige por su Estatuto, la Ley N° 26702, Ley General del Sistema Financiero y del Sistema de Seguros y Orgánica de la Superintendencia de Banca y Seguros, Ley General, la Ley N° 10769, la Ley General de Sociedades y las disposiciones establecidas por la

---

<sup>1</sup> <http://vido.escet.urjc.es/Seminario06.html>

Superintendencia de Banca y Seguros, el Banco Central de Reserva del Perú y la Contraloría General de la República.

La Caja Metropolitana es una empresa de operaciones múltiples definida en el artículo 282° de la Ley General como una empresa especializada en otorgar créditos pignoratícios al público en general, encontrándose también facultada para realizar operaciones activas y pasivas con los concejos provinciales y distritales y con las empresas municipales dependientes de los primeros, así como para brindar servicios bancarios a dichos concejos y empresas.

La Caja Metropolitana tiene por finalidad fomentar el ahorro, desarrollar el crédito de consumo en las modalidades de préstamos pignoratícios y personales, concediendo adicionalmente créditos a la micro y pequeña empresa e hipotecarios para financiación de vivienda.

Adicionalmente, puede realizar las siguientes operaciones:

- Captar ahorros del público.
- Operar con moneda extranjera.
- Efectuar cobros, pagos y transferencias de fondos.
- Emitir giros contra sus propias oficinas y/o bancos corresponsales.
- Actuar como fiduciarios en fideicomisos.
- Efectuar préstamos en general, con o sin garantía específica.
- Emitir Cartas Fianzas.

Para dar una visión general de la organización, a continuación describimos en grandes rasgos el organigrama vigente según lo que se observa en la figura 1

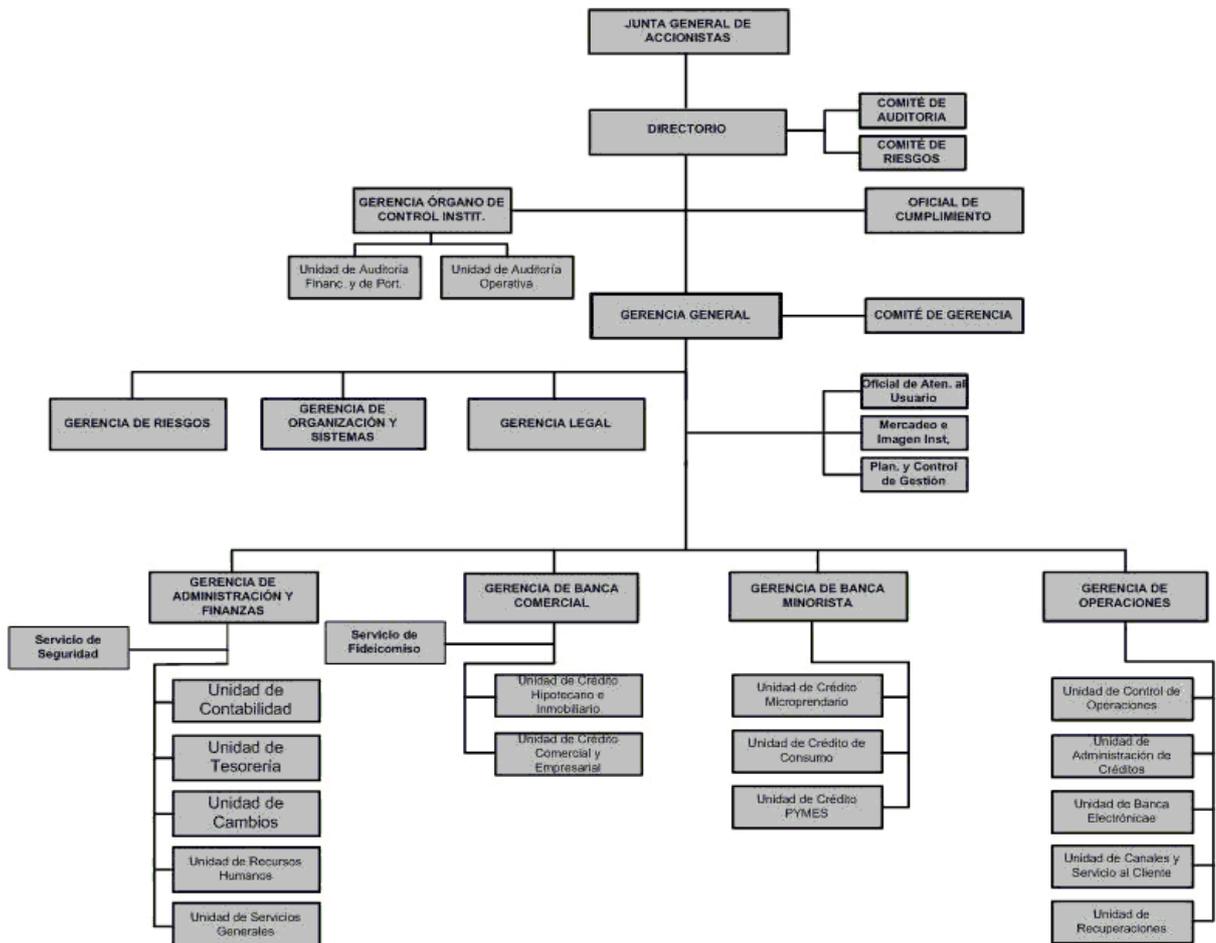


Figura 1. Organigrama actual de la CMCPL

- **Junta General de Accionistas**

Órgano supremo de la Institución, conformada por todos los Accionistas. Sus acuerdos obligan a la Sociedad y a todos sus Accionistas aunque no hayan concurrido a las Sesiones de la Junta General.

- **Directorio**

Órgano colegiado que fija los objetivos, las estrategias, el Plan Anual y las Políticas de la Caja Metropolitana, define los riesgos a asumir y encomienda a la Gerencia General la gestión diaria de la Caja Metropolitana.

- **Presidencia**

Es el más alto cargo directivo de la Caja Metropolitana, cuya función principal es representar al Directorio en actos públicos y privados y hacer cumplir las disposiciones emanadas por el Directorio. Representa a la Caja Metropolitana ante la Municipalidad Metropolitana de Lima.

- **Gerencia General**

Implementar las estrategias aprobadas por el Directorio, gestionando las actividades diarias de la Caja Metropolitana, de la misma manera supervisa, controla y decide sobre su administración, sus actos, sus operaciones y su funcionamiento y controles internos en general.

- **Gerencia del Órgano de Control Institucional**

Encargado de ejecutar el Control Gubernamental posterior interno y externo, cuya misión es promover la correcta y transparente gestión de los recursos y bienes de la Caja Metropolitana, mediante la ejecución de acciones de control.

- **Oficial de Cumplimiento**

Vigilar el cumplimiento de lavado de activos y/o del financiamiento de terrorismo si lo hubiere, con la finalidad de determinar las transacciones que podrían ser calificadas como sospechosas.

- **Gerencia de Riesgos**

Identificar, medir, controlar, administrar y reportar los riesgos operativos en los sistemas, procesos, productos, servicios o funciones que la Caja Metropolitana puede enfrentar en el desarrollo de sus actividades.

- **Gerencia de Organización y Sistemas**

Diseñar, desarrollar, implementar y optimizar normas, políticas, procedimientos, funciones, comunicación y tecnología, facilitando el proceso y cumplimiento de funciones de las distintas Unidades Organizativas de la Caja Metropolitana.

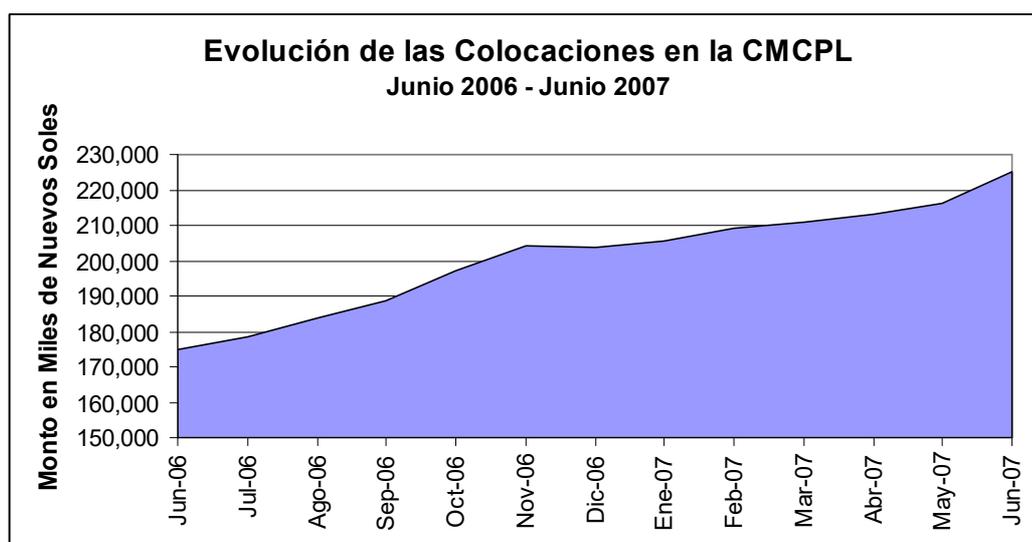
- **Gerencia Legal**

Ejercer la protección y la defensa legal de los derechos e intereses de la Caja Metropolitana brindando asesoría legal a las distintas Unidades Organizativas de la Caja Metropolitana a fin de reducir posibles contingencias originadas en el desarrollo de sus actividades.

#### 1.2.2. Aspectos comerciales de la organización

Para graficar a grandes rasgos los principales indicadores de esta entidad financiera podemos decir que en lo que va del último año, se ha tenido un crecimiento de sus *captaciones* del 12.22% con un saldo total a de S/.

100,440,000 al 30 de junio de 2007, contando con un incremento en el grado de confianza de sus clientes, debido a lo atractivo de sus tasas pasivas. Igualmente hay que resaltar que esta institución tiene una participación en el mercado de las cajas municipales a nivel nacional del 3.04%, llegando a ocupar el noveno lugar en el ranking de captaciones. Podemos observar la evolución reciente de las colocaciones en los últimos doce meses en la Figura 2.



*Figura 2. Evolución de las colocaciones en la CMCPPL en el periodo de Junio 2006 – Junio 2007*

También cabe indicar que en el mismo periodo, en la cartera de *colocaciones* se tiene un incremento del 28.80% con un saldo de colocaciones de S/. 225,256,000<sup>2</sup> destacando los créditos hipotecarios con un 31.44% de participación del total de la cartera de créditos de la empresa, los que son orientados a los sectores medios y bajos que no tienen fácil acceso a la vivienda y que se vienen otorgando con financiamiento en convenio con

<sup>2</sup> <http://www.sbs.gob.pe/PortalSBS/Boletin/BoletinCM/default.htm>

COFIDE, también tenemos los créditos de consumo o personales con una participación del 30.09% los cuales son orientados últimamente a la conversión de vehículos motorizados a gas. Respecto a su ubicación en colocaciones en el ranking financiero de las cajas municipales, la Caja Metropolitana ocupa el séptimo lugar con 5.97% de participación en el mercado.

### Ranking de Créditos, Depósitos y Patrimonio

(Al 30 de Junio de 2007)

(En miles de nuevos soles)

#### Créditos Directos

	<b>Empresas</b>	<b>Monto</b>	<b>Participación</b>	<b>Porcentaje</b>
			<b>( % )</b>	<b>Acumulado</b>
1	CMAC Trujillo	673 605	17.86	17.86
2	CMAC Arequipa	653 353	17.32	35.19
3	CMAC Piura	563 516	14.94	50.13
4	CMAC Sullana	346 136	9.18	59.31
5	CMAC Cusco	304 061	8.06	67.37
6	CMAC Huancayo	274 024	7.27	74.64
<b>7</b>	<b>CMCP Lima</b>	<b>225 256</b>	<b>5.97</b>	<b>80.61</b>
8	CMAC Tacna	196 466	5.21	85.82
9	CMAC Ica	158 387	4.20	90.02
10	CMAC Paita	122 705	3.25	93.27
11	CMAC Maynas	117 969	3.13	96.40

12	CMAC Del Santa	111 091	2.95	99.35
13	CMAC Pisco	24 649	0.65	100.00

---

### Depósitos Totales

	<b>Empresas</b>	Monto	Participación ( % )	Porcentaje Acumulado
1	CMAC Piura	734 755	22.25	22.25
2	CMAC Arequipa	513 447	15.55	37.80
3	CMAC Trujillo	506 874	15.35	53.15
4	CMAC Cusco	304 093	9.21	62.36
5	CMAC Sullana	287 323	8.70	71.07
6	CMAC Huancayo	263 398	7.98	79.04
7	CMAC Tacna	148 652	4.50	83.55
8	CMAC Ica	138 171	4.18	87.73
<b>9</b>	<b>CMCP Lima</b>	<b>100 440</b>	<b>3.04</b>	<b>90.77</b>
10	CMAC Maynas	99 939	3.03	93.80
11	CMAC Paita	98 161	2.97	96.77
12	CMAC Del Santa	87 954	2.66	99.44
13	CMAC Pisco	18 610	0.56	100.00

---

### Patrimonio

	<b>Empresas</b>	<b>Monto</b>	<b>Participación ( % )</b>	<b>Porcentaje Acumulado</b>
1	CMAC Arequipa	140 450	16.72	16.72
2	CMAC Piura	134 937	16.06	32.78
3	CMAC Trujillo	122 774	14.61	47.39
4	CMAC Cusco	91 800	10.93	58.32
5	CMAC Sullana	73 627	8.76	67.08
<b>6</b>	<b>CMCP Lima</b>	<b>65 562</b>	<b>7.80</b>	<b>74.88</b>
7	CMAC Huancayo	65 082	7.75	82.63
8	CMAC Tacna	42 060	5.01	87.64
9	CMAC Ica	35 199	4.19	91.82
10	CMAC Maynas	23 719	2.82	94.65
11	CMAC Paita	20 575	2.45	97.10
12	CMAC Del Santa	19 294	2.30	99.39
13	CMAC Pisco	5 097	0.61	100.00

Fuente: SBS

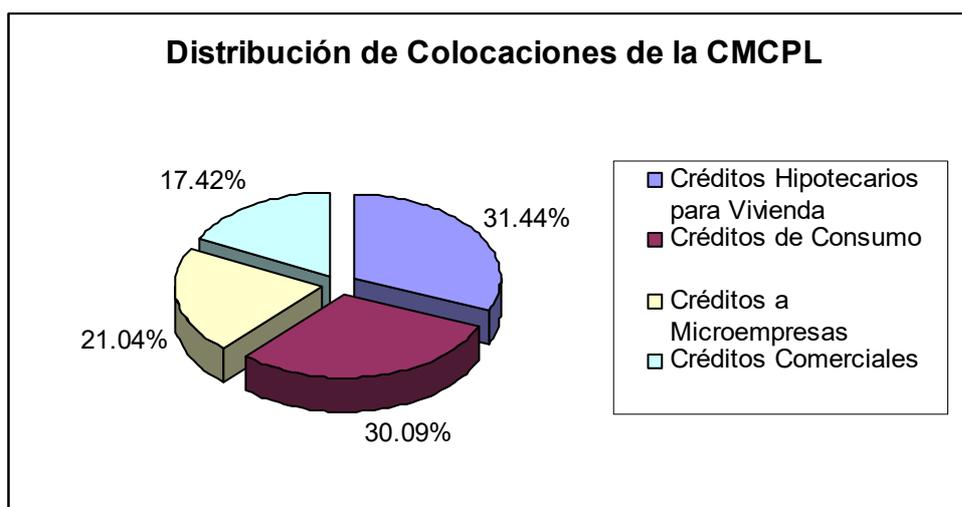
*Cuadro 1. Ranking de Colocaciones, Captaciones y Patrimonio por Caja Metropolitana a la fecha*

Como se puede observar en el Cuadro 1, la Caja Metropolitana ocupa el sétimo lugar en el ranking monto colocado en créditos, dentro del ámbito de las

Cajas Municipales. Asimismo ocupa actualmente el noveno lugar en depósitos (captaciones) del mercado mencionado.

En los últimos años se viene evidenciando un crecimiento sostenido en las colocaciones de la Caja Metropolitana, ello basado en gran medida a la nueva gama de productos de crédito que han salido a la luz, tales como los créditos hipotecarios (MiVivienda y recientemente MiHogar) en moneda nacional, los créditos para conversión de autos a gas (para taxistas Setame o público en general), asimismo casi el 12% de la cartera de colocaciones corresponde a los créditos *pignoraticios* o prendarios, es decir créditos con garantía en joyas que ha sido desde los inicios de la Caja Metropolitana el producto con mayor difusión de la institución en el entorno de la capital. También la Caja Metropolitana cumple el papel de ente recaudador de la Municipalidad de Lima, ejemplo de esto es la recaudación de pago de papeletas y tributos de la SAT que en la actualidad se ofrece como servicio.

A continuación en el cuadro 2 y figura 3, mostramos en detalle la participación de mercado de la empresa por tipo de crédito a la fecha, demostrando lo señalado líneas arriba, se puede apreciar la fortaleza de la empresa en los créditos de tipo hipotecario y consumo que juntos equivalen a cerca del un 61% de la cartera:



*Figura 3. Distribución de Colocaciones en la Caja Metropolitana a la fecha*

**Distribución de las colocaciones en la Caja Metropolitana**

**(Al 30 de Junio de 2007)**

(En miles de nuevos soles)

1	Créditos Hipotecarios	70 816	31.44%
2	Créditos de Consumo	67 786	30.09%
3	Créditos a Microempresas	47 405	21.04%
4	Créditos Comerciales	39 248	17.42%
<b>TOTAL</b>		<b>225 256</b>	<b>100.00%</b>

Fuente: SBS

*Cuadro 2. Distribución de las colocaciones en la Caja Metropolitana a la fecha*

Además podemos observar la ubicación actual de la Caja Metropolitana comparándola con las demás Cajas Municipales, como puede verse en los siguientes cuadros 3, 4, 5 y 6.

### Ranking de Créditos Directos por Tipo

(Al 30 de Junio de 2007)

(En miles de nuevos soles)

#### Créditos Comerciales

	Empresas	Monto	Participación (%)	Porcentaje Acumulado
1	CMAC Piura	204 874	27.37	27.37
2	CMAC Trujillo	139 650	18.65	46.02
3	CMAC Arequipa	86 595	11.57	57.59
4	CMAC Cusco	85 989	11.49	69.07
5	CMAC Sullana	47 684	6.37	75.44
6	CMAC Huancayo	41 601	5.56	81.00
<b>7</b>	<b>CMCP Lima</b>	<b>39 248</b>	<b>5.24</b>	<b>86.24</b>
8	CMAC Tacna	36 001	4.81	91.05
9	CMAC Del Santa	32 024	4.28	95.33
10	CMAC Ica	13 517	1.81	97.13
11	CMAC Maynas	12 568	1.68	98.81
12	CMAC Paita	8 245	1.10	99.91

13	CMAC Pisco	640	0.09	100.00
----	------------	-----	------	--------

---

Fuente : SBS

*Cuadro 3. Ranking de Colocaciones Comerciales por Caja Metropolitana a la fecha*

**Ranking de Créditos Directos por Tipo**

**(Al 30 de Junio de 2007)**

(En miles de nuevos soles)

**Créditos a Microempresas**

	Empresas	Monto	Participación (%)	Porcentaje Acumulado
1	CMAC Arequipa	369 851	19.99	19.99
2	CMAC Trujillo	349 968	18.92	38.91
3	CMAC Piura	264 053	14.27	53.19
4	CMAC Sullana	186 743	10.10	63.28
5	CMAC Cusco	146 454	7.92	71.20
6	CMAC Huancayo	125 581	6.79	77.99
7	CMAC Ica	106 191	5.74	83.73
8	CMAC Tacna	93 304	5.04	88.77
9	CMAC Paita	49 768	2.69	91.46

10	CMAC Maynas	48 189	2.61	94.07
<b>11</b>	<b>CMCP Lima</b>	<b>47 405</b>	<b>2.56</b>	<b>96.63</b>
12	CMAC Del Santa	45 036	2.43	99.07
13	CMAC Pisco	17 274	0.93	100.00

---

Fuente : SBS

*Cuadro 4. Ranking de Colocaciones a Microempresas por Caja Metropolitana a la fecha*

### Ranking de Créditos Directos por Tipo

(Al 30 de Junio de 2007)

(En miles de nuevos soles)

#### Créditos de Consumo

	Empresas	Monto	Participación (%)	Porcentaje Acumulado
1	CMAC Arequipa	165 651	16.22	16.22
2	CMAC Trujillo	157 204	15.40	31.62
3	CMAC Sullana	107 550	10.53	42.15
4	CMAC Huancayo	100 934	9.88	52.03
5	CMAC Piura	94 589	9.26	61.30
<b>6</b>	<b>CMCP Lima</b>	<b>67 786</b>	<b>6.64</b>	<b>67.94</b>

7	CMAC Cusco	67 757	6.64	74.57
8	CMAC Paita	64 692	6.34	80.91
9	CMAC Tacna	62 673	6.14	87.05
10	CMAC Maynas	55 401	5.43	92.47
11	CMAC Ica	36 115	3.54	96.01
12	CMAC Del Santa	34 031	3.33	99.34
13	CMAC Pisco	6 735	0.66	100.00

---

Fuente : SBS

*Cuadro 5. Ranking de Colocaciones Consumo por Caja Metropolitana a la fecha*

### **Ranking de Créditos Directos por Tipo**

**(Al 30 de Junio de 2007)**

(En miles de nuevos soles)

#### **Créditos Hipotecarios para Vivienda**

---

	Empresas	Monto	Participación (%)	Porcentaje Acumulado
<b>1</b>	<b>CMCP Lima</b>	<b>70 816</b>	<b>46.70</b>	<b>46.70</b>
2	CMAC Arequipa	31 256	20.61	67.31
3	CMAC Trujillo	26 783	17.66	84.97

4	CMAC Huancayo	5 908	3.90	88.87
5	CMAC Tacna	4 487	2.96	91.82
6	CMAC Sullana	4 160	2.74	94.57
7	CMAC Cusco	3 861	2.55	97.11
8	CMAC Ica	2 564	1.69	98.80
9	CMAC Paita	1 812	1.20	100.00
10	CMAC Del Santa	-	-	-
11	CMAC Maynas	-	-	-
12	CMAC Pisco	-	-	-
13	CMAC Piura	-	-	-

---

Fuente : SBS

*Cuadro 6. Ranking de Colocaciones Hipotecario por Caja Metropolitana a la fecha*

Esta institución cuenta actualmente con 19 agencias distribuidas en el departamento de Lima y 1 en la provincia de Chiclayo. Posee una infraestructura tecnológica que cumple con interconectar dichas agencias. La agencia principal administra los sistemas de información y la base de datos, manteniéndose un modelo centralizado de la información para asegurar la consistencia y seguridad de la información operativa de cada agencia.

Recientemente la SBS (Superintendencia de Banca y Seguros) ha autorizado a la Caja Metropolitana el poder brindar a sus clientes nuevos

productos antes solo reservados para las entidades bancarias, y que en función a cumplir determinados requisitos previos serán implementados a corto plazo. Puntualmente tenemos la reciente autorización de poder brindar el producto de Factoring y Descuentos de documentos valorados (letras, facturas y pagares).

Aun teniendo en cuenta lo anteriormente mencionado también se evidencia una fuerte competencia en el sector de Cajas Municipales por captar nuevos clientes, en lo que va del año se ha intensificado la apertura de nuevas sucursales de Cajas Municipales en la capital, atraídas por las buenas perspectivas económicas y la oportunidad de captar nuevos clientes. Así ya contamos en la capital desde algunos meses con agencias de diversas Cajas Municipales en la capital (CMAC Trujillo, CMAC Piura, CMAC Huancayo, CMAC Arequipa, CMAC Sullana) y en las provincias del departamento de Lima (CMAC Ica, CMAC Del Santa, CMAC Pisco), las cuales están posicionándose estratégicamente en el segmento de créditos a las Pymes, créditos comerciales de diversa índole, de consumo y en depósitos.

Teniendo en cuenta estos factores es que la Caja Metropolitana, en su estrategia de expansión de actividades requiere de una alternativa de “hacer negocio” que tome la iniciativa en proyectos que aprovechen lo que a priori podrían identificarse como amenazas (la expansión del resto de cajas municipales, sobretudo en la capital) y de ampliar su base de clientes tanto directa como indirectamente al poder ofrecerles nuevas alternativas de financiamiento. Al poder ampliar su alcance tecnológico la empresa se encontraría en mejor situación de enfrentar nuevos desafíos, que asimismo proveerá de una nueva forma de trabajo en el área informática y de negocio acelerando los tiempos de desarrollo y de salida al mercado de nuevos

productos. Al plantearse la posibilidad de utilizar una Orientación a Servicios se puede comunicar con otras entidades incluidas las otras Cajas Municipales, convirtiendo así una amenaza en una oportunidad de negocio.

Así entonces es claro sintetizar que la empresa adolece de un buen posicionamiento en lo que se refiere a créditos comerciales y orientados a las microempresas ya que en este aspecto la competencia ha acaparado la mayoría del mercado de Cajas municipales. Una de las explicaciones de esto es lo que mencionábamos líneas arriba: La llegada la capital y a las provincias de lima de las Cajas Municipales de provincias que cada día captan mas clientes. El impulso necesario a este tipo de crédito se puede dar a través de la implementación de un producto como el Factoring, el cual describiremos más adelante

### 1.2.3. La realidad de la arquitectura tecnológica en la empresa

Actualmente la arquitectura existente en la organización es la conocida Cliente/Servidor de dos capas, la cual hace el desarrollo de nuevas implementaciones extremadamente complejo para un entorno como el actual donde se requieren cambios muy rápidos para adaptarse a las necesidades de los clientes, el entorno y sobre todo para responder a la competencia con agilidad.

Las aplicaciones se desarrollan principalmente en lenguajes de programación de plataforma Microsoft y si bien se tiene claramente definido las funcionalidades y flujos de proceso de desarrollo, se hace imperioso un cambio

de arquitectura hacia un modelo multi-capa donde la independencia de plataforma sea un objetivo primordial a conseguir.

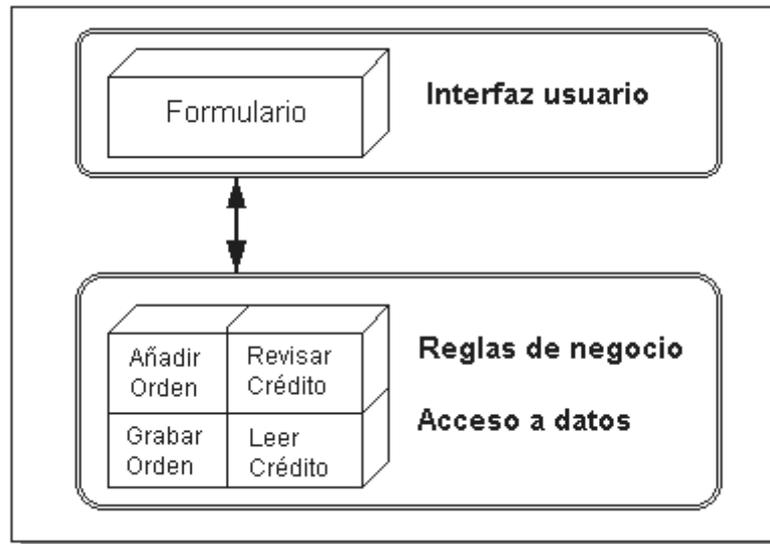
Algunas de las desventajas de una arquitectura de este tipo que pueden mencionarse son:

- El modelo de dos capas se encuentra ligado fuertemente a la implantación física.
- Cuando varios programadores realizan aplicaciones contra la misma base de datos se dificulta la administración del código.
- Si se desea modificar la aplicación cliente es necesario reinstalar esta en las N computadoras.

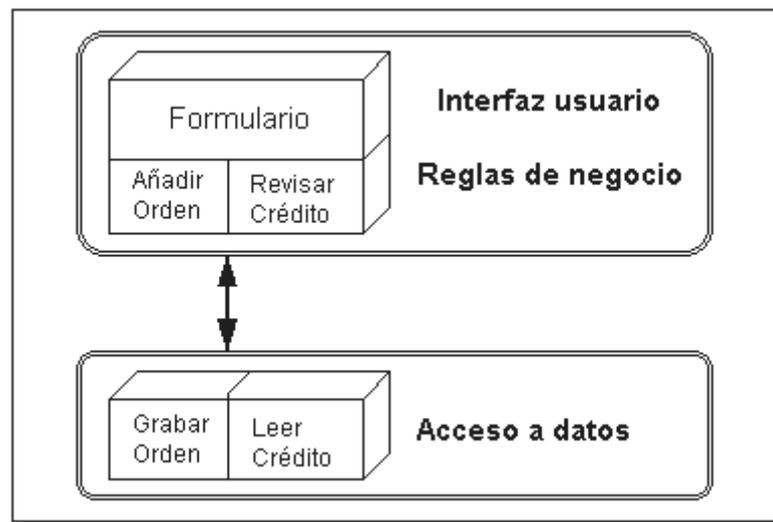
Actualmente no existe una definición clara de la ubicación de una capa intermedia de negocio ya que esta varía dependiendo del desarrollo, o incluso del programador designado por lo que se tienen dos variantes, una en la que la lógica de negocio se encuentra en el programa ejecutable, junto con la interfaz de usuario y la otra donde la lógica es implementada a través de store procedures es decir en la capa de acceso a datos, tal como podemos observar en los figura 4 y 5. Esto dificulta el inventario del sistema, así como incrementa los tiempos de respuesta de la organización haciendo más difícil integrar sistemas que puedan existir en el presente como a posteriori.

Demás está decir que este tipo de arquitectura es considerado hoy en día primitiva, puesto que existen evoluciones tecnológicas que facilitan el

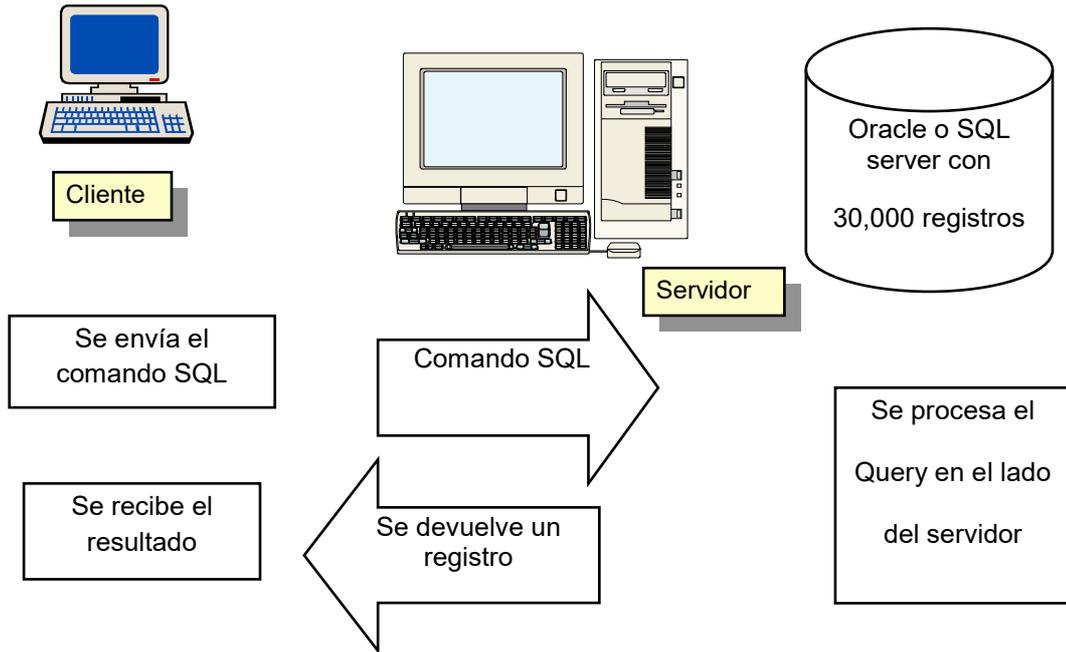
proceso de desarrollo y salida al mercado de nuevas funcionalidades (productos financieros) en la empresa.



*Figura 4. Arquitectura 2-capas dividida en dos entidades con el interfaz por un lado y las reglas de negocio junto con el Acceso a Bases de Datos por otro*



*Figura 5. Arquitectura 2 -capas con el acceso a la Base de Datos y las reglas de negocio junto a la interfaz de usuario*



*Figura 6. Proceso de acceso a datos en un modelo cliente/servidor*

En la figura 6 podemos observar el proceso de acceso a datos típico de una arquitectura Cliente / servidor tradicional donde no existe una capa intermedia, simplemente se solicita al servidor un resultado en base a una consulta vía comando sql, este modelo tuvo mucha aceptación en la década de los 80, donde el número de usuarios de una aplicación típica no era demasiado

En el marco de las intercomunicación entre la empresa y otras del mismo ramo, se tiene como experiencia previa que en el pasado se han realizado convenios con otras Cajas Municipales (CMAC Cusco y CMAC Paita), para que los clientes de sus respectivas Cajas Municipales puedan realizar operaciones de diversa índole sobres sus cuentas a través de la red de agencias de la Caja Metropolitana. Pero se ha podido observar con el transcurrir de los meses que el retorno de inversión no es el esperado

principalmente debido a que los costos implementación y mantenimiento son mayores a los esperados.

Estas instituciones cuentan con una plataforma también de tipo Microsoft, es decir la plataforma tecnológica es similar, lo cual presenta un panorama homogéneo entre las instituciones y satisface los requerimientos de una implementación de una arquitectura orientada a servicios básica.

Uno de los puntos más cuestionados ha sido que el esquema escogido en su oportunidad para este tipo de implementación fue el de utilizar *middlewares*, que permitieran la comunicación y el paso de transacciones entre las entidades empleando tramas de datos. Si bien este tipo de software es muy empleado en la actualidad en el sistema financiero, presenta desventajas en el tema del licenciamiento ya que en el mercado existen pocas alternativas de donde escoger, y requieren mantener equipos en las instalaciones de la Caja Metropolitana -en el peor de los casos inclusive requiriendo mantener un servidor por cada convenio con cada Caja Metropolitana -, además existe (y es la observación más importante a este modelo) la presencia de un intermediario o tercero que se encarga de la implementación en ambos nodos, estando muchas veces ligado a una tecnología, protocolo o arquitectura específica que implemente el proveedor del *middleware*.

Así si bien este modelo basado en *middleware* es válido, ya que cumple con lograr que dos o más empresas se comuniquen entre sí -en este caso con la ventaja de que sus plataformas tecnológicas son similares -, requiere de cierto grado de dependencia a un proveedor y no se puede realizar cambios de manera tan rápida si antes coordinar previamente con el proveedor del

*middleware* y con el otro nodo (la otra entidad financiera por ejemplo) sobre alguna modificación en los parámetros o en el tipo de información que se va a transmitir lo cual hace engorroso la implementación de este tipo de soluciones

Por tanto, algunos de las preocupaciones principales en este tipo de implementación que se han encontrado a lo largo del tiempo han sido:

- Dependencia del proveedor del servicio de software *middleware*, Normalmente utilizan sistemas propietarios o que requieren la asistencia de técnicos especialistas.
- Adquisición de licencias del software *middleware*, que por ser pocos los proveedores en el mercado actual representan una fuerte inversión por parte de la institución que quiera adquirirlos
- Limitación de futuros requerimientos por parte de la Caja Metropolitana a no ser que el proveedor de *middleware* las implemente previamente, lo que puede llevar a retrasos en el aprovechamiento de oportunidades de negocio.
- Mantenimiento de servidores y conexiones de túnel tipo VPN (red privada virtual) que requieren líneas dedicadas de conexión punto a punto.
- Temas concernientes con la seguridad de la información que es transmitida, normalmente en tramas que requieren encriptación por parte del proveedor del *middleware*

### **1.3. Antecedentes del problema**

En el mercado tanto nacional como internacional no han sido aprovechadas las características de la implementación de arquitecturas orientadas a servicios para resolver problemas de negocio como los que plantea un producto como el Factoring, pero son conocidas las soluciones de investigaciones previas en el uso de los Servicios Web en un gran número de instituciones, a continuación veamos algunas de ellas:

#### **1.3.1. Caso de Éxito: Caixa Galicia<sup>3</sup>**

*País:* España

*Sector:* Sector financiero

*Perfil del cliente:*

Caixa Galicia se constituyó en 1978 por fusión de la Caja de Ahorros y Monte de Piedad de La Coruña y Lugo y la Caja General de Ahorros y Monte de Piedad de Ferrol. En junio de 2000 contaba con 653 oficinas en España.

*Situación de Negocio:*

Anticipándose al nuevo marco normativo sobre valoración de bienes inmuebles, Caixa Galicia se planteó realizar una solución tecnológica que cubriese todas las etapas en la concesión de un préstamo hipotecario.

---

<sup>3</sup> [http://www.microsoft.com/spanish/msdn/spain/casestudies/cs\\_caixagalicia02.msp](http://www.microsoft.com/spanish/msdn/spain/casestudies/cs_caixagalicia02.msp)

### *Solución:*

En junio de 2002 Caixa Galicia inició el desarrollo de la nueva aplicación web, la Extranet del Mercado Hipotecario, en colaboración con dos empresas del grupo financiero: Softgal y Tasa Galicia.

### *Beneficios:*

Personalización y rapidez de respuesta, Seguridad y estabilidad, Ahorro de costes, Integración y escalabilidad, Flexibilización de las tareas

*Software:* Microsoft .NET Foundation Services, XML Web Services

*Escenario:* Sector financiero

### *Introducción*

Basada en la tecnología .NET y los Servicios Web, Caixa Galicia ha desarrollado la Extranet del Mercado Inmobiliario, una novedosa herramienta que engloba todos los procesos necesarios para conceder un préstamo hipotecario. Con ella, los empleados de Caixa Galicia disponen de todos los elementos necesarios para tener la capacidad de respuesta que precisa una entidad financiera. Además, han logrado solucionar uno de los problemas característicos de Internet: la Integración de aplicaciones desarrolladas por distintos proveedores. Todo esto se ha traducido en una simplificación del trabajo y, por ende, en una importante reducción de costes.

### *Situación*

En marzo de 2003, el Ministerio de Economía dio luz verde a la nueva legislación sobre normas de valoración de bienes inmuebles, que derogaba la

anterior regulación del sector, vigente desde noviembre de 1994. El principal objetivo era homologarla a la normativa europea e introducir cambios que ayudasen a potenciar la calidad técnica y formal de las valoraciones.

Así, por ejemplo, se contempla por vez primera la posibilidad de realizar la tramitación on line de los informes de tasación, con la ventaja añadida de poder incorporar la firma electrónica en los documentos.

Anticipándose a este nuevo marco normativo, Caixa Galicia comenzó a plantearse en 2002 la realización de una nueva solución tecnológica que cubriese todas las etapas en la concesión de un préstamo hipotecario por parte de la entidad financiera. "Queríamos aprovechar el cambio normativo para desarrollar una solución que extrajese todo el beneficio posible del entorno de Internet y los servicios web", apunta Fernando Pereira, jefe de Diseño y Desarrollo de la Dirección General Adjunta de Sistemas de Caixa Galicia.

### *Situación*

En junio de 2002, Caixa Galicia inició el desarrollo de la nueva solución, denominada Extranet del Mercado Hipotecario, en colaboración con Softgal, una empresa creada en 1988 por la propia entidad para ofrecer servicios de consultoría en tecnologías de la información, y Tasa Galicia, firma que también pertenece al grupo financiero gallego y especializada en la tasación de todo tipo de bienes e inmuebles. Se trata de una aplicación web que engloba todos los servicios para conceder un préstamo hipotecario. Para ello, está dividida en tres grandes módulos: Verificación Registral (antes de iniciar la gestión del riesgo es necesario comprobar la identificación registral del inmueble para el que se desea solicitar un crédito hipotecario), Tasación (a continuación se

procede a valorar el inmueble a hipotecar) y Gestión Hipotecaria (por último, y una vez concedido el préstamo hipotecario, es necesario abonar los impuestos obligados e inscribir el bien en el Registro de la Propiedad). La creación del módulo Tasación concluyó a principios de 2003; los otros dos, estaba previsto que se pusieran en funcionamiento el pasado mes de octubre, tras pasar un periodo de prueba.

La base tecnológica en la que se ha basado el desarrollo de la Extranet del Mercado Hipotecario ha sido la plataforma .NET de Microsoft y, más concretamente, Microsoft WinForms .NET. En un principio, los responsables de Caixa Galicia y Softgal se plantearon la conveniencia de utilizar la nueva plataforma de Microsoft, dada su juventud y poca experiencia en el mercado.

Finalmente, decidieron apostar por ella: "Su empleo casaba perfectamente con la filosofía de nuestro proyecto, especialmente en lo referente a la utilización de los servicios Web", indica Jorge García. "Desde el primer momento ha proporcionado a los técnicos un entorno muy productivo para desarrollar aplicaciones aparentemente muy difíciles de realizar", añade. Para José Antonio Ortega, director de la División de Tecnología de Softgal, uno de los principales motivos que llevó a la utilización de esta tecnología fueron "los niveles de integración que .NET ofrece, así como su amplio abanico de herramientas, basadas en estándares que garantizan un alto grado de compenetración con otras plataformas, como los host bancarios". A esto, Carlos González, responsable de Producción de la División de Tecnología de Softgal, añade "la integración con todas las tecnologías necesarias para el éxito del proyecto: XML, Servicios Web, seguridad, soporte de certificados digitales."

Además, para poder crear un escenario escalable que permitiese soportar la invocación de su funcionalidad desde diferentes sistemas, se optó asimismo por la utilización de Servicios Web.

Por último, se empleó XML como tecnología de acceso a datos, de tal forma que los documentos se reciben, manipulan y envían en este estándar de comunicación.

De los tres apartados, el más complejo desde el punto de vista técnico ha sido el de Tasación, dada la necesidad de definir todos los campos que debían incluir los informes de toda clase de inmuebles (pisos, naves industriales, barcos, edificios.). "Tuvimos que reunirnos con los tasadores especializados en cada apartado para desarrollar la interfaz que presentase los datos de todos los inmuebles", señala Jorge García.

En cuanto a la aplicación cliente, se desarrolló sobre Microsoft WinForms .NET para permitir al tasador la descarga de toda la información que necesita para realizar su labor y trabajar posteriormente en modo desconectado allí donde lo desee.

El empleo de .NET ha permitido dotar a este proyecto de destacadas características funcionales, como la creación de todo el entorno Web de cumplimentación de un informe de tasación, "aportando una interfaz de usuario equivalente a cualquier aplicación de un cliente tradicional", afirma Carlos González. Además, la combinación de .NET y Servicios Web ha permitido añadir a la nueva aplicación una serie de funcionalidades complementarias que resultan de gran valor para optimizar el trabajo de los empleados de Caixa Galicia. Una de ellas es el simulador de tasación que, a partir de los datos y

valores de mercado, permite obtener un valor aproximado de una tasación. Otras herramientas han sido los tutoriales para que los tasadores se encuentren permanentemente informados de todas las novedades que surjan en el sector; un sistema de ayudas para resolver cualquier duda técnica, y un sistema de control que, a través de un sencillo cuestionario, permite comprobar el conocimiento normativo de los tasadores. En un futuro está prevista la incorporación de otras funcionalidades, como la carga y descarga de planos y fotografías pertenecientes a los inmuebles que son objeto de tasación.

El objetivo de Caixa Galicia es que esta herramienta continúe creciendo e incorporando a "nuevos actores, como notarios o abogados, que se encargarán de revisar y verificar en línea que todas las escrituras son correctas antes de iniciar el proceso de concesión de un préstamo hipotecario", señala Fernando Pereira, para quien la Extranet del Mercado Hipotecario es sólo el comienzo de una larga colaboración con Microsoft. "Es un proyecto que define el patrón de diseño que tendrán las futuras aplicaciones de Caixa Galicia", concluye.

### *Beneficios*

- Personalización y rapidez de respuesta. Para Manuel Ansede, de Tasa Galicia, los beneficios que aporta esta nueva herramienta a Caixa Galicia son de vital importancia en un mercado como el financiero, en el que el elemento diferenciador que hace que un cliente se decante por una entidad u otra es el servicio personalizado y la capacidad de respuesta. "La Extranet del Mercado Hipotecario nos proporciona todos

los elementos necesarios para tener la capacidad de respuesta que las entidades financieras requieren", asegura.

- Seguridad y estabilidad. "Esta herramienta hace posible que la cobertura de las hipotecas que concede Caixa Galicia posea un valor seguro, sostenible y estable en el tiempo
- Ahorro de costes. "Se ha conseguido un importante ahorro de costes administrativos al simplificar la labor de los profesionales de Caixa Galicia, pero además se dispone de toda la información en tiempo real al eliminar la necesidad de enviar físicamente la información", apunta el responsable de Tasa Galicia. Esto último ha sido posible gracias a la incorporación de los certificados digitales y la firma electrónica. "En un futuro, cuando el empleo de la firma digital comience a extenderse, documentos como las escrituras presentes en la Extranet del Mercado Hipotecario también podrán compartirse en tiempo real con notarías y registros", explica.
- Integración y escalabilidad. "La Extranet del Mercado Hipotecario soluciona un problema característico de Internet, como es la integración de aplicaciones desarrolladas por distintos proveedores", afirma Jorge García, consultor tecnológico de Caixa Galicia. "Nuestro propósito era integrar todos los procesos que ofrecen las herramientas pertenecientes a los tres módulos, de tal forma que para los empleados de las oficinas de Caixa Galicia se convierta en un proceso único y horizontal", añade. "La tecnología .NET proporciona un entorno de desarrollo que permite trabajar con las últimas tecnologías y estándares de forma sencilla y rápida, lo que garantiza la escalabilidad del sistema y la extensión

funcional de los productos al simplificarse la comunicación con otros sistemas, incluso los que no son de Microsoft, a través de los web services", explica José Antonio Ortega. Por su parte, Carlos González indica: "Los web services tienen el valor añadido de proporcionar una forma común de comunicar sistemas, con independencia de su plataforma, mediante la utilización de estándares de facto como SOAP, WSDL o UDDI, lo que facilita además la comunicación con otros Servicios Web y aplicaciones".

- Flexibilización de las tareas. "XML ha hecho posible la realización de informes flexibles, dinámicos, configurables por medio de plantillas o directamente por el propio tasador"

### *Interfaces de los Servicios Web*

A continuación en las figuras 7, 8, 9 y 10 se muestran algunas de las interfaces de los Servicios Web implementados en la solución de Caixa Galicia

Cuentas y depósitos **CONSULTA DE PRÉSTAMOS Y CRÉDITOS**

**Préstamos y créditos**

Valores y fondos

Tarjetas

Recibos

Pago de impuestos

Planes de pensiones

Servicios móviles

Correspondencia

Varios

Posición

On Broker

Simuladores

Recomendaciones

Amortización      Simulador de cuotas      Simulador de amortización

Si desea **ampliar la consulta** de condiciones de su préstamo o crédito, pinche sobre el que desea consultar. Si desea **amortizar un préstamo** pinche sobre amortizar.

Tipo de Contrato	Moneda	Número	Saldo Actual	Vencido	Límite
Préstamo Personal	EUR	500-0000-00000-0	10.427,46 Eur	131,90 Eur	12.020,24 Eur
Préstamo Hipotecario	EUR	500-0000-00000-1	25.941,59 Eur	0 Eur	30.050,61 Eur
Cuentas de Crédito	EUR	550-5986-00000-1	108.977,59 Eur	18.825,77 Eur	90.151,82 Eur
Avales	EUR	500-0000-00000-2	8.969,58 Eur	0 Eur	8.969,58 Eur
<b>Total</b>			<b>154.316,22 Eur</b>	<b>18.957,67 Eur</b>	<b>141.192,25 Eur</b>

[¿Desea más información sobre los préstamos de Caixa Galicia?](#)

La amortización de préstamos hipotecarios, debe realizarse teniendo en cuenta las siguientes características:

- Ciertas modalidades de préstamos hipotecarios tienen una comisión por la amortización anticipada, lo que aparece reflejado en el contrato.
- El importe de la amortización debe coincidir con el de la cuota de capital o bien un múltiplo de la misma y una vez efectuada, tendrá fecha valor del día de pago de un vencimiento.
- La amortización de préstamos hipotecarios reduce el plazo del vencimiento de los mismos, excepto algunos préstamos con condición de adelanto de la cuota de los próximos vencimientos; dicha información aparece en el contrato del préstamo.

Figura 7. Interfaz Web de la solución de Caja Galicia – Consulta de Créditos

Cuentas y depósitos **AMORTIZACIÓN DE PRÉSTAMOS**

**Préstamos y créditos**

Valores y fondos

Tarjetas

Recibos

Pago de impuestos

Planes de pensiones

Servicios móviles

Correspondencia

Varios

Posición

On Broker

Simuladores

Recomendaciones

Si desea **amortizar** un préstamo, selecciónelo previamente, cubra los campos necesarios y pulse **AMORTIZAR**.

	Tipo de Contrato	Moneda	Número	Saldo Actual	Vencido	Límite
<input checked="" type="radio"/>	Préstamo Personal	EUR	500-0000-00000-0	10.427,46 Eur	131,90 Eur	12.020,24 Eur
<input type="radio"/>	Préstamo Hipotecario	EUR	500-0000-00000-1	25.941,59 Eur	0 Eur	30.050,61 Eur
	Cuentas de Crédito	EUR	500-0000-00000-3	108.977,59 Eur	18.825,77 Eur	90.151,82 Eur
	Avales	EUR	500-0000-00000-4	8.969,58 Eur	0 Eur	8.969,58 Eur
<b>Total</b>				<b>154.316,22 Eur</b>	<b>18.957,67 Eur</b>	<b>141.192,25 Eur</b>

Datos de la Amortización

Fecha: 16 / 6 / 2005

Importe:  Amortizar todo lo vencido pendiente de cobro  
 60,10 Eur

Amortizar

Atrás

Figura 8. Interfaz Web de la solución de Caja Galicia – Amortizaciones

caixa **activa** [Tarifas](#) | [Seguridad](#) | [Condiciones](#) | [Ayuda](#) | [Contacto](#)

Prueba en el ticker de Caixa Activa Salir

Cuentas y depósitos **AMORTIZACIÓN DE PRÉSTAMOS**

**Préstamos y créditos**

Valores y fondos  
Tarjetas  
Recibos  
Pago de impuestos  
Planes de pensiones  
Servicios móviles  
Correspondencia  
Varios  
Posición

On Broker  
Simuladores  
Recomendaciones

Su orden de Amortización es la siguiente:

<b>Préstamo</b>	500-0000- 00000-0
<b>Tipo</b>	Préstamo Personal
<b>Fecha</b>	16-06-2005
<b>Importe</b>	60,10 Eur

Si es correcto, teclee la coordenada **D1** y pulse **ACEPTAR**,  
en caso contrario pulse **CORREGIR** para realizar los cambios necesarios  
y si desea cancelar la operación pulse **CANCELAR**

Aceptar Corregir Cancelar

Figura 9. Interfaz Web de la solución de Caja Galicia – Amortizaciones II

caixa **activa** [Tarifas](#) | [Seguridad](#) | [Condiciones](#) | [Ayuda](#) | [Contacto](#)

ticker de Caixa Activa **CONSULTA DE PRÉSTAMOS Y CRÉDITOS.** CAIXA GALICIA Salir

Cuentas y depósitos **CONSULTA DE PRÉSTAMOS Y CRÉDITOS.**

**Préstamos y créditos**

Valores y fondos  
Tarjetas  
Recibos  
Pago de impuestos  
Planes de pensiones  
Servicios móviles  
Correspondencia  
Varios  
Posición

On Broker  
Simuladores  
Recomendaciones

**DATOS DEL PRÉSTAMO**

**Nº Prestamo** 500-0000- 00000-0

**DATOS GENERALES**

<b>Fecha Formalización</b>	<b>Fecha Vencimiento</b>	<b>Fecha Cambio Cuadro</b>
25-01-1994	24-01-2009	
<b>Cuenta Cargo</b>	<b>Garantía</b>	<b>Sistema Amortización</b>
2091-0059-06-0000000002	HIPOTECARIA	E. PROG. GEOMETRICA ANUAL.MOPU

**PENDIENTE DE PAGO**

Recibos pendientes de Pago	Número	Importe	Fecha Próximo recibo
	25	131,90 Eur	26-06-2005
<b>Recibos reclamados</b>	<b>Importe</b>		
	0 Eur		
<b>Recibos en Mora</b>	<b>Importe</b>		
	0 Eur		

**CONDICIONES**

Concedido	Dispuesto	Saldo	
12.020,24 Eur	12.020,24 Eur	10.427,46 Eur	
<b>Plazo en meses</b>	<b>Carencia</b>	<b>Próxima Amortización</b>	<b>Próxima liquidación</b>
180	0	25-06-2007	25-06-2007
<b>Periodo Fact Intereses</b>	<b>Periodo Fact, Capital</b>	<b>Interés Cuota</b>	<b>Capital Cuota</b>
001 MES	001 MES	118,27 Eur	154,37 Eur

**INTERESES Y COMISIONES**

Concepto	Tipo
INTERES NOMINAL	5,66
INTERES DE DEMORA	9,66
COMISION S/ADEL-CANC	0

Atrás

Figura 10. Interfaz Web de la solución de Caja Galicia – Consulta de Créditos II

### 1.3.2. Otros Casos de Éxito: <sup>4</sup>

#### **Gobierno de Dinamarca:**

El gobierno danés planea instituir una red basada en Servicios Web para comercio electrónico masivo, que permitirá generar documentos XML tales como órdenes de compra y facturas, para transacciones de los sectores público y privado. El nuevo sistema manejará mas de 200 millones de transacciones al año y dispondrá de un registro nacional de servicios, de manera de presentarse, de cara a las compañías danesas, como un modelo de referencia -basado en estándares de Servicios Web - para realizar transacciones confiables. El sistema se apoyará en los estándares SOAP 1.1, UDDI 3.0, WSDL 1.1, WS-Security 1.0 y WS-ReliableMessaging, y reemplazará a la red EDI propietaria que es actualmente usada para negocios por el gobierno danés, generando así un importante ahorro en costos de transformación de datos y posibilitando el comercio electrónico vía Web

#### **Srikelron Inc.: Un Marketplace de Servicios Web**

La compañía Srikelron Inc. desarrolló y opera su “Web Sevices MarketPlace”, diseñado específicamente como un medio para publicar, suscribir y comprar Servicios Web, que provee además funciones para “billing” y “tracking” del uso de los mismos. En esencia, es un directorio online con toda la infraestructura necesaria para que puedan encontrarse mutuamente tanto “providers” como consumidores de Servicios Web. Los Servicios Web allí

---

<sup>4</sup> <http://www.slideshare.net/ProfesorHariSeldon/eser5>

ofrecidos son tanto propios de Strikelron como de terceros, y cubren variadas funcionalidades, tales como verificación de direcciones, cálculo de impuestos sobre las ventas, obtención de cotizaciones, etc. En el caso de los Services Web provistos por [www.webservices.org](http://www.webservices.org), Strikelron retiene el 30 % del monto producido por la “venta” (uso) del servicio a consumidores.

### **Nokia - TNT**

Nokia se ha unido con la compañía TNT para crear una aplicación piloto basada en Servicios Web, para el seguimiento de envíos de paquetes en tiempo real. En el proyecto, personal de TNT utiliza equipos móviles Nokia 9500 para ingresar la información de seguimiento directamente en el sistema central de TNT. El empleado usa el celular para tomar una “foto” del código de barras adherido al paquete. Esta foto es convertida en un “mensaje” de Servicios Web y enviada, a través del mismo dispositivo, al sistema central de TNT.

### **Continental Airlines**

Como muchas otras compañías en el negocio del transporte aéreo, Continental Airlines enfrentaba desafíos que requerían continuar proporcionado un alto nivel de servicio. Mediante una relativamente pequeña cantidad de esfuerzo e inversión, esta compañía “envolvió” los datos de sus sistemas de administración de operaciones de vuelo con “wrappers” de Servicios Web. Esto posibilita acceder a dicha información desde dispositivos como PDAs y celulares, proporcionando a sus clientes y agentes datos en tiempo real del estado de sus vuelos, lo que constituye una invaluable prestación.

## **I-Med**

I-Med es una empresa chilena que tramita las autorizaciones de las órdenes de atención médica para el Sistema de Salud de ese país, compuesto por las ISAPREs (Instituciones de Salud Previsional o “financiadores”) y el Fondo Nacional de Salud. i-Med implementó una solución de servicios Web para hacer más eficiente el intercambio de información entre los prestadores y las ISAPREs, logrando una mayor transparencia en la gestión y la reducción en los tiempos de ejecución de los procesos. Hay 9 financiadores (ISAPREs) conectados al sistema, con un volumen de 4000 transacciones diarias, involucrando a un universo de 3 millones de pacientes, lo que representa verdadera “conexión” del mercado de salud en Chile.

## **Bank-Verlag**

Bank-Verlag es una subsidiaria de la Asociación de Bancos Alemanes y brinda servicios a ese sector financiero. Opera el sistema “iRecherche”, que actúa como una base de datos central de consultas, para ayudar a los bancos a identificar transferencias de pago fallidas o irresueltas, detectar las causas del problema y seguir su estado. La solución con servicios Web permite a los clientes de Bank- Verlag (los bancos) cursar las consultas programáticamente, desde sus propios sistemas internos de archivado y búsqueda de documentos electrónicos (como una alternativa a las consultas vía browser).

## **Amazon**

Amazon también provee una interfaz de Servicios Web para integrarse con terceros vendedores, para posibilitarles a éstos el envío de información sobre sus últimas líneas de productos y así actualizar la base datos de Amazon. Anunciada a mediados de septiembre de 2003, es también una de las más importantes experiencias de despliegue de servicios Web a nivel mundial. Para este desarrollo Amazon utiliza como plataforma a Systinet WASP.

### **Dollar Rent-A-Car**

A través de una alianza comercial y mediante el desarrollo de un Servicio Web con una inversión mínima, esta empresa se conectó a los sistemas de la compañía aérea "Southwest". De esa manera, los clientes de esa aerolínea pueden contratar un vehículo de "Dollar" en el mismo momento en que están haciendo sus reservas de pasajes en el sitio Web de "Southwest".

CSE Insurance (San Francisco, California): Construyó un Servicio Web en la forma de un programa de cálculo de primas de pólizas, disponible en tiempo real para sus productores, y que puede ser usado también por aplicaciones de terceros.

### **Storebrand ASA**

Con sede en Oslo, Noruega, suministra servicios financieros (incluyendo banca, administración de activos y seguros de vida y salud) a unos 280.000 clientes. La aplicación basada en Servicios Web, captura automáticamente datos de empleados desde los sistemas de nóminas de las compañías clientes, calculando los montos de pensiones, y envía la información a la base de datos de Storebrand. Los beneficios: Storebrand ya no necesita ingresar y chequear

manualmente la información de los empleados de sus compañías clientes, lográndose un aumento de eficiencia y liberando personal para otras tareas. Plataforma: El desarrollo está basado en plataforma IBM.

### **Nordstrom.com**

Quiere utilizar Servicios Web para transmitir a los fabricantes de calzado la información de órdenes de compra y de inventarios, así como para disponer (en tiempo real) reportes estadísticos que permitan aparear mejor los inventarios con las estimaciones de ventas. Plataforma de desarrollo: “Nordstrom.com” emplea la plataforma del vendedor IONA Technologies.

### **Life Time Fitness**

Con sede en Minnesota, EE.UU., esta empresa opera una cadena de 23 “resorts” de deportes ubicados en distintos estados. La aplicación basada en Servicios Web services: Integra el sitio Web y la intranet de la empresa con un sistema online de “scheduling”, suministrado por la compañía Xtime (que es un Service Provider). Los beneficios: Los miles de miembros de Life Time pueden reservar las instalaciones y servicios de la compañía desde el sitio Web de la empresa. Los empleados pueden hacer lo mismo desde la Intranet. Plataforma de desarrollo: El desarrollo está basado en plataforma Sun.

### **Con-Way Transportation Services**

“Con-Way” Transportation Services es una empresa de transportes y logística, que opera en EE.UU., Canadá y Méjico. Situación inicial: Como

transportadora de un alto volumen de cargas entre EE.UU. y Canadá, que requiere frecuentes cruces de frontera entre ambos países, cada camión de Con-Way perdía de 2 a 3 horas de tiempo por los papeleos propios de Aduana. La aplicación web services: Con-Way desarrolló un Servicio Web que conecta en forma automática a las Aduanas todos los formularios debidamente completados, permitiendo reducir drásticamente el tiempo de control, de manera que cada camión pueda cruzar la frontera en menos de 1 minuto.

## **Bekins**

Compañía dedicada al negocio de las mudanzas internacionales, logística y transportes. Está establecida en el mercado desde 1891 y su sede central está en Hillside, Illinois.

*Situación inicial:* Bekins obtiene muchos de sus viajes a través de sus agentes en todo EE.UU. Tiene links con los principales mediante EDI. Las órdenes de la mayoría de los agentes (y el consiguiente espacio que se va ocupando en los camiones) son tomadas por empleados de Bekins por teléfono, fax o correo electrónico. La integración online del mainframe IBM de Bekins con los sistemas de sus agentes y socios es prohibitiva, por su alto costo y plazo de concreción. Por razones de eficiencia operativa, Bekins trata de asegurarse de que sus camiones partan hacia sus destinos con la máxima carga posible.

*Funcionalidades a exponer:* Bekins quiere exponer como Servicios Web varios componentes de sus aplicaciones actuales, que muestren la disponibilidad de espacio en sus camiones, horarios y tarifas. Los agentes de

Bekins podrán invocar tales Servicios Web desde sus propias aplicaciones, automatizando y abaratando para ambas partes una integración.

*Beneficios:* Además de resolver el problema puntual, el acercamiento a los Servicios Web permitirá a Bekins ofrecer a sus clientes otros servicios que antes eran operativamente muy costosos, debido a la complejidad de la integración online con los agentes. Otras partes de las aplicaciones Bekins y de sus agentes podrían ser expuestas como Servicios Web: chequeo de crédito de clientes, pre armado de declaraciones de aduana para transportes internacionales y conversiones de cargos en monedas extranjeras

#### **1.4. Justificación e importancia de la investigación**

##### 1.4.1. Justificación de carácter práctico

Desde el punto de vista práctico, este trabajo sugerirá la aplicación de una moderna tendencia en el terreno de la arquitectura de software, como es la orientación a servicios en una organización financiera que busque por un lado un mayor desarrollo empresarial no solo a nivel de tecnología sino a nivel de procesos de negocio incrementando con ello la productividad del capital humano de la organización. También con esto se consigue la mejora o inclusive el rediseño de los flujos de procesos de negocio, ya que no es un cambio que afecta únicamente al área de tecnología de la empresa sino a la forma de ver los productos como servicios que pueden ser independientes y fácilmente acoplados, consiguiendo la reducción de los periodos de desarrollo y los costos involucrados, la forma en la que se interconectan los bloques

básicos de negocio para ir creciendo en complejidad de manera menos caótica, esto con la finalidad de responder prontamente a los cambios del entorno financiero del país.

Asimismo propone el cambio de paradigma en la organización que se tenía con respecto a la administración de las tecnologías de información, esto porque requiere de un cambio de visión en la misma ya que implica conocer bien los procesos del negocio, clasificarlos, extraer las funciones que son comunes a ellos, estandarizarlas y formar con ellas capas de servicios que serán requeridas por cualquier proceso de negocio, no solo para un caso específico como el que tocamos en este trabajo (el Factoring) sino para cualquier nueva implementación.

#### 1.4.2. Justificación de carácter metodológico

Actualmente muchas Empresas en nuestro país poseen un desarrollo de Software usando la Arquitecturas de Cliente Servidor (C/S) o de N-Capas. Al usar estas dos metodologías de desarrollo de software se tienen las siguientes ventajas y desventajas respectivamente:

- Cliente Servidor

##### *Ventajas:*

- Alto rendimiento transaccional.
- Alta disponibilidad.
- Entorno probado y personal experimentado.

- Control total del ordenador, al ser éste único y residente en un único Centro de Proceso de Datos.
- Concentración de todo el personal de explotación y administración del sistema en un único Centro de Proceso de Datos.
- Alto nivel de seguridad

*Inconvenientes:*

- Alto precio del ordenador, al requerirse mucha potencia de tratamiento para dar servicio a todos los usuarios que estén conectados y gran espacio en disco para albergar todos los datos del organismo.
  - Alta dependencia de las comunicaciones si existen. En caso de caída de una línea, todos los puestos de trabajo dependientes de dicha línea quedan inoperantes.
  - Arquitecturas propietarias
- N-Capas

*Ventajas*

- Funcionamiento autónomo de los sistemas locales, lo que origina un buen tiempo de respuesta.
- Abre posibilidades de trabajo mucho más flexibles y potentes.

*Inconvenientes*

- Requiere un intenso flujo de informaciones (muchas veces no útiles, como pantallas y datos incorrectos) dentro de la red, lo que puede elevar los costes de comunicaciones.
- Supone una mayor complejidad.
- Si los sistemas no están integrados, pueden producirse problemas de inconsistencia de datos.

Al usar la Arquitectura SOA podremos permitir la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como HTTP) y de manera independiente al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo. Un Servicio Web es un contenedor que encapsula funciones específicas y hace que estas funciones puedan ser utilizadas en otros servidores.

Entre sus ventajas tenemos: son programables, están basados en XML, que es un lenguaje abierto, son auto descriptivos y pueden buscar registros de otros Servicios Web.

Al comparar los Servicios Web sobre otras tecnologías decimos que los Servicios Web presentan escaso acoplamiento es decir el cliente no necesita conocer nada acerca de la implementación del servicio al que está accediendo, salvo la definición WSDL; independencia del lenguaje de programación (el servidor y el cliente no necesitan estar escritos en el mismo lenguaje); independencia del modo de transporte. SOAP puede funcionar sobre múltiples protocolos de transporte, como por ejemplo HTTP, HTTPS, SMTP o FTP.

#### 1.4.3. Justificación de carácter teórico

Desde el punto de vista teórico, el presente trabajo de investigación va a permitir enriquecer la concepción teórica sobre arquitecturas de tecnologías de información, enfocados en el encapsulamiento, publicación y exposición de servicios a través de interfaces transparentes para el consumidor, permitiendo con ello la independencia de la implementación de cada uno de estos servicios.

Al tener un enfoque orientado a los servicios las interfaces se define de una manera neutral que debe ser independiente de la plataforma de hardware, del sistema operativo y del lenguaje de programación en los que se implemente el servicio. Esto permite que los servicios, puedan ser construidos en una gran variedad de sistemas, interactúen entre sí de una manera uniforme y universal por medio de un lenguaje común estandarizado

#### 1.4.4. Importancia de la investigación

La importancia de este trabajo radica en que la implementación de una nueva arquitectura -orientada a servicios- con lleva a una mejora sustancial en los tiempos de desarrollo y salida al mercado de nuevas funcionalidades o de la exposición de funcionalidad que ya se encuentra implementada en diversos sistemas de la organización y que pueda ser consumida por los clientes de la empresa, socios de negocio, además su importancia esta en el aprovechamiento de una importante oportunidad de negocio en el nicho de las Cajas Municipales como es el de contar con un producto que no es ampliamente difundido en este sector pero que si ha resultado satisfactoriamente implementado en el sector de la banca, usando una arquitectura flexible que permita rápidamente captar una cartera de créditos

inclusive usando las ventanillas de otras instituciones, sin dejar de mencionar que esta arquitectura también beneficia a los clientes finales ya que permite otorgar servicios independientes –que ya se encuentren implementados, en los sistemas informáticos de la Caja Metropolitana - y que pueden ser de diversa índole como las consultas de sus financiamientos. De este modo se plantea obtener los siguientes beneficios:

- Mejorar el posicionamiento de la empresa en el mercado de colocaciones a nivel tanto local como nacional
- Aumentar la flexibilidad de los sistemas al exponer servicios independientes de la plataforma en la que se encuentren los consumidores de los servicios
- Reducir drásticamente los costos y tiempos de implementación de estos servicios ya que por sus características son encapsulados, reutilizables y no dependen de terceros como en el caso de los *middleware* para su adaptación, impactando en una mayor rapidez de la organización para adaptarse a los cambios del mercado

## **1.5. Delimitaciones del problema**

### **1.5.1. Limitaciones temporales**

En el aspecto temporal no existen mayores limitantes más que la de la propia vigencia de la tecnología empleada ya que esta es cambiante o puede evolucionar en función de las necesidades de la organización

### 1.5.2. Limitaciones espaciales

Se limita a nivel geográfico al Perú ya que se aplica a una institución financiera que se encuentra ubicada en el país y brinda servicios en el ámbito nacional.

### 1.5.3. Otras Limitaciones

El problema está limitado también por las características de la institución que si bien es una entidad financiera no cuenta con los recursos con los que podría contar una entidad de mayor envergadura como un banco. En este caso existen limitantes en el aspecto de infraestructura. Gran parte de la plataforma tecnológica, sobretodo la que da mayor soporte a sus procesos de negocio se encuentra ligada a soluciones Microsoft, en lo que se refiere a los sistemas de base de datos y aplicativos core.

## **2. FORMULACION DEL PROBLEMA**

### **2.1. Objetivos**

#### 2.1.1. Objetivos Generales

Proponer una arquitectura tecnológica orientada a servicios para entornos homogéneos, que soporten el desarrollo e implementación de soluciones de negocio, mediante la exposición de funcionalidad existente, como el Factoring electrónico en la Caja Metropolitana.

#### 2.1.2. Objetivos específicos

- Diseño de la arquitectura tecnológica empresarial
- Determinar una metodología para la generación de servicios en la empresa
- Identificación de estándares y protocolos a utilizarse para la comunicación
- Bosquejo de la solución de negocio de Factoring basado en arquitectura orientada a servicios

## **2.2. Definición del problema**

El problema, que es tema de esta investigación reside en buscar una alternativa a la arquitectura actual, la cual ha quedado ya en un estado de obsolescencia y que no permite la rápida adecuación de los sistemas de información con los requerimientos del negocio, para poder implementar con agilidad y rapidez nuevos servicios a sus clientes a través de la exposición de servicios e inclusive permitir la interoperabilidad entre la Caja Metropolitana y otras entidades financieras utilizando una arquitectura basada en servicios (SOA).

Asimismo debe soportar la interoperabilidad hacia dentro de la propia empresa permitiendo que sus propias aplicaciones puedan integrarse sin necesidad de reconstruir las diversas aplicaciones sobre una plataforma homogénea. En este caso particular servirá para dar soporte a la creación de un nuevo producto crediticio en la institución: El Factoring electrónico.

Al tomar en consideración la problemática descrita en la realidad líneas arriba y considerando la confluencia de dos factores:

- La oportunidad de negocio en la que actualmente la Caja Metropolitana es la única entidad financiera en el sector de las cajas municipales en contar con la autorización de otorgar líneas de financiamiento de Factoring,
- La necesidad de una evolución en la arquitectura tecnológica actual de la empresa, que se autónoma de la empresa es decir sin dependencia de terceros para la interoperabilidad de aplicaciones.

Es que se toma la determinación de bosquejar una arquitectura que provea de una adecuada arquitectura tecnológica que impacte drásticamente en la organización, cambiando de paradigma el cual permita la integración de aplicaciones y que posibilite a través de esta además –por medio de convenios con otras instituciones financieras - exponer funcionalidad ya implementada de los sistemas de información que vienen trabajando en la institución, logrando con ello una ventaja competitiva contra el resto de las Cajas Municipales.

Actualmente la Caja Metropolitana no cuenta con una arquitectura que soporte tales características de integración e intercomunicación entre diversas entidades, existen proyectos anteriormente implementados que permiten la interoperabilidad de aplicaciones sobretodo con sistemas externos a la organización, pero están basados en comunicación de tipo middleware, los cuales además dependen de un proveedor específico y que no garantizan una rápida adecuación con otras instituciones sin antes tener que coordinar con el proveedor tecnológico.

Entre los aspectos que se busca solucionar se encuentran los siguientes:

- Permitir la integración de sistemas legados que se encuentran funcionando dentro de la organización bajo una plataforma homogénea y que requieren ser integrados para cumplir con un flujo de proceso de negocio, en este caso puntual con el producto de Factoring Electrónico.
- Permitir realizar operaciones vía otras entidades financieras o socios de negocio estratégicos los cuales consumirán nuestros servicios Web que encapsulan funcionalidad existente en nuestros sistemas internos. Para

por ejemplo permitir transacciones de retiros, cobranza o consultas de créditos, sin necesidad de establecer una fuerte dependencia entre los sistemas de información de las entidades con la que se celebren convenios y la Caja Municipal de Crédito Popular de Lima. Lo cual repercute en un rápido acoplamiento de aplicaciones y un menos tiempo de salida de nuestros servicios al mercado.

- Ofrecer a nuestros clientes la facilidad de poder realizar determinadas operaciones que les demandan gran cantidad de tiempo o costos, (Ej.: el envío electrónico de documentos para su calificación) esto agrega valor al proceso y beneficia a la imagen de la organización.
- Sentar las bases para el desarrollo de nuevos sistemas de información con un enfoque orientado a la re-utilización de código, aplicaciones y servicios existentes a lo largo de la organización, disminuyendo los tiempos de desarrollo, reduciendo la complejidad y eliminando la necesidad de mantener interfaces rígidas y dependientes de los cambios en cada uno de los aplicativos existentes.

El problema se encuentra completamente definido al ser descrito como el de diseñar una arquitectura tecnológica empresarial flexible capaz de soportar la integración de soluciones de negocio existentes y futuras, basadas en servicios tanto a nivel interno como externo que permita una respuesta más rápida a los cambios del entorno en la Caja Metropolitana.

### **3. MARCO TEORICO CONCEPTUAL**

#### **3.1. Bases Teóricas**

##### 3.1.1. El Financiamiento de la PYMES en el Perú

En el Perú, al igual que la mayoría de países de Latinoamérica, se ha observado en los últimos años un importante incremento del número de pequeñas y medianas empresas (PYMES), debido principalmente a los siguientes factores:

Entre las características de la PYMES tenemos<sup>5</sup>:

- **Alta Generación De Empleo:** Existen alrededor de 3.1 millones de PYMES ( 1.7 millones de PYMES urbanas y 1.4 millones de PYMES rurales)
  - Dan ocupación al 74% de la PEA (5.6 millones de trabajadores).
  
- **Alta Contribución Al PBI:** Las PYMES contribuyen con el 43% del PBI
  - PYMES urbanas con 34%
  - PYMES rurales con 9%

---

<sup>5</sup> [http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Empre/flores\\_ka/cap02.PDF](http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Empre/flores_ka/cap02.PDF)

- **Informalidad:** Sólo el 18% PYMES posee RUC.
  - El 78% de las PYMES urbanas están organizadas como "persona natural con negocio propio".
  - El 75% de las PYMES urbanas no cuenta con licencia de funcionamiento.

En el cuadro 7 mostramos la participación de las PYMES por actividad económica en el ámbito nacional:

<b>Actividad Económica</b>	<b>PYMEs N°</b>	<b>Estimadas %</b>	<b>PYMEs Contribuyentes N°</b>
Agropecuaria		1'380,000 45	42,746
Manufactura		255,000 8	125,688
Comercial		1'120,000 36	534,556
Servicios		345,000 11	519,199
<b>Total</b>		<b>3'100,000 100</b>	<b>1'222,189</b>

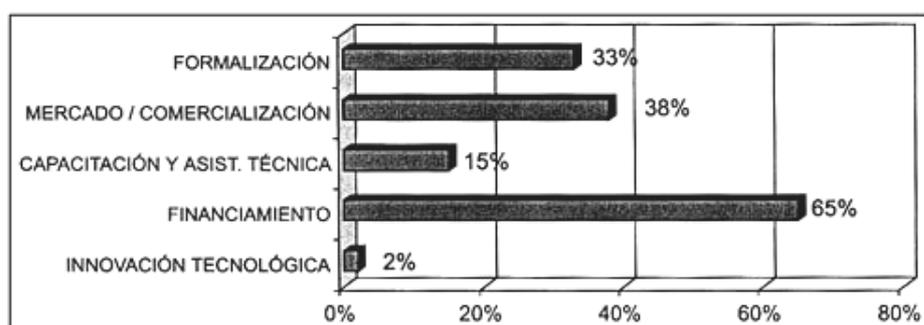
Fuente: INEI

*Cuadro 7. Proceso de acceso a datos en un modelo cliente/servidor*

El cuadro 8 refleja los principales problemas que enfrentan las pequeñas y medianas empresas en nuestro país.<sup>6</sup>

<sup>6</sup> [http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07\\_n13/a03.htm](http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07_n13/a03.htm)

1. Según su experiencia en la empresa ¿Cuáles de las siguientes necesidades resultan las más apremiantes?	
Innovación tecnológica	2%
Financiamiento	65%
Capacitación y asistencia técnica	15%
Mercado / Comercialización	38%
Formalización	33%



*Cuadro 8. Principales problemas revelados por Microempresarios - Pymes*

Sólo el mercado de las 1.7 millones PYMES urbanas presentan necesidades de financiamiento por el orden de los US\$ 4,000 millones. Se estima que un millón de las PYMES urbanas son potenciales sujetos de crédito y sus necesidades de financiamiento son de alrededor de US\$ 2,500 millones. Las PYMES han obtenido financiamiento formal por aproximadamente US\$ 250 millones. Es decir sólo se han atendidos el 10% de los requerimientos financieros.

Entre las ofertas de financiamiento en nuestro país, no existe una oferta estructurada de financiamiento para las PYMEs; recién en los últimos años se han creado entidades financieras especializadas para estas empresas.

Actualmente, la oferta de financiamiento es efectuada por una variada gama de entidades que canalizan recursos externos e internos, bajo diversas modalidades, entre ellas: CAF, BID, UNION EUROPEA, AID, COFIDE, FONCODES, FONDEMI, PACT, ONG's, EDPYMES, Cajas Rurales de Ahorro y Crédito, Cajas Municipales de Ahorro y Crédito, FONDEAGRO, Financieras y los Bancos.

En el cuadro 9 podemos observar que de lejos el financiamiento es el problema más agobiante para las empresas que están surgiendo en el país.

2. ¿Cuáles de las siguientes necesidades resultan más difícil de solucionar?	
Innovación tecnológica	20%
Financiamiento	75%
Capacitación y asistencia técnica	12%
Mercado / Comercialización	25%
Formalización	21%

*Cuadro 9. Ranking en dificultad de solucionar problemas revelados por  
Microempresarios - Pymes*

Normalmente las empresas toman el crédito a corto plazo que es una deuda que generalmente se programa para ser reembolsada dentro de un año ya que generalmente es mejor pedir prestado sobre una base no garantizada, pues los costos de contabilización de los préstamos garantizados frecuentemente son altos pero a su vez representan un respaldo para recuperar

A continuación se detallamos las diferentes fuentes de financiamiento que pueden ser utilizadas por las empresas:

**Fuentes de Financiamiento sin garantías específicas<sup>7</sup>** consiste en fondos que consigue la empresa sin comprometer activos fijos específicos como garantía.

### *Cuentas por Pagar*

Representan el crédito en cuenta abierta que ofrecen los proveedores a la empresa y que se originan generalmente por la compra de materia prima. Es una fuente de financiamiento común a casi todas las empresas. Incluyen todas las transacciones en las cuales se compra mercancías pero no se firma un documento formal, no se exige a la mayoría de los compradores que pague por la mercancía a la entrega, sino que permite un periodo de espera antes del pago. En el acto de compra el comprador al aceptar la mercancía conviene en pagar al proveedor la suma requerida por las condiciones de venta del proveedor, las condiciones de pago que se ofrecen en tales transacciones, normalmente se establecen en la factura del proveedor que a menudo acompaña la mercancía.

### *Pasivos Acumulados*

Una segunda fuente de financiamiento espontánea a corto plazo para una empresa son los pasivos acumulados, estos son obligaciones que se crean por servicios recibidos que aun no han sido pagados, los más importantes que acumula una empresa son impuestos y salarios, como los impuestos son pagos

---

<sup>7</sup> [www.monografias.com/trabajos33/fuentes-financiamiento/fuentes-financiamiento.shtml](http://www.monografias.com/trabajos33/fuentes-financiamiento/fuentes-financiamiento.shtml)

al gobierno la empresa no puede manipular su acumulación, sin embargo puede manipular de cierta forma la acumulación de los salarios.

### *Línea de crédito*

Es un acuerdo que se celebra entre un banco y un prestatario en el que se indica el crédito máximo que el banco extenderá al prestatario durante un período definido.

### *Convenio de crédito revolvente:*

Consiste en una línea formal de crédito que es usada a menudo por grandes empresas y es muy similar a una línea de crédito regular. Sin embargo, incluye una característica importante distintiva; a medida que el crédito se va cancelando por parte del cliente, esta línea va paulatinamente volviendo a su monto original cuando se cancela la deuda.

### *Documentos negociables*

El documento negociable consiste en una fuente promisorio sin garantías a corto plazo que emiten empresas de alta reputación crediticia y solamente empresas grandes y de incuestionable solidez financiera pueden emitir documentos negociables.

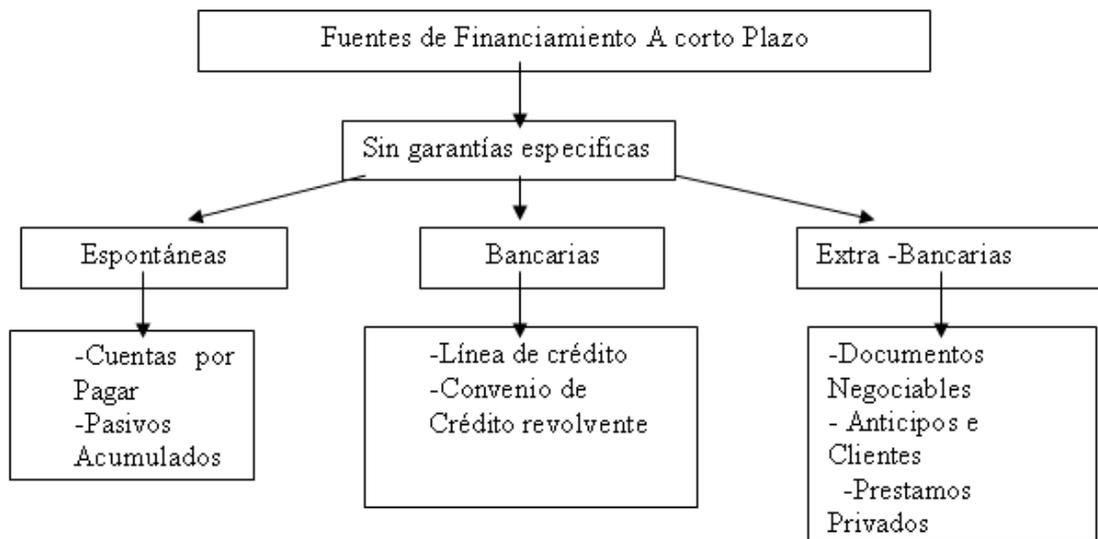
### *Anticipo de clientes.*

Los clientes pueden pagar antes de recibir la totalidad o parte de la mercancía que tiene intención de comprar.

### *Préstamos privados.*

Pueden obtenerse préstamos sin garantía a corto plazo de los accionistas de la empresa ya que los que sean adinerados pueden estar dispuestos a prestar dinero a la empresa para sacarla delante de una crisis.

Resumimos en la figura 11 las principales fuentes de financiamiento a corto plazo sin garantías.

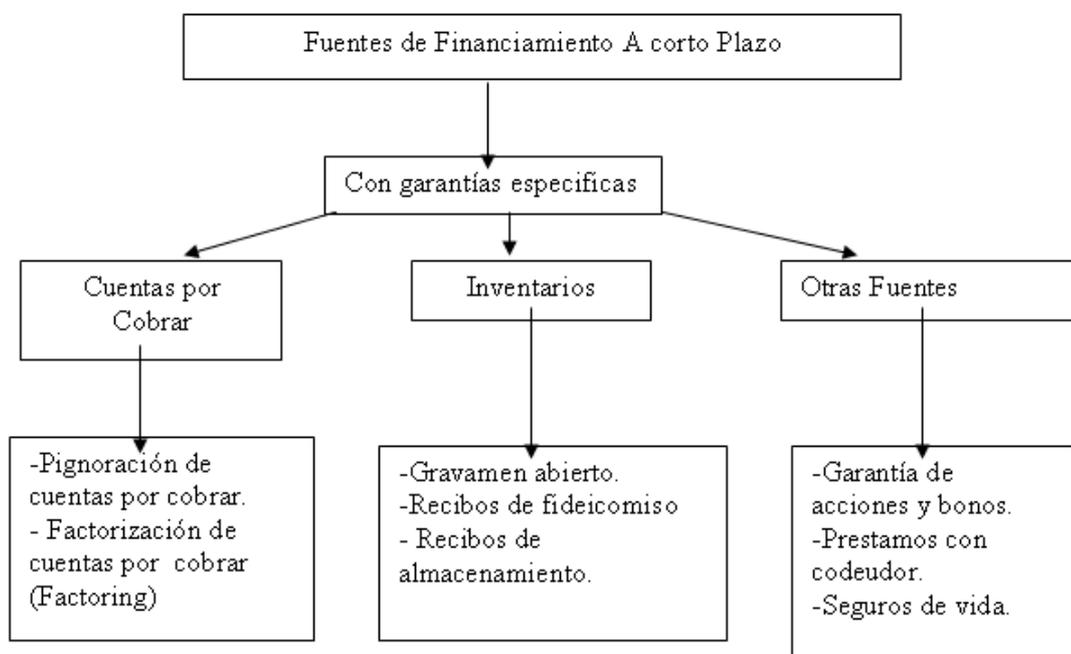


*Figura 11. Fuente de financiamiento a Corto plazo sin garantías*

**Fuentes de Financiamiento con garantías específicas<sup>8</sup>** consiste en que el prestamista exige una garantía colateral que muy comúnmente tiene la forma de un activo tangible tal como cuentas por cobrar o inventario. Además el prestamista obtiene participación de garantía a través de la legalización de un convenio de garantía. Se utilizan normalmente tres tipos principales de participación de garantía en préstamos a corto plazo los cuales son: Gravamen abierto, Recibos de depósito y Préstamos con certificado de depósito. En la

<sup>8</sup> [www.monografias.com/trabajos33/fuentes-financiamiento/fuentes-financiamiento.shtml](http://www.monografias.com/trabajos33/fuentes-financiamiento/fuentes-financiamiento.shtml)

figura 12 se muestran los mas representativos entre los que se incluye el Factoring



*Figura 12. Fuente de financiamiento a Corto plazo con garantías*

### 3.1.2. El Factoring como fuente de Financiamiento para las empresas

Actualmente vivimos un auge económico debido a la buenas perspectivas de inversión que nuestro país viene generando, esto impulsa a su vez el desarrollo de la empresa peruana sobre todo el de las Mypes y Pymes las que enfrentan una problemática común, y es que para las empresas se ha vuelto frecuente enfrentarse a problemas de financiamiento de corto plazo y al pago a tiempo de sus ventas al crédito por el aplazamiento de las fechas de cobro por ejemplo de las facturas emitidas, la falta de liquidez inmediata y la necesidad de conocer permanentemente la solvencia crediticia de sus deudores , lo que constituye un riesgo para lograr el equilibrio financiero de una

compañía, por lo que es necesario buscar fuentes de financiamiento. Uno de estos instrumentos es el llamado Factoring mecanismo que convierte las ventas al crédito como si fueran al contado, evitando que la empresa se enfrente a des-balances por falta de liquidez inmediata.<sup>9</sup>

La empresa entendida como un ente colectivo con un objetivo lucrativo mediante la realización ya sea a través de la entrega de bienes y/o servicios, en el ejercicio de dichas actividades se vincula con otras empresas en calidad de proveedor, recibiendo como contraprestación el pago por el bien recibido o por el servicio prestado. Es en este contexto que sea práctica usual que el pago acordado con su cliente se realice después de transcurrido un determinado plazo, lo que se conoce en el mercado como operaciones comerciales al crédito.

En esta situación, las empresas proveedoras de bienes y/o servicios requieren contar con suficiente capacidad financiera para poder continuar con sus operaciones de prestación de bienes o servicios al crédito hasta que sus clientes vayan cancelando sus obligaciones en los plazos pactados, lo que implica mantener una estructura organizativa de control de los vencimientos y cobros respectivos, existiendo inherentemente la incertidumbre de que llegado el momento para el pago, los clientes no cumplan, con el consiguiente perjuicio en tiempo, dinero y asunción de nuevos costos operativos para recuperar dichas cuentas, lo que a la larga puede implicar una peligrosa situación de iliquidez, de paralización de sus actividades y por último de su probable salida del mercado ante la rapidez y liquidez de la competencia.

---

<sup>9</sup> INFORMATIVO CABALLERO BUSTAMANTE, Vol. 33 Nro. 567 2da Quincena Mayo 2005

Sin embargo, existe la alternativa de que estas mismas empresas proveedoras logren que estos créditos por cobrar les sirvan a su vez para financiar nuevas operaciones. Esta alternativa de financiamiento se configura a través de las operaciones de Factoring. Este mecanismo ha venido consolidándose en las últimas décadas primero en el exterior y luego en nuestro país ya que es útil para empresas de todo tipo de envergadura, y lo es especialmente para las micro y pequeñas empresas, las cuales por no contar con una adecuada infraestructura y medios de cobranza, requieren del apoyo de entidades especializadas que se encarguen de tales menesteres.

### **Definición de Factoring**

El Factoring es la operación mediante la cual una empresa especializada denominada Factor, compra a una persona natural o jurídica, denominada Cliente (o Cedente), instrumentos de contenido crediticio de corto plazo como pueden ser facturas, pagares o títulos valores representativos de deuda, asumiendo el riesgo de cobranza de estos. Es decir, el cliente cede al factor dicha cartera de cobros, con el fin de obtener financiamiento inmediato, pagándole a cambio una comisión. Luego el Factor se encarga de hacer dichos cobros al Deudor (o Aceptante). En algunos casos supone el otorgamiento de servicios adicionales al Cliente, cuando el factor se compromete a ello, a cambio de una retribución.

El Factoring permite a las empresas, obtener liquidez de forma casi inmediata y les evita tener que recurrir a endeudamientos. El Factoring engloba una serie de servicios sobre la base de la cesión de créditos comerciales; en

estos se hallan los servicios administrativo-financiero, como son la gestión integral de los créditos cedidos. Dentro de estos servicios se encuentran:

- El análisis y clasificación de las solicitudes de crédito de futuros clientes.
- La evaluación y clasificación de los niveles de solvencia de los clientes y el establecimiento de los límites de crédito a cada uno de ellos
- La gestión de cobro de las facturas cedidas
- La administración de la cartera de clientes de la empresa, estableciendo el control de los pagos de los deudores y las incidencias de cobro que puedan ocurrir.

La financiación de las ventas mediante el abono anticipado de los créditos comerciales cedidos por el usuario a la compañía de Factoring.<sup>10</sup>

### **Etapas del Factoring**

El proceso del Factoring consta de las siguientes etapas:

- La empresa de Factoring efectúa una evaluación de la situación del Cliente a partir de la información financiera y de créditos que éste le facilita.
- Se aprueba la operación.

---

<sup>10</sup> INFORMATIVO CABALLERO BUSTAMANTE, Vol. 34 Nro.587 2da Quincena Marzo 2006, Vol. 34 Nro.602 1era Quincena Noviembre 2006

- El cliente y la empresa de Factoring firman el contrato. En ese momento el primero cede su cartera de créditos, entregando a la segunda las facturas, facturas conformadas y demás documentos.
- La empresa paga al cliente un porcentaje del valor de los documentos entregados, según lo establecido en el contrato.
- Al vencimiento, la empresa paga el saldo del -valor de los documentos, deduciendo el interés correspondiente.

### **Servicios adicionales que se prestan**

Como se dijo anteriormente, los servicios adicionales que contempla el Factoring, se otorgan sólo cuando así se pacta en el contrato, y a cambio de un pago por ellos. Estos podrían consistir en lo siguiente:

Análisis de la solvencia de los compradores, estableciendo el nivel de riesgo de cada uno, estudios de mercado, investigación de la clientela, asesoría integral, etc.

Cobranza de las facturas cedidas por el Cliente, respetando el medio de cobro pactado por éste con su Deudor (recibo, letra, transferencia, etc.). También pueden incluir servicios contables, información comercial, etc. Administración de las cuentas por cobrar, informando regularmente al Cliente acerca de la situación de cobro de cada una, así como también de cada Deudor. También supone enviar, con la misma regularidad, información a cada Deudor acerca del estado de su cuenta.

Anticipos de fondos: Si bien éstos figuran como un servicio adicional, en realidad constituyen la esencia del factoring, pues son los que dan a esta actividad de financiamiento su atractivo principal. Los anticipos, que permiten disponer de los importes de las facturas cedidas, se otorgan sólo cuando así se pacta por contrato, fijándose un tope de entre 80% y 100% del importe de cada documento. Al constituir préstamos del Factor al Cliente, los anticipos se remuneran con una tasa de interés, también señalada en el contrato.

Asunción del riesgo. Este es otro de los fundamentos del factoring. Cuando así se establece en el contrato, el Factor asumen totalmente el riesgo de insolvencia o no pago del Deudor.

### **Clases de Factoring**

Existen múltiples formas de clasificar al Factoring. La más importante es la que toma en cuenta la disponibilidad financiera y la asunción del riesgo:

- **Factoring sin Recurso o Financiación.** Es aquél en el cual los pagos del Factor al Cliente no contemplan anticipos, sino que se efectúan conforme vencen los documentos. El riesgo de no pago del documento es asumido por el Factor, razón por la cual a veces éste rechaza las cuentas de algunos deudores con problemas. Es la versión más tradicional, por lo cual también es conocido como Factoring Puro. Generalmente va acompañada por servicios contables y administrativos por parte del Factor.

- **Factoring con Recurso o Financiación** Es aquél en el cual el Factor anticipa la financiación, sin esperar al vencimiento de los créditos. El riesgo de no pago de los documentos es asumido por el Cliente, quien al vencimiento de éstos debe devolver al Factor el importe anticipado, así como los intereses pactados y los gastos. Es la versión más moderna del factoring, y la más empleada. Para el Factor, el beneficio está dado por el interés que cobra por los recursos otorgados.

También se puede clasificar el Factoring según el destino de las ventas. De acuerdo a este criterio puede ser Nacional (cuando tanto el Cliente como el Factor y los Deudores residen en el mismo país) o Internacional o de Exportación (cuando el Cliente y los Deudores residen en distintos países).

En el Factoring Internacional o de Exportación existen dos Factores: uno exportador, que reside en el mismo país del Deudor y que cumple las funciones típicas del Factor, cobrando a aquélla deuda pendiente; y otro importador, que coordina con el Factor exportador, y que se halla en contacto con el Cliente.

## **Ventajas y Desventajas<sup>11</sup>**

### **Ventajas**

Algunas de las principales ventajas del Factoring consisten en que permite:

---

<sup>11</sup> INFORMATIVO CABALLERO BUSTAMANTE, Vol. 34 Nro.587 2da Quincena Marzo 2006, Vol. 34 Nro.602 1era Quincena Noviembre 2006

- Mejorar instantáneamente la liquidez de la empresa, al convertir una cartera de cobros en liquidez inmediata, gracias al anticipo recibido, cuando así se pacta en el contrato. Es decir, hace posible vender al crédito y cobrar al contado. En épocas de inflación, esto supone también no quedar expuesto al riesgo de pérdida del poder adquisitivo de los créditos.
- Eliminar el riesgo del no cobro de las facturas en los contratos sin recurso y reducirlo en los contratos con recurso.
- Planificar los flujos financieros y mejorar la gestión financiera y comercial, al eliminar la incertidumbre acerca de las cuentas por cobrar. La empresa sabe con certeza que en una fecha determinada recibirá el importe de los cobros menos la comisión de Factoring.
- Simplificar la contabilidad, porque permite a la empresa vendedora convertir su cartera de múltiples clientes en una cuenta con un solo cliente, que además le cancela de inmediato (si así se pacta).
- Ahorrar gastos de personal, gastos administrativos y costos de oficina por concepto de seguimiento de las cuentas por cobrar, facturación, contabilidad y por eventuales procesos judiciales de cobranza, ya que todos ellos pasan a ser responsabilidad del Factor.
- Reducir los gastos financieros derivados de retrasos en los cobros.
- Contar con una institución intermediaria como el Factor, que conoce bien el mercado y las condiciones financieras de los deudores, conocimientos que puede poner al servicio del Cliente.

- Eliminar la incertidumbre del Cliente, permitiendo a éste mejorar su situación financiera y otorgar mejores condiciones a sus propios clientes, incrementando sus ventas.

### **Desventajas**

- Entre las principales desventajas del Factoring se puede mencionar las siguientes:
- Tiene un alto costo, pues la tasa de interés que se aplica suele ser mayor a la de descuentos. Sin embargo, esta desventaja se compensa por el ahorro que se produce al ya no tener que incurrir en los gastos de cobranza;
- No considera transacciones efectuadas con productos perecederos;
- Sólo se aplica en operaciones de corto plazo;
- El cliente queda sujeto al particular criterio de evaluación de riesgo que pueda tener el Factor;
- No suele aplicarse a las empresas en dificultades financieras, aunque éstas sean coyunturales;
- Al ser una operación interpersonal, pues la empresa de factoring no conoce al deudor como sí lo hacía su acreedor, otorga mucho menos flexibilidad a aquél a la hora de presentar eventuales inconvenientes con los pagos;
- El factor sólo compra las cuentas por cobrar que le parecen de mayor calidad, en cuanto a plazo, monto y posibilidad de cobro.

## **Derechos y Obligaciones**

### **Derechos y Obligaciones del Factor**

El factor-tiene derecho, como mínimo, a lo siguiente:

- Realizar todos los actos de disposición con relación a los instrumentos de contenido crediticio adquiridos;
- Cobrar una retribución por los servicios adicionales que haya brindado.

En tanto que está obligado, por lo menos, a:

- Adquirir los instrumentos de acuerdo a las condiciones pactadas;
- Brindar los servicios adicionales a los que se comprometió;
- Pagar al cliente por los instrumentos adquiridos;
- Asumir el riesgo crediticio de los deudores de los instrumentos.

### **Derechos y Obligaciones del Cliente** <sup>12</sup>

El cliente tiene, por lo menos, los siguientes derechos:

- Exigir el pago por los instrumentos con contenido crediticio transferidos, en el plazo y bajo las condiciones pactadas;

---

<sup>12</sup> INFORMATIVO CABALLERO BUSTAMANTE, Vol. 34 Nro.602 1era Quincena Noviembre 2006

- Exigir el cumplimiento de otros servicios adicionales que se hubieren pactado.

En tanto que se sujeta, por lo menos, a las siguientes obligaciones:

- Garantizar la existencia, exigibilidad y vigencia de los instrumentos al tiempo de celebrarse el Factoring;
- Transferir al Factor los instrumentos según lo acordado o lo establecido por Ley;
- Notificar a los deudores de los instrumentos acerca de la realización del Factoring, cuando sea el caso;
- Recibir los pagos efectuados por los deudores de los instrumentos y transferírseles al Factor, cuando así lo haya acordado con éste;
- Informar al Factor y cooperar con éste para permitir la mejor evaluación de su propia situación patrimonial y comercial, así como la de sus deudores;
- Proporcionar toda la documentación vinculada con la transferencia de los instrumentos;
- Retribuir al Factor por los servicios adicionales recibidos.

La operación de Factoring debe realizarse con conocimiento de los deudores, a menos que, por la naturaleza de los instrumentos adquiridos, ello no resulte necesario.

Se presume que éstos conocen del Factoring cuando se tiene evidencia de la recepción de la notificación correspondiente en sus domicilios legales o en aquéllos señalados en los Instrumentos, o cuando mediante cualquier otra forma se evidencie indubitadamente este hecho.

### **Empresas autorizadas a actuar como factores**

Las empresas facultadas para operar como Factores son:

1. Las empresas de Factoring que hayan sido autorizadas a realizar tal función por la SBS;
2. Las empresas bancarias y otras entidades de operaciones múltiples que cumplen los requisitos establecidos en el Módulo 1 del Artículo 290° de la Ley N° 26702. Estas últimas son las empresas financieras, cajas municipales de ahorro y crédito, cajas rurales de ahorro y crédito y EDPYMES.

A continuación presentamos un análisis usando una matriz FODA: Fortalezas, Oportunidades, Debilidades y Amenazas concerniente al Factoring como producto crediticio.

<b>FORTALEZAS</b>	<b>DEBILIDADES</b>
Financiamiento de cuentas a cobrar como garantía.	Heterogeneidad en la tasa de interés

Utilización por empresas que necesitan liquidez y no tienen garantías suficientes.	Adicionalmente a la tasa de interés, se agregan otros costos y comisiones
Puede acompañar las necesidades de crecimiento de las empresas.	No acepta algunos documentos
Ahorro de tiempo y menores gastos de cobranza y contabilidad.	Sujeto a la evaluación del riesgo de los compradores
Financia por flujo de caja y no por estado patrimonial.	Implica abrir el acceso a la información de la empresa lo que se agrava en el caso del ámbito rural
Utiliza instrumentos ejecutivos de distinto tipo (Factura de Crédito, Pagare)	Excluye a los productos de tipo perecederos
Distintas modalidades de uso (con recurso, sin recurso)	Normalmente para proveedores de empresas bien posicionadas en el mercado
Permite obtener financiamiento en corto plazo	
Estabiliza el flujo de fondos.	
Movilización de la cartera de deudores.	
Transferencia del riesgo de	

insolvencia	
-------------	--

<b>OPORTUNIDADES</b>	<b>AMENAZAS</b>
Corte de la cadena de créditos institucional y disminución de la extra-bancaria	Inestabilidad económica y financiero.
Permite articulación entre exportadores e importadores	Entorno recesivo y caída de operaciones comerciales
Escaso desarrollo	Desconocimiento y desconfianza de los usuarios
Buena alternativa en la actualidad para pre-financiar exportaciones.	“Lobby” de las grandes empresas
Alternativa de financiamiento para el sector.	Alta evasión impositiva
Entorno agro exportador	
Desconfianza en el sector bancario	

*Cuadro 10. Análisis FODA del Factoring*

### 3.1.3. Aspectos operativos del Factoring

Podemos distinguir una fase inicial de contacto factor – cliente, en la que este último facilita al factor los datos mínimos necesarios para que pueda

pasarle una oferta. Y una segunda fase de aceptación de la oferta que enlaza con una tercera, previa a la firma del contrato, en la que el factor solicita datos concretas sobre el cliente y sus deudores, a fin de efectuar un estudio consistente en analizar:

- El sector económico en que se desenvuelve el cliente.
- Su calificación dentro del sector.
- Su solvencia moral y su capacidad técnica y profesional.
- La evolución de sus ventas en los últimos años.
- La solvencia y comportamiento de pago de los deudores, a fin de evaluarlos y fijarles líneas de riesgo.

Una vez que el factor, a la vista de los datos recogidos y de los estudios realizados, ha dado su conformidad a la firma del contrato y se ha fijado el límite máximo de pagos parciales a anticipar al cedente y las líneas de riesgo para cada deudor, se procede a la materialización de dicha firma, estando ya en condiciones de empezar a operar.<sup>13</sup>

En el momento de entregar los ejemplares de los documentos de crédito (factura) al factor, deberá manifestar si desea que le anticipe algún pago parcial y en qué cuantía.

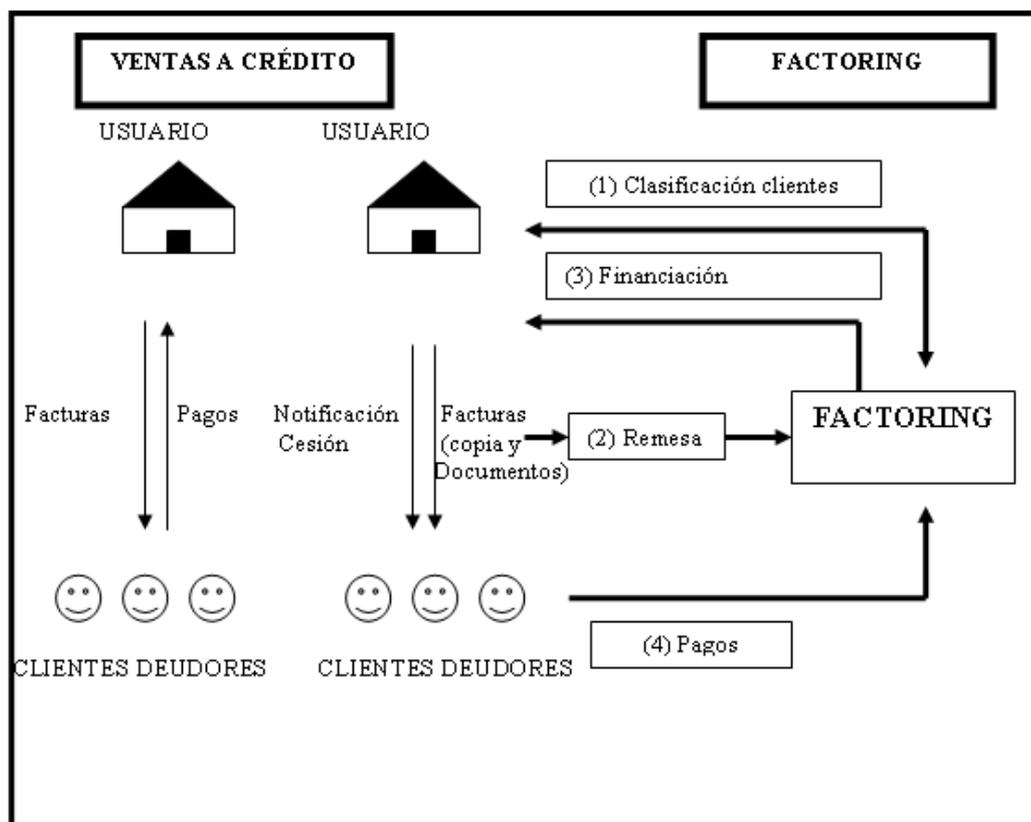
Una vez que se han hecho las correcciones y validaciones necesarias, se emite un documento de aceptación de remesa, la correspondiente factura

---

<sup>13</sup> [www.monografias.com/trabajos46/factoring/factoring2.shtml](http://www.monografias.com/trabajos46/factoring/factoring2.shtml)

por la tarifa del factoraje e intereses y la carta orden al banco, en este último caso, para que abone en la cuenta del Cliente el citado pago.

A partir de dicho momento el factor se hace cargo de la gestión y administración de los documentos de crédito (factura) recibidos, correspondientes a los créditos que le han sido cedidos, y en función de la modalidad de Factoring contratada prestará sus servicios. Resumimos el flujo de una operación de Factoring en la figura 13



*Figura 13. Flujo operativo simplificado del Factoring.*

En el Factoring de exportación, igual que en el Factoring nacional, el factor exportador (con la colaboración de su corresponsal, el factor importador),

se hace cargo de la gestión y administración de los documentos de crédito (facturas) cedidos y en función de la modalidad de Factoring contratado.

#### 3.1.4. El Factoring internacional

El Factoring Internacional se presenta de dos formas en el comercio internacional:

- Factoring “de exportación”, en el que las empresas de factoring y la factorada domicilian en un mismo país, mientras que los deudores cedidos lo hacen en el extranjero. José Benito Fajre menciona que “este mecanismo es muy parecido al del factoring interno, sólo que los créditos recién podrán ofrecerse cuando la mercadería ya haya sido enviada al comprador foráneo, por lo que debe presentarse el documento demostrativo de su despacho, por ejemplo, la carta de porte, el conocimiento de embarque, etc.”
- Factoring “de importación”, que se da cuando el cliente o factorado reside en un país distinto al del factor, por lo que decide transmitirle todos los créditos originados en su actividad empresarial que tiene contra deudores domiciliados en el país del factor.

Estos subtipos de factoring, que trascienden las fronteras, tienen gran importancia en el comercio internacional, entre otras razones, porque facilitan

las operaciones de importación y exportación, y porque son mecanismos muy eficaces y económicos para el conocimiento de los mercados extranjeros.

En la operación de Factoring Internacional están involucrados 4 integrantes fundamentales:<sup>14</sup>

- **Cedente o Cliente:** es el exportador que vende (cede) al factor los créditos comerciales que se han generado en el desarrollo de su actividad exportadora.
- **Factor de exportación o cesionario:** es la entidad del país exportador, que compra los créditos comerciales originados en las ventas internacionales de su cliente (exportador) y, simultáneamente, presta los servicios adicionales relacionados con estos créditos.
- **Factor de importación o corresponsal:** es la entidad de país importador, que actúa como corresponsal del factor de exportación para realizar las funciones necesarias en el país importador. Se relaciona con el factor de exportación mediante un acuerdo de mutua colaboración.
- **Deudor:** es el comprador extranjero del exportador y, por tanto, está obligado al pago de los créditos comerciales objeto de la cesión (por lo general, facturas).

---

<sup>14</sup> [www.gestiopolis.com/Canales4/fin/factoraje.htm](http://www.gestiopolis.com/Canales4/fin/factoraje.htm)

A continuación en la figura 14, se muestra una secuencia de pasos de Factoring a nivel Internacional:

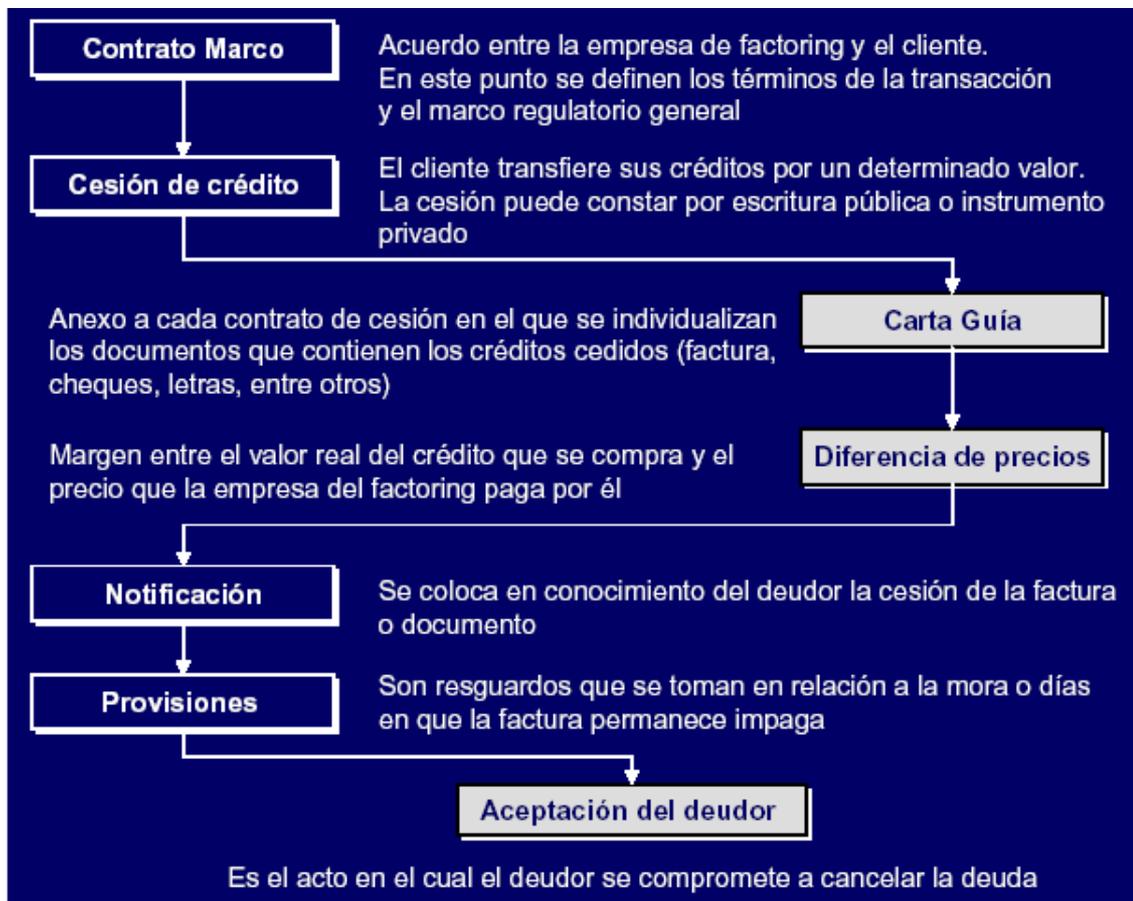


Figura 14. Flujo operativo del Factoring detallado.

## La experiencia Chilena <sup>15</sup>

### Antecedentes Generales de Chile

- N° de habitantes: 16 millones.
- Indicadores Económicos:
  - PIB: US\$ MM 113.054
  - Crecimiento: 4,0%

<sup>15</sup> GONZÁLEZ B., FABIOLA, RODRÍGUEZ S., LUIS. Tesis "El Factoring en Chile", USACH, 1996.

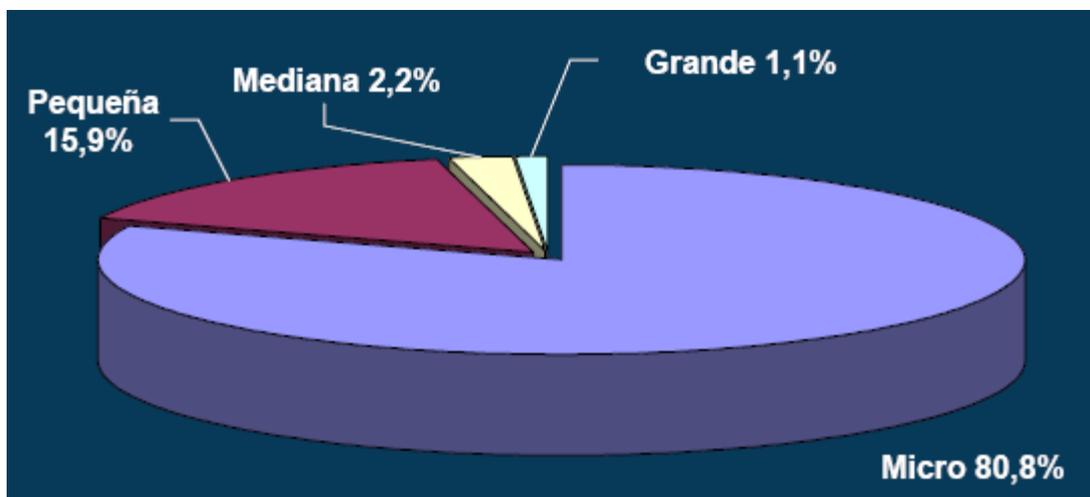
- Inflación: 2,6 %
- Tasa de desocupación: ~6%

### *Estructura Productiva de Chile*

- Norte: Minería, Pesca, Agricultura
- Centro: Industria, Servicios, Agricultura, Ganadería
- Sur: Forestal, Pesca, Agricultura, Minería

### *Número de Empresas Pymes*

En el medio chileno se puede observar que una gran cantidad de empresas se ubican en el sector micro, siendo casi el 80% del total, en la figura 15 observamos la distribución por tamaño de empresa.



*Figura 15. Distribución porcentual de Pequeñas y Medianas empresas.*

### *Factoring en Chile - Reseña Histórica*

La experiencia chilena en lo que se refiere al Factoring se remonta a 1989 cuando al legislación dio posibilidad de instituir las primeras empresas de

Factoring en el país, pero inicialmente con un alcance muy limitado esto por medio de subsidiarias durante la década de los 90. Observamos la evolución en la figura 16.

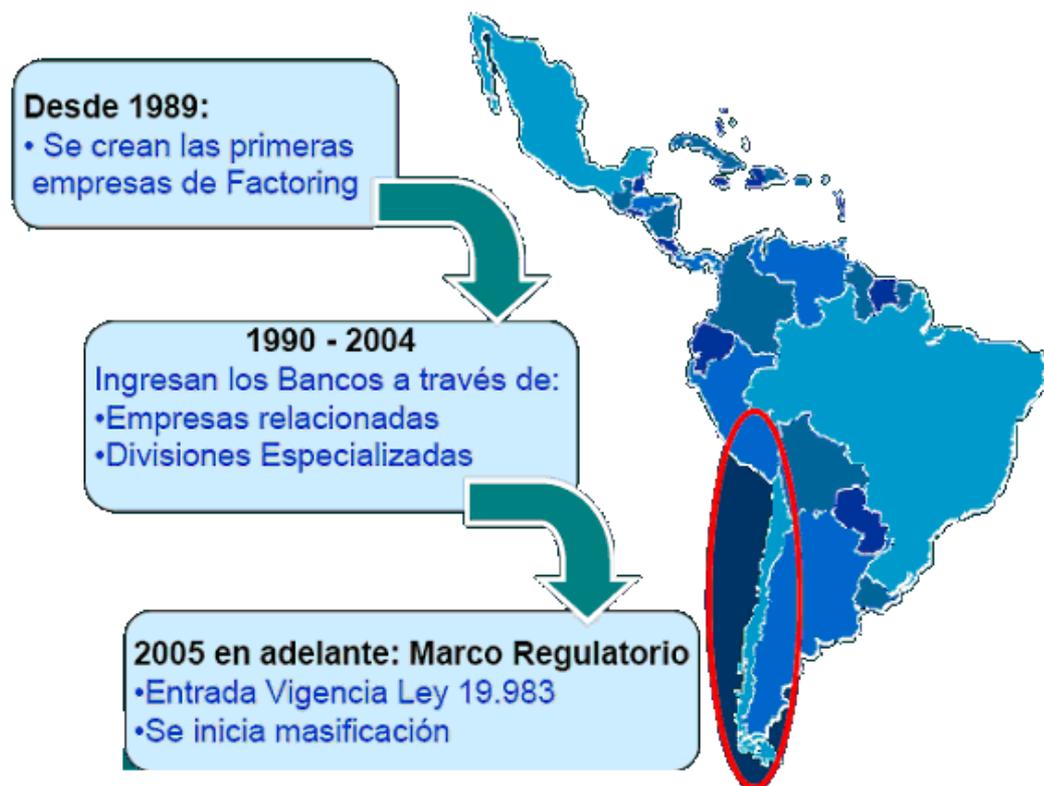


Figura 16. Evolución del Factoring en Chile.

- *Aspectos Jurídicos del Factoring en Chile*
  - Ley No 19.983, regula la transferencia y otorga merito ejecutivo a la copia de la factura
  - Vigencia a partir de Abril de 2005
  - Establece marco regulatorio para la actividad del Factoring
  
- *Limitaciones*

- Siempre existe un desequilibrio de poderes entre cedentes y grandes deudores.
  - El cedente no quiere afectar las relaciones comerciales con su deudor.
  - Los Factoring deben seguir enfocando la evaluación crediticia hacia el cliente.
  - En Chile el 95% de las operaciones son con responsabilidad del cedente.
- *Barrera que encontramos en los Pagadores*

Los deudores o pagadores obstaculizan el Factoring por:

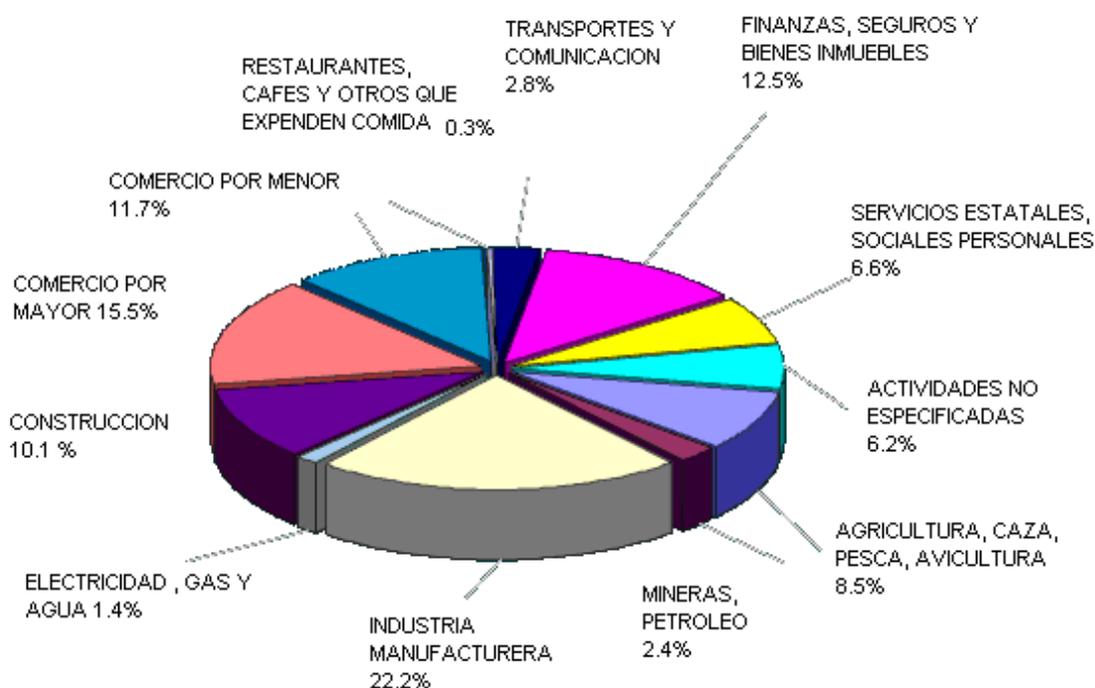
    - Les interesa el negocio (pronto pago).
    - Quieren seguir ejerciendo control sobre sus proveedores.
    - Evitan nuevas cargas administrativas.
- *Porque ha crecido el Factoring en Chile*
    - Por la activa participación de la Banca nacional y extranjera en el negocio.
    - Por el marco regulatorio establecido con la Ley 19.983.
    - Por la constante difusión y por ende mayor conocimiento del producto.
    - Por la gran cantidad de empresas oferentes del producto.
    - Por la clara necesidad de capital de trabajo en el crecimiento de la economía.
    - Por las restricciones normativas de la Banca, desaparece la intermediación.

### *Actividades Económicas que usan Factoring<sup>16</sup>*

---

<sup>16</sup> [www.ugm.cl/ugmnews/ppt/pvaras/rcarvallo.ppt](http://www.ugm.cl/ugmnews/ppt/pvaras/rcarvallo.ppt)

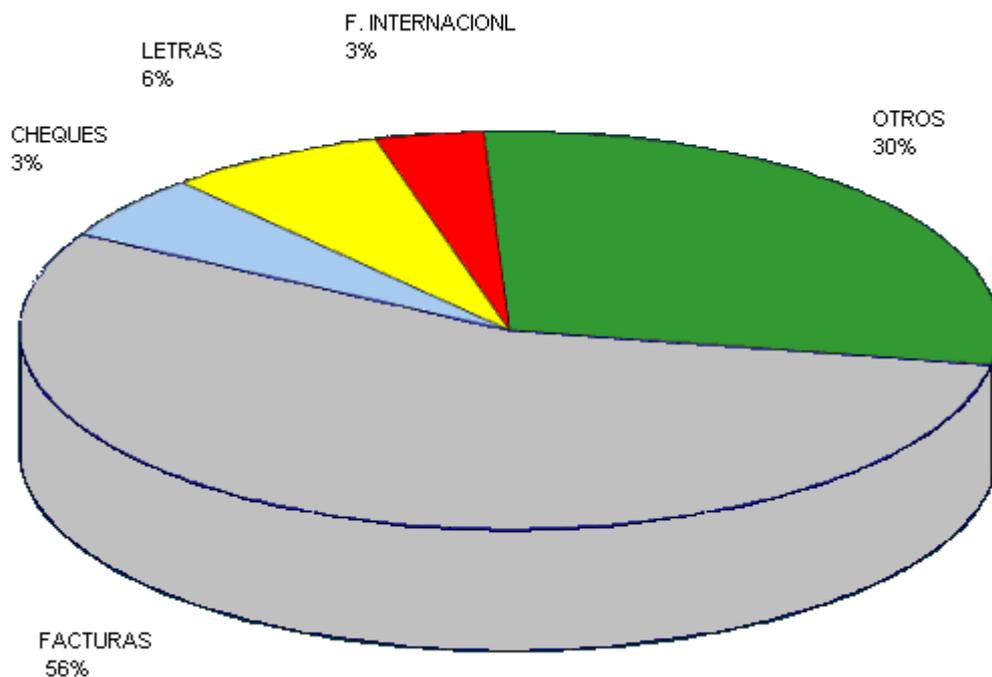
En el vecino país del sur, el Factoring es un producto crediticio muy difundido y en parte ha contribuido al desarrollo de las empresas que en su momento requirieron de financiamiento para lograr sus objetivos de negocio. Como podemos ver en la figura 17, estas son muy variadas yendo desde diversos sectores como el de manufactura, construcción, agricultura etc.



*Figura 17. Actividades económicas que usan el Factoring.*

### *Tipos de Documentos adquiridos por Factoring*

Asimismo en lo que respecta a los documentos valorados mas empleados en las operaciones de Factoring encontramos que son las Facturas comerciales las más empleadas tal como podemos ver en la figura 18.



*Figura 18. Tipos de Valores involucrados en el Factoring.*

**Servicios:**

- Cobertura de Riesgo financiero de clientes en el extranjero
- Anticipar flujos de facturas de Exportación
- Cobranza Internacional
- Anticipos de flujos de Cartas de Crédito y Cobranza de Exportación.
- Evaluación Crediticia de clientes extranjeros
- Financiamiento a Importadores

*Asociación Chilena de Factoring<sup>17</sup>*

---

<sup>17</sup> [www.achef.cl](http://www.achef.cl)

Asociación gremial fundada en 1994 donde se agrupan trece empresas relacionadas con los principales bancos chilenos y/o extranjeros. Su objetivo principal es divulgar la práctica del Factoring y su interés en el desarrollo de la economía del país y de las empresas. Entre las empresas Asociadas tenemos:

- BCI Factoring
- Banchile Factoring
- Corp Banca
- Bice Factoring
- Bandesarrollo Factoring
- Citibank
- Incofin
- Factoring Security
- Santander Santiago Factoring
- Scotiabank
- Bank Boston
- Banco HNS

#### 3.1.5. Introducción a SOA

A lo largo de los años, en nuestras empresas se han ido acumulando una gran cantidad de aplicaciones distintas que se fueron desarrollando para tratar de resolver las distintas necesidades que iban surgiendo: ERPs, CRMs, Bases de datos, Mainframes, Sistemas CICS junto con aplicaciones WEB / J2EE, .NET, etc. Pero estas aplicaciones no se crearon con la idea de interactuar entre ellas, sino como herramientas para resolver un problema en

un momento dado. En definitiva, el paisaje que nos encontramos es parecido a un archipiélago de aplicaciones<sup>18</sup>.

La realidad del mercado nos ha llevado a las siguientes conclusiones:

- Nuestras aplicaciones internas están de alguna manera condenadas a entenderse, si queremos dar una respuesta ágil, a las necesidades de negocio que surgen.
- Las aplicaciones de nuestra empresa están también condenadas a entenderse de alguna manera, con las aplicaciones de las empresas con las que queremos cooperar para ofrecer mejor servicio a nuestros clientes.

Es, en este ámbito, donde SOA pretende hacer su aparición.

La palabra arquitectura en el mundo del software se podría definir como un conjunto de decisiones que hemos de tomar a la hora de organizar nuestras aplicaciones, como van a interactuar entre ellas, qué software se va a usar a la hora de comunicar entre ellas, protocolos, qué plataformas, máquinas, sistemas operativos, lenguajes de programación, qué tipo de mensajes se van a intercambiar, etc. Las decisiones que tomamos a la hora de decidir nuestra arquitectura son fundamentales, no tanto a corto plazo, sino más bien a largo plazo, y pueden ser a veces una trampa mortal.

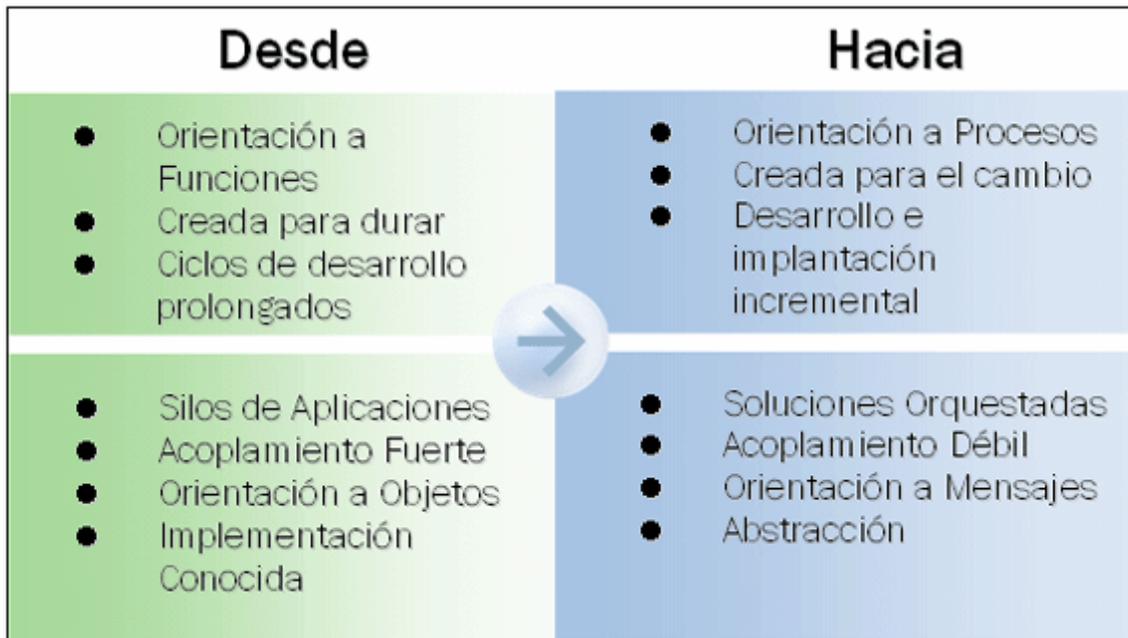
La adopción de una arquitectura basada en servicios es actualmente una necesidad para la mayoría de las empresas. Esta necesidad no se justifica por una moda pasajera sino por la necesidad de lograr crear una infraestructura

---

<sup>18</sup> [www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SOA2](http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SOA2)

sólida que soporte la integración de los sistemas críticos de la empresa. Además, disponer de una infraestructura SOA también es un requerimiento para que una empresa pueda lograr una implementación exitosa de sistemas que permita alcanzar las metas que propone la teoría del BPM (Business Process Management, *Administración de Procesos de Negocio*), lo que permite responder más rápido a las necesidades del negocio.

Las condiciones del mercado son más dinámicas que nunca. Esto significa que las organizaciones deben incrementar sus oportunidades lo más rápido posible y encontrar nuevas vías para satisfacer las cada vez más exigentes expectativas de los clientes. Más servicios, mejor adecuación, flexibilidad, mayores prestaciones en línea, tiempos de respuesta más cortos, fiabilidad, precisión. La lista de las necesidades de los clientes es interminable. Es importante saber si la organización puede dar respuesta a estas demandas y, en tal caso, conocer el camino que le permita distanciarse de la competencia. Una Arquitectura Orientada a Servicios (SOA) es la clave para dar respuesta a estas necesidades tanto ahora como en el futuro. En la figura 19 podemos graficar cuales son las principales diferencia entre las arquitecturas clásicas y la orientación a servicios.



*Figura 19. Evolución desde una arquitectura clásica hacia una arquitectura SOA*

El número de empresas en las que la arquitectura SOA permite optimizar y agilizar los procesos de negocio es cada vez mayor. Al organizar las funciones de las aplicaciones centrales de su organización en servicios que comparten información y se basan en estándares, se conseguirán unos procesos mucho más eficientes y orientados al cliente.<sup>19</sup>

Sin embargo, adoptar SOA no es una tarea sencilla, especialmente para aquellas organizaciones que no están acostumbradas a desarrollar sistemas distribuidos. Las principales compañías acometen la complejidad de sus aplicaciones y entornos informáticos con la arquitectura orientada a servicios (SOA), que ayuda a desarrollar servicios empresariales modulares fáciles de

<sup>19</sup> [huibert-aalbers.com/IT\\_Insight/Spanish/WhitePapers/ITI007Sp-SuccessfulSOAImplementation.pdf](http://huibert-aalbers.com/IT_Insight/Spanish/WhitePapers/ITI007Sp-SuccessfulSOAImplementation.pdf)

integrar y reutilizar, construyendo una infraestructura informática verdaderamente flexible y adaptable. Con un enfoque de SOA, su organización informática dedicará más recursos y presupuesto a innovar y a prestar nuevos servicios empresariales.

### 3.1.6. Definición de SOA

La mejor manera de comenzar a explicar SOA, es explicar qué NO es:<sup>20</sup>

- *SOA no es un software*, no es un MOM (Middleware orientado a Mensajes), no es una EAI (Aplicación de integración empresarial), aunque una arquitectura SOA puede apoyarse en un conjunto de herramientas como MOMs o EAls para conseguir su objetivo.
- *SOA no es una metodología de proyecto*, aunque a la hora de iniciar un proyecto orientado a conseguir una arquitectura SOA en una empresa, algunas metodologías se ajustan mejor que otras: es preferible seguir un modelo en iteraciones como las metodologías ágiles, que seguir metodologías Waterfall o Cascada
- *SOA no es otra forma de llamar a los Servicios Web*, aunque los Servicios Web son una herramienta válida para conseguir una arquitectura SOA, Pero, ¿Se puede implementar SOA sin usar Servicios Web? En teoría sí, pero en la práctica no. La gran ventaja de SOA es poder construir una misma arquitectura que incorpore tecnología y productos de diferentes proveedores, y por tanto la clave es la interoperabilidad. Construir un arquitectura software distribuida en n-

---

<sup>20</sup> [www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SOA2](http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SOA2)

capas y descompuesta en servicios débilmente acoplados con tecnología de un sólo fabricante no sirve de mucho, en realidad es un sobreesfuerzo para conseguir lo mismo que con una arquitectura tradicional. El beneficio que suele haber radica en reutilización y mantenimiento de las aplicaciones, pero desde luego muchas veces no compensa. La clave en las aplicaciones compuestas que reutilizan todos los servicios de una organización es la interoperabilidad y esta se consigue con los estándares conocidos como Servicios Web. Inicialmente existían 3: SOAP, WSDL y UDDI, pero a día de hoy la cosa se ha complicado, ya que las arquitecturas SOA corporativas necesitan de muchos más para ser interoperables a todos los niveles. Un caso habitual es la interoperabilidad entre aplicaciones JAVA y .NET usando una misma arquitectura SOA

La Arquitectura Orientada a Servicios (SOA) es un modelo que interrelaciona unidades funcionales diferentes de una aplicación, denominado servicios, a través de interfaces y contratos bien definidos entre estos servicios. La interfaz se define de una manera neutral que debe ser independiente de la plataforma de hardware, del sistema operativo y del lenguaje de programación en los que se implemente el servicio. Esto permite que los servicios, contruidos en una variedad de tales sistemas, interactúen entre sí de una manera uniforme y universal.

Esta característica de tener una definición de interfaz neutral que no esté fuertemente sujeta a una implementación en particular se conoce como “*loose*

*coupling*” o acoplamiento débil entre los servicios. El beneficio de un sistema débilmente acoplado es su agilidad y capacidad para sobrevivir a cambios evolutivos en la estructura e implementación de las partes internas de cada servicio que compone toda la aplicación. Por otra parte, *tight-coupling* significa que las interfaces entre los diferentes componentes de una aplicación están estrechamente interrelacionados en función y forma, haciendo que sean frágiles cuando se requiera alguna forma de cambio en partes de la aplicación o en toda la misma.

La necesidad de sistemas débilmente acoplados surgió asimismo de la necesidad de que las aplicaciones de negocios se tornaran más ágiles sobre la base de las necesidades de la empresa de adaptarse a su entorno cambiante, tales como políticas cambiantes, fuerzas de negocios, enfoques de negocios, asociaciones, posición en la industria y otros factores relacionados con los negocios que tienen influencia en la misma naturaleza de los mismos. Una empresa que puede actuar flexiblemente con respecto a su entorno como una empresa “*On Demand*”, es decir bajo demanda, donde hay cambios en la forma en que se realizan las cosas o el trabajo según sea necesario.

La arquitectura orientada a los servicios no es nueva, pero es un modelo alternativo con respecto a los modelos orientados a los objetos estrechamente acoplados de una manera más tradicional que han emergido en las últimas décadas. Al tiempo que los sistemas basados en SOA no excluyen el hecho de que se puedan construir servicios individuales con diseños orientados a los objetos, el diseño total está orientado a los servicios. Dado que permite objetos dentro del sistema, SOA puede estar basado en objetos, pero no está, en su totalidad, orientado a los objetos. La diferencia está en las interfaces

propiamente dichas. Un ejemplo clásico de un sistema proto-SOA que ha estado alrededor por un tiempo es *Common Object Request Broker Architecture (CORBA)*, que define conceptos similares a SOA<sup>21</sup>.

No obstante, el SOA actual es diferente en cuanto a que confía en un avance más reciente basado en *el eXtensible Markup Language (XML)*. Al describir interfaces en un lenguaje basado en XML denominado *Web Services Definition Language (WSDL – Lenguaje de Definición de Servicios Web)*, los servicios se han movido hacia un sistema de interfaz más dinámico y flexible que el *Interface Definition Language (IDL – Lenguaje de Definición de Interfaces)* anterior que se encontraba en CORBA.

### 3.1.7. Mitos y Verdades sobre SOA<sup>22</sup>

- Mitos
  - SOA es una Tecnología
  - SOA es nuevo y revolucionario
  - SOA asegura que el negocio y IT trabajen juntos
  - SOA requiere de muchos consultores
  - Necesitamos construir SOA
- Verdades <sup>23</sup>

---

<sup>21</sup> [www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

<sup>22</sup> [http://www.worktec.com.ar/agsi2006/encuentro/SOA\\_Vision\\_de\\_Accenture.ppt](http://www.worktec.com.ar/agsi2006/encuentro/SOA_Vision_de_Accenture.ppt)

<sup>23</sup> <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=SOA2>

- SOA es una filosofía de diseño independiente de cualquier producto, tecnología o marca de mercado
- SOA puede ser realizado vía Servicios Web, pero usando solo Servicios Web no necesariamente implica SOA
- SOA no es una metodología
- SOA debe ser incremental y desarrollado sobre las inversiones actuales
- SOA es un medio, no un fin

### 3.1.8. Servicios Web

La adopción de una solución de diseño basada en SOA no exige implantar Servicios Web. No obstante, como ya comentamos anteriormente, los servicios Web son la forma más habitual de implementar SOA. Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Los servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con socios de negocios.<sup>24</sup>

Los Servicios Web se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP (*Simple Object Access Protocol*) para el intercambio de datos y el lenguaje WSDL (*Web Services Description Language*) para describir las funcionalidades de un

---

<sup>24</sup> [download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real\\_World\\_SOA.pdf](http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf)

servicio Web. Existen más especificaciones, a las que se denomina genéricamente como la arquitectura WS-\*, que definen distintas funcionalidades para el descubrimiento de servicios Web, gestión de eventos, archivos adjuntos, seguridad, gestión y fiabilidad en el intercambio de mensajes y transacciones.

¿Qué es un servicio exactamente? Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella. Desde la perspectiva de la empresa, un servicio realiza una tarea concreta: puede corresponder a un proceso de negocio tan sencillo como introducir o extraer un dato como “Código del Cliente”. Pero también los servicios pueden acoplarse dentro de una aplicación completa que proporcione servicios de alto nivel, con un grado de complejidad muy superior –por ejemplo, “introducir datos de un pedido”-, un proceso que, desde que comienza hasta que termina, puede involucrar varias aplicaciones de negocio.

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos de tecnología de información, a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse y re combinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el seno de una organización. Las aplicaciones compuestas (también llamadas “dinámicas”) son lo que permite a

las empresas mejorar y automatizar sus procesos manuales, disponer de una visión consistente de sus clientes y socios comerciales y orquestar sus procesos de negocio para que cumplan con las regulaciones legales y políticas internas. El resultado final es que las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva.

### 3.1.9. Tecnologías componentes de SOA<sup>25</sup>

En sí mismo SOA es un concepto abstracto sobre cómo se debe unir el software. Confía en las ideas y tecnologías más concretas implementadas en XML y en servicios de Web para existir en la forma de software. Asimismo, para funcionar con efectividad, también requiere soporte de seguridad, administración de políticas y una mensajería confiable. Esto puede ser mejorado aún más mediante la adición del procesamiento transaccional distribuido y la administración de estado del software distribuido.

La distinción entre los servicios de SOA y los servicios de Web reside en el diseño. El concepto de SOA no define exactamente cómo deberán interactuar específicamente, sólo cómo los servicios pueden comprenderse entre sí y cómo pueden interactuar. Es la diferencia entre definir una estrategia sobre cómo realizar un proceso, y la táctica sobre cómo realmente se hace. Por otra parte, los servicios de Web tienen pautas sobre cómo la mensajería entre los servicios necesita interactuar; es decir, es la implementación táctica de un

---

<sup>25</sup> [http://www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

modelo SOA que se ve más comúnmente en los mensajes de *SOAP* entregados sobre *HTTP*. De este modo, esencialmente los servicios de Web son un subconjunto específico acerca de cómo se puede implementar SOA.

Los Servicios Web son sólo un método para implementar SOA. Otros protocolos que también implementan directamente interfaces de servicio con WSDL y se comunican con mensajes de XML pueden estar asimismo comprendidos en SOA. Según lo investigado, ahora CORBA y los sistemas IBM MQ también pueden participar en SOA usando nuevos dispositivos que funcionan con WSDL. Si dos servicios necesitan intercambiar datos, también necesitan usar el mismo protocolo de mensajería pero las interfaces de datos permiten los mismos intercambios de información.

Para establecer un control apropiado de todo este intercambio de mensajes, así como también para aplicar las necesidades de seguridad, política y confiabilidad, hay un nuevo objeto de software que entra en la escena de SOA. Es el *Enterprise Service Bus (ESB)*, que es responsable del control y del flujo apropiado, y tal vez también de las conversiones de todos los mensajes entre los servicios, usando cualquier cantidad de protocolos de mensajería posibles. No se requiere absolutamente el ESB, sobre todo cuando el proyecto es de envergadura mediana y aun no se tiene un amplio portafolio de servicios, pero es un componente vital para administrar apropiadamente sus procesos de negocios en SOA. El ESB propiamente dicho puede ser un único motor o aun un sistema distribuido compuesto por muchos ESBs, todos funcionando juntos para mantener operativo el sistema SOA. Conceptualmente, ha evolucionado del mecanismo “store-and-forward” que se

encontraba en los conceptos anteriores de la ciencia de la computación, tales como Message Queue y la computación transaccional distribuida.

Con respecto al desarrollador, las herramientas que usa necesitan conocer las capacidades de SOA y permitirle trabajar efectivamente con objetos de SOA. Esto incluye el proceso de diseñar el modelo, desarrollando servicios y objetos de servicio, y probar la aplicación de SOA en su totalidad. De este modo, las herramientas del desarrollador deberán estar listas para *Service-Oriented Application Design and Development (SOAD)* es decir Desarrollo de Aplicaciones Orientadas a Servicios, ejemplos de esto es la plataforma de desarrollo .NET de Microsoft.

### 3.1.10. Relación de SOA con otras tecnologías<sup>26</sup>

SOA puede interactuar con una cantidad de otras tecnologías, pero con respecto a esto el encapsulado y el agregado de componentes tienen un rol significativo. Tal como se indicó anteriormente, un servicio de SOA puede ser un objeto simple, un objeto complejo, una colección de objetos, un proceso que contiene otros procesos, y asimismo una colección entera de aplicaciones que dan un único resultado. Fuera del servicio se lo ve como una única entidad, pero dentro de sí mismo puede tener *cualquier nivel de complejidad que sea necesario*.

SOA no es específico en lenguaje, excepto tal vez con respecto a XML y WSDL. Los servicios se pueden implementar en cualquier lenguaje de

---

<sup>26</sup> [http://www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

programación siempre y cuando pueda generar e interactuar con WSDL. SOAP en sí mismo no es un requisito absoluto, pero es un sistema de mensajería común. De este modo, los servicios miembros en SOA pueden ser implementados en casi cualquier variedad de lenguaje de programación y plataforma que dé soporte a WSDL.

Una aplicación basada en CORBA tiene muchos de los componentes necesarios para realizar interfaz con SOA. Si bien el Interface Description Language (IDL) en CORBA es conceptualmente similar a WSDL, no es exacto y por tanto necesita ser primero correlacionado con WSDL. Asimismo, los protocolos SOA de más alto nivel, tal como para la administración de procesos y políticas, necesitan ser usados en vez de los conceptos similares de CORBA. Debe tenerse en cuenta que esto es solo cuando un componente de CORBA (representado como un servicio) necesite interactuar con un servicio de SOA; dentro del modelo CORBA, todos los componentes de subconjuntos individuales pueden seguir funcionando como antes.

Los servicios de SOA son independientes de lenguaje de programación, pero los lenguajes Java y los nuevos de .NET están entre los principales lenguajes de desarrollo. La disponibilidad de interfaces bien definidas, así como también las abundantes implementaciones de estos lenguajes de varios protocolos, le dan a los desarrolladores una ventaja cuando se construye en este modelo. Aquí el lenguaje desempeña un rol en el desarrollo funcional de cada servicio, manipulando objetos de datos, y la interacción con otros objetos que estén lógicamente encapsulados dentro del servicio.

### 3.1.11. Principios de la orientación a servicios<sup>27</sup>

Un problema con el que nos podemos encontrar a la hora de construir una aplicación SOA es si la aplicación construida realmente es una aplicación "SOA Compliant". Para comprobar si una aplicación lo es, la mejor forma de hacerlo es chequeando que la aplicación cumpla con los Principios de la Orientación a Servicios.

No existe una definición estándar de cuáles son los Principios de la Orientación a Servicios, por lo tanto, lo único que se puede proporcionar es un conjunto de Principios que estén muy asociados con la Orientación a Servicios. Estos Principios según Thomas Erl<sup>28</sup> son:

- *“Los Servicios deben ser reutilizables”*: Todo servicio debe ser diseñado y construido pensando en su reutilización dentro de la misma aplicación, dentro del dominio de aplicaciones de la empresa o incluso dentro del dominio público para su uso masivo.
- *“Los Servicios deben proporcionar un contrato formal”*: Todo servicio desarrollado, debe proporcionar un contrato en el cual figuren: el nombre del servicio, su forma de acceso, las funcionalidades que ofrece, los datos de entrada de cada una de las funcionalidades y los datos de salida. De esta manera, todo consumidor del servicio, accederá a este mediante el contrato, logrando así la independencia entre el consumidor

---

<sup>27</sup> <http://arquitecturaorientadaaservicios.blogspot.com/2006/06/principios-de-la-orientacin-servicios.html>

<sup>28</sup> Autor de uno de los libros más vendidos sobre SOA y editor de la serie "Prentice Hall Service-Oriented Computing Series " <http://www.thomaserl.com/>

y la implementación del propio servicio. En el caso de los Servicios Web, esto se logrará mediante la definición de interfaces con WSDL.

- *“Los Servicios deben tener bajo acoplamiento”*: Es decir, que los servicios tienen que ser independientes los unos de los otros. Para lograr ese bajo acoplamiento, lo que se hará es que cada vez que se vaya a ejecutar un servicio, se accederá a él a través del contrato, logrando así la independencia entre el servicio que se va a ejecutar y el que lo llama. Si conseguimos este bajo acoplamiento, entonces los servicios podrán ser totalmente reutilizables.
- *“Los Servicios deben permitir la composición”*: Todo servicio debe ser construido de tal manera que pueda ser utilizado para construir servicios genéricos de más alto nivel, el cual estará compuesto de servicios de más bajo nivel. En el caso de los Servicios Web, esto se logrará mediante el uso de los protocolos para orquestación (WS-BPEL) y coreografía (WS-CDL).
- *“Los Servicios deben de ser autónomos”* Todo Servicio debe tener su propio entorno de ejecución. De esta manera el servicio es totalmente independiente y nos podemos asegurar que así podrá ser reutilizable desde el punto de vista de la plataforma de ejecución.
- *“Los Servicios no deben tener estado”*: Un servicio no debe guardar ningún tipo de información. Esto es así porque una aplicación está formada por un conjunto de servicios, lo que implica que si un servicio almacena algún tipo de información, se pueden producir problemas de inconsistencia de datos. La solución, es que un servicio sólo contenga

lógica, y que toda información esté almacenada en algún sistema de información sea del tipo que sea.

- *“Los Servicios deben poder ser descubiertos”*: Todo servicio debe poder ser descubierto de alguna forma para que pueda ser utilizado, consiguiendo así evitar la creación accidental de servicios que proporcionen las mismas funcionalidades. En el caso de los Servicios Web, el descubrimiento se logrará publicando los interfaces de los servicios en registros UDDI.

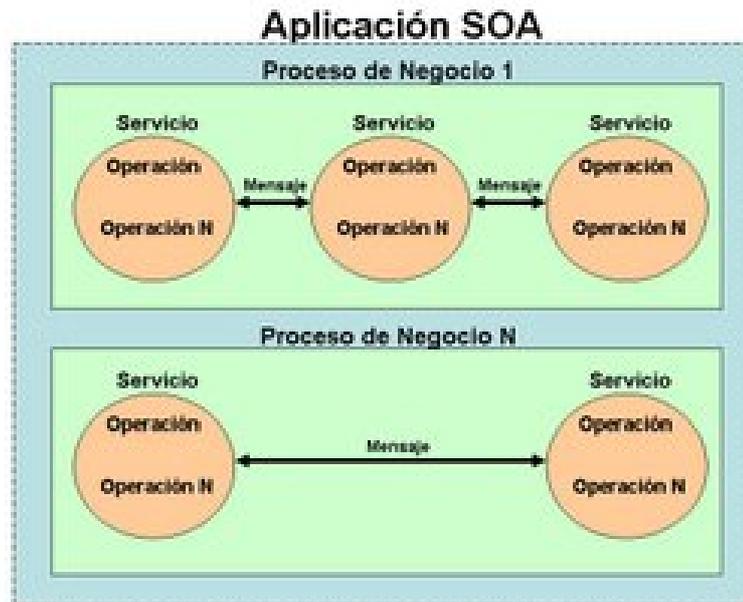
Cuando se desarrollan aplicaciones SOA es muy útil y necesario tener en cuenta siempre estos principios, ya que nos van a dar las pautas necesarias para tomar ciertas decisiones de diseño complejas. Como se habrá podido observar, una característica muy importante de los Principios de la Orientación a Servicios, es que todos ellos se inter-relacionan

### 3.1.12. Elementos esenciales de SOA<sup>29</sup>

Una aplicación SOA estará formada por un conjunto de procesos de negocio. A su vez esos procesos de negocio estarán compuestos por aquellos servicios que proporcionan las operaciones que se necesitan ejecutar para que el proceso de negocio llegue a buen término. Por último para ejecutar esas operaciones es necesario el envío de los datos necesarios mediante los correspondientes mensajes.

---

<sup>29</sup> <http://tecnoblog.entel.es/?p=23>



*Figura 20. Ejemplo de aplicación SOA*

Entre los elementos de SOA tenemos:

### **Proceso de negocio**

Son un conjunto de operaciones ejecutadas en una determinada secuencia, intercambiando mensajes entre ellas (Ver Figura 20) con el objetivo de realizar una determinada tarea. También se define como proceso de negocio a un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. Cada proceso de negocio tiene sus entradas, funciones y salidas<sup>30</sup>. Las entradas son prerequisites que deben tenerse antes de que una función pueda ser aplicada. Cuando una función es aplicada a las entradas de un método, tendremos ciertas salidas resultantes.

<sup>30</sup> [tecnoblog.entel.es/?p=23](http://tecnoblog.entel.es/?p=23)

Hay dos tipos de principales de procesos de negocio:

- **Procesos Centrales:** estos procesos dan el valor al cliente, son la parte principal del negocio
- **Procesos de Soporte:** dan soporte a los procesos centrales. Por ejemplo un proceso contable, un proceso de logística, etc.

### **Servicio**

Es un contenedor de lógica, función sin estado, auto-contenida, que acepta una(s) llamada(s) y devuelve una(s) respuesta(s) mediante una interfaz bien definida. Los servicios pueden también ejecutar unidades discretas de trabajo como serían editar y procesar una transacción. Los servicios no dependen del estado de otras funciones o procesos. La tecnología concreta utilizada para prestar el servicio no es parte de esta definición.

### **Orquestación**

Secuenciar los servicios y proveer la lógica adicional para procesar datos. No incluye la presentación de los datos.

### **Proveedor**

Función que brinda un servicio en respuesta a una llamada o petición desde un consumidor.

### **Consumidor**

Función que consume el resultado del servicio provisto por un proveedor; podrán ser tanto aplicaciones WEB, CRMs, ERPs, procesos Batch que se ejecutan de manera nocturna, etc.

## Repositorio de servicios

Un repositorio de servicios será algún componente de la arquitectura SOA que permitirá tanto a los Consumidores como a otros servicios, descubrir que servicios existen, cual es su interfaz y donde se encuentran físicamente. Los objetivos de este componente serán:

- Crear un nivel de indirección para localizar a los servicios
- Servir como repositorio de información de los servicios existentes, contratos, calidad de los mismos, etc.

## Bus de servicios (ESB)

Es el concepto que se refiere a la infraestructura de transporte de mensajes entre el motor de procesos y los servicios de los que dispone la empresa. Este componente de la arquitectura se utiliza sobre entornos heterogéneos de tecnologías de información. Tal como se ilustra en la Figura 21

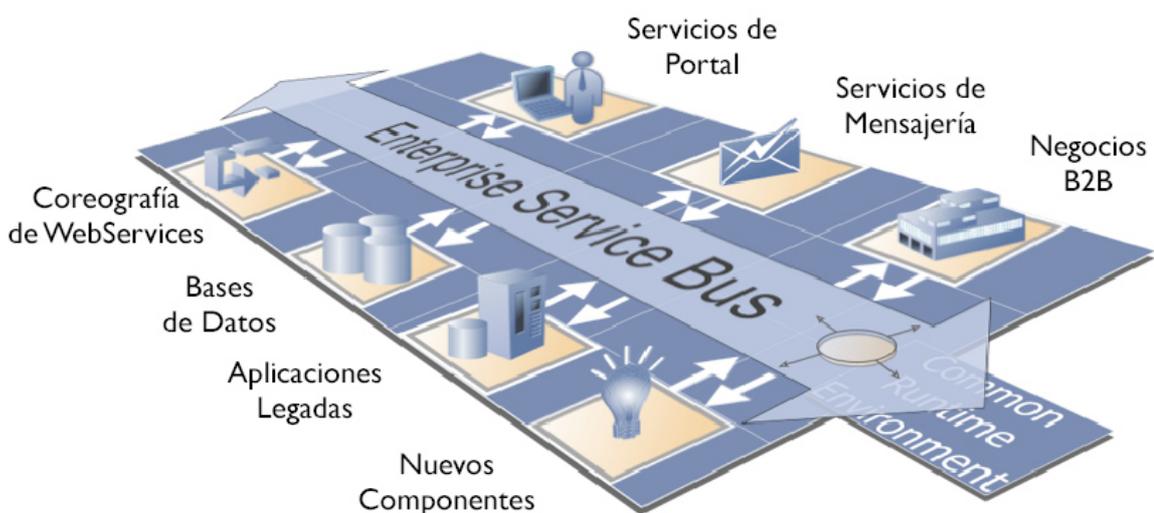


Figura 21. Elementos que interactúan con el Enterprise Service Bus

Este componente fundamental en la arquitectura SOA debe ofrecer:

- Conectividad entre frontales de aplicación y los servicios.
- Debe ser agnóstico de lenguajes de programación y tecnologías. Es decir debe ofrecer una forma de comunicación universal, para que todos puedan entenderse (por ejemplo, puede usar XML como formato de comunicación de los mensajes)
- Debe ser capaz de ofrecer diferentes paradigmas de comunicación (sincronismo y asincronismo).
- Debería ser capaz de ofrecer otra serie de funcionalidades transversales como:
  - Trazabilidad de las operaciones (capacidad de “logging”)
  - Mecanismos de seguridad (autenticación, autorización, encriptación, no repudio)
  - Mecanismos de transaccionalidad: protocolo de commit en dos fases (2PC) o transacciones en cadena y mecanismos de compensación, etc.
  - Enriquecimiento de mensajes, adaptación etc.
  - Control centralizado, mecanismos de monitorización.

- Que incluyese un procesador de BPM, que permitiese construir servicios de mayor valor añadido en base a servicios básicos, simplemente definiendo la lógica en algún lenguaje (ej. BPEL)

## **Mensaje**

Para poder ejecutar una determinada operación, es necesario un conjunto de datos de entrada. A su vez, una vez ejecutada la operación, esta devolverá un resultado. Los mensajes son los encargados de encapsular esos datos de entrada y de salida.

## **Operación**

Es la unidad de trabajo o procesamiento en una arquitectura SOA.

### 3.1.13. Tipos de arquitectura SOA

SOA como estrategia en las empresas permite construir diferentes tipos de soluciones tecnológicas, estas se aplican dependiendo del escenario en el que se ubique la organización, y considerando que no son rígidas, es decir pueden combinarse para dar origen a una arquitectura customizada. Estas son:<sup>31 32</sup>

#### **Nivel 1: Escenario de Creación de Servicios (*Service Creation Scenario*)**

Este tipo de escenario plantea la estructura básica de una arquitectura SOA (un proveedor de servicios, un consumidor de servicios y un directorio de

---

<sup>31</sup> <http://www.ibm.com/developerworks/library/ar-arprac2/>

<sup>32</sup> <http://www.ibm.com/developerworks/webservices/newto/>

publicación de los mismos), bajo este esquema se busca exponer funcionalidad de los sistemas existentes o un nuevo sistema de servicios. Los servicios pueden ser consumidos por otros servicios u otras aplicaciones cliente dentro de la empresa y con otras empresas.

Este nivel se basa en la identificación, construcción e publicación de servicios web con un directorio de descubrimiento de los servicios es decir UDDI. Para esto la plataforma de los aplicativos comprometidos debe ser homogénea.

## **Nivel 2: Escenario de Conectividad de Servicios (*Service Connectivity Scenario*)**

Este escenario plantea una arquitectura donde se tiene que integrar los servicios de los consumidores y de los proveedores, permitiendo el rehúso de funcionalidad existente o nueva a través de múltiples canales de comunicación.

Esta arquitectura es apropiada para las empresas que tienen un conjunto de servicios core (centrales) o sistemas que se desea estén disponibles como servicios para una gran variedad de clientes internos y externos. Plantea la posibilidad de interconectar procesos que interaccionan tanto hacia afuera como hacia dentro de la organización. Normalmente se asocia al concepto de ESB (*Enterprise Service Bus* o Bus de Servicio Empresarial) sobre todo si el número de aplicativos y interconexiones se hace poco manejable y la plataformas tecnológicas de cada sistema son heterogéneas.

### **Nivel 3: Escenario de Interacción y Colaboración (*Interaction and Collaboration Scenario*)**

La autenticación (single sign-on) y las funcionalidades basadas en roles (portales) son clave en este escenario para consolidar el acceso a la información y aplicaciones desde dentro de la empresa y entre empresas.

El uso de portales, documentos digitales, workflows y contenidos son los que priman en soluciones de este tipo. Ejemplos de esto son el uso de herramientas colaborativas que se enfocan en la *productividad* del personal y que pueden ser soluciones basadas en: Sharepoint de Microsoft, Oracle Portal o Websphere Portal

### **Nivel 4: Escenario de Administración de Procesos de Negocio (*Business Process Management Scenario BPM*)**

La innovación y optimización a través de la implementación de estrategias de negocio vía el modelado, ensamble, despliegue y monitoreo de procesos a lo largo de todo el ciclo de vida es el que prima en este escenario.

BPM actúa como un habilitador para las empresas definiendo e implementando estrategias de negocio que se alinean a sus políticas, información, personal y tecnología, todo esto en tiempo real. Se enfoca en la mejora continua de procesos. Este tipo de arquitecturas es generalmente utilizado por grandes corporaciones y que bajo cuya dirección se encuentran empresas de distinto tipo de actividad pero que pertenecen a un mismo grupo empresarial y que comparten recursos, logística y procesos.

## Nivel 5: Escenario de Información como Servicio (*Information as a Service Scenario*)

Este escenario plantea el uso de información resumida y fiable como servicios desde múltiples Fuentes de datos. Incluye aplicaciones como: data mining, BI. Se concentra en el conocer las necesidades y tendencias de mercado, es decir en el análisis de la información.

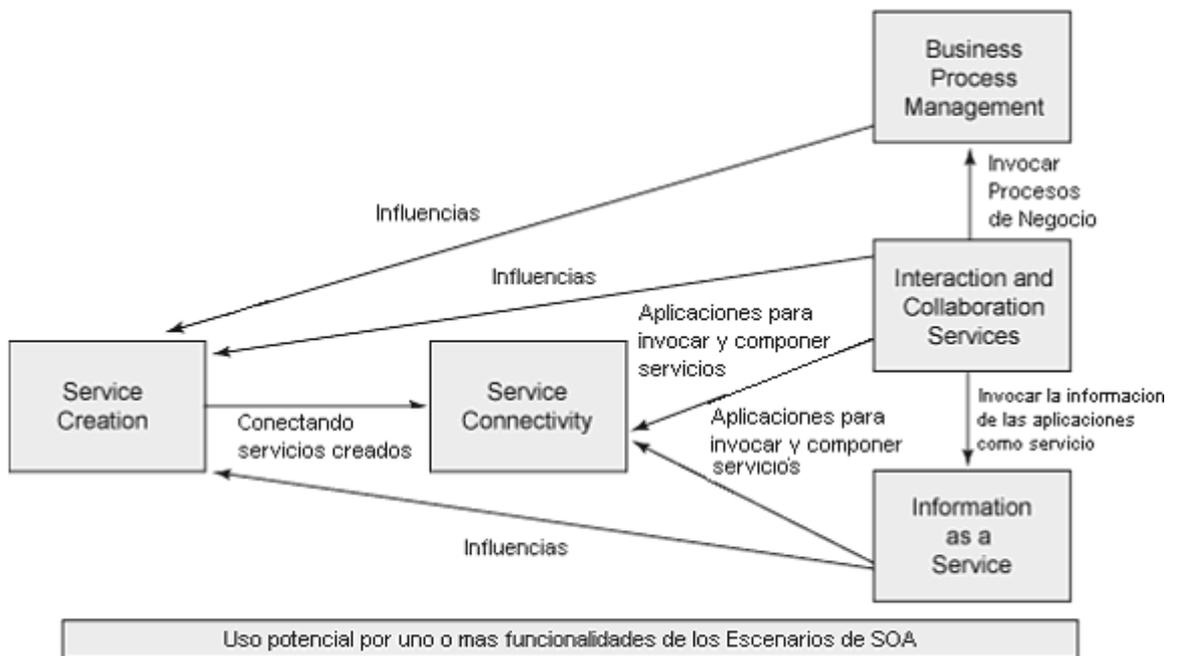


Figura 22. Escenarios de SOA y sus relaciones

## Interrelacionando los escenarios SOA

Pero estos cinco escenarios no tienen porque ser excluyentes entre si, de hecho una solución SOA en algún momento y dependiendo de cuanto crezca en complejidad puede ir incorporando comportamientos propios de otro escenario o ser un escenario hibrido donde se puedan relacionar los comportamientos de cada uno, así en la figura 22 se muestra la potencialidad de la integración de funcionalidades de los escenarios SOA

El escenario de “*Service Creation*” (Nivel 1) esencialmente crea servicios. Los servicios se pueden identificar del modelo del proceso del negocio en el escenario de “*Business Process Management*” (Nivel 4). Los servicios identificados en el escenario de “*Information as a Service*” (Nivel 5) y el escenario de “*Interaction and Collaboration Service*” (Nivel 3) están siendo usados en el escenario de creación de servicios.

Los escenarios de “*Interaction and Collaboration*” (Nivel 3) mantienen el panorama de acceso que se basa en el acceso a los servicios de la empresa a través de su portal para invocar procesos de negocio del escenario de “*Business Process Management*” (Nivel 4); tambien se pueden invocar servicios definidos en el escenario de “*Information as a Service*” (Nivel 5).

Los servicios definidos por el escenario de “*Interaction and Collaboration*” (Nivel 3) y el escenario de “*Information as a Service*” (Nivel 5), usan el escenario de “*Service Connectivity*” (Nivel 2) para comunicarse entre ellos.

Los servicios creados en el escenario de “*Service Creation*” (Nivel 1) tambien usan el escenario de “*Service Connectivity*” (Nivel 2) para proveer conexiones

entre los servicios creados de modo que juntos puedan formar servicios compuestos más complejos

#### 3.1.14. Capas de la Arquitectura SOA

En una arquitectura orientada a servicios la determinación de las capas que la componen es en general diversa y no existe un único modelo a seguir ya que cada solución se orienta hacia la solucionar a problemática de cada organización, pero si se indica que por lo menos debe contener:

##### **Capa de Presentación**

Está formado por:<sup>33</sup>

- Componentes de interfaz de usuario.
- Componentes de proceso de usuario.
- Independiente de implementación del resto de aplicación.
- Envía y recibe componente de entidades de negocio.
- Componentes de presentación:
  - No inician, participan, ni votan en transacciones
  - Obtiene una referencia al proceso actual del usuario si necesitan desplegar su data o actuar en su estado.

---

<sup>33</sup> <http://www.ibm.com/developerworks/library/ar-archtemp/>

- Pueden encapsular tanto la funcionalidades de visualización como de controlador

## **Capa de Negocio**

Está formado por:

- Interfaces de Servicio
  - Punto de entrada para abstraer implementaciones internas.
  - No debería cambiar cuando la implementación del componente de negocio cambia
  - Pueden existir distintas interfaces para la misma funcionalidad de negocio
  - Necesidad de interoperabilidad y rendimiento influye en el diseño de la interfaz de negocio
- Componentes de Negocio
  - Son invocados por la capa de presentación, interfaz de servicio, u otro proceso de negocio, usualmente en conjunto con alguna data de negocio para operar.
  - Proveen lógica de negocio o encapsulan otras lógicas de negocio
  - Son la raíz de las transacciones y votan en las transacciones donde participan
  - Validan operaciones de entrada y salida

- Exponen operaciones de compensación
- Pueden llamar a servicios externos a través de agentes de servicios, otros componentes de negocio e iniciar flujos propios de negocio
- Pueden levantar excepciones hacia el llamador del servicio si existe un problema con las transacciones involucradas.
- Componentes de Proceso de Negocio
  - Maneja procesos que involucran múltiples pasos y transacciones largas
  - Expone una interfaz que implementa un proceso que permite a la aplicación conversar con un servicio
  - Utiliza solo componentes
  - No necesita mantener estado conversacional mas allá de la actual actividad de negocio y la funcionalidad puede ser implementada como una transacción atómica única.
  - Necesita encapsular funcionalidad y lógica reutilizable por muchos procesos.
  - Lógica es intensiva o necesita a APIs y estructuras.
  - Necesita control fino sobre flujo de datos y lógica.

## **Capa de Servicio**

Está formado por:

- Entidades de Negocio
  - Proveen acceso programático con estados a los datos del sistema.
  - Son usados como argumentos de los procesos de negocio
  - Son serializables
  - No acceden directamente a la base de datos
  - No manejan transacciones, eso lo hacen los procesos de negocio.
  
- Componentes de acceso a datos
  - Proveen los métodos de CREATE, READ, UPDATE y DELETE
  - Proveen método específicos para el motor de datos.
  - Encapsulan la complejidad del modelado de datos.
  - Estas son clases sin estado.
  - Típicamente invocan procedimientos almacenados.
  - Para generalizar funciones comunes usa herencia.
  - Para soportar diversos procesos de negocio usa como argumentos los métodos de las entidades de negocio.

3.1.15. Quien define las pautas de SOA

Mucha gente se pregunta qué organismo es el encargado de estandarizar o por lo menos gestionar SOA. La respuesta es muy sencilla puesto que no hay ningún organismo que pueda hacerlo, ya que SOA es un concepto abstracto, el cual se rige única y exclusivamente por los principios de la Orientación a Servicios.<sup>34</sup>

Pero entonces, ¿No hay manera de controlar la evolución de SOA?, ¿Cada fabricante podrá hacer lo que quiera? Estas preguntas tienen una respuesta sencilla. Para comenzar, es necesario dejar un aspecto muy claro. Los Servicios Web, CORBA, MQSERIES, etc; son posibles tecnologías que se pueden utilizar a la hora de implementar una Arquitectura Orientada a Servicios y estas tecnologías sí que están estandarizadas y gestionadas por diversas organizaciones. Por lo tanto, las organizaciones que dirigen el rumbo de SOA, son aquellas que estandarizan las diferentes tecnologías utilizadas para implementar una Arquitectura Orientada a Servicios.

Por ejemplo, veamos el caso de los Servicios Web. Realmente los Servicios Web están formados por diferentes tecnologías, y por ello son varias las organizaciones que participan en su gestión. Concretamente son tres las organizaciones involucradas:

### **World Wide Web Consortium**

Organismo muy conocido porque es el encargado de la estandarización de HTML y XML (y tecnologías relacionadas). Referente a los Servicios Web, es el encargado de gestionar el protocolo de comunicación de los Servicios Web (SOAP), y el lenguaje de descripción de interfaces (WSDL). Más

---

<sup>34</sup> [arquitecturaorientadaaservicios.blogspot.com/2006/04/quin-define-las-pautas-de-soa.html](http://arquitecturaorientadaaservicios.blogspot.com/2006/04/quin-define-las-pautas-de-soa.html)

recientemente, W3C también se ha dedicado a estandarizar algunas de las extensiones WS-\* de los Servicios Web.. El W3C se caracteriza por ser muy formales y rigurosos a la hora de definir y gestionar tecnologías y protocolos, ofreciendo siempre las mejores garantías.

## **OASIS**

Anteriormente conocida como SGML Open, cambió su nombre para redirigir sus acciones de SGML hacia XML. Esta organización es bastante conocida por gestionar dos tecnologías muy conocidas. Es el encargado de desarrollar el estándar UDDI para el registro de Servicios Web, y también se encarga de gestionar la especificación ebXML es cual es un estándar para el intercambio de datos entre aplicaciones B2B. Actualmente también se encarga de desarrollar extensiones WS-\* para los Servicios Web. Estas extensiones son WS-BPEL creada para la orquestación de Servicios Web y WS-Security para todos los aspectos relacionados con la seguridad.

## **Web Services Interoperability**

Este organismo es reciente (apareció en 2002), y su principal objetivo es asegurar que se utilizan los estándares adecuados y no definirlos ni desarrollarlos. Por ello, este organismo ha definido un documento llamado perfil básico (Basic Profile), en el cual se indican cuales son los estándares que se deberían utilizar para diseñar arquitecturas inter-operables. Es decir, este documento es utilizado como mecanismo para generar arquitecturas SOA "compliant". Actualmente también se han preocupado por un aspecto tan importante como la seguridad, y han publicado un perfil de seguridad básico cuya finalidad es la misma que el anterior perfil, pero relacionado con la

seguridad. Además esta organización ha prometido seguir publicando perfiles para distintos aspectos relevantes de las arquitecturas.

### 3.1.16. Ventajas y Desventajas de SOA

#### **Ventajas<sup>35</sup>**

- Reducción de los periodos de desarrollo y los costes, los servicios SOA se reutilizan con facilidad y se ensamblan enseguida en nuevas aplicaciones compuestas.
- Costes de mantenimiento más bajos, los servicios reutilizables reducen la cifra y la complejidad interna de los servicios informáticos.
- Mayor calidad de servicios, la mayor reutilización de servicios devenga en una mayor calidad mediante múltiples ciclos de prueba por parte de distintos consumidores de servicios.
- Costes de integración más bajos, los servicios normalizados saben cómo interactuar entre sí, lo que facilita la rápida conexión de aplicaciones distintas.
- Menos riesgos, con menos servicios, pero reutilizables, se disfruta de mayor control sobre las políticas de responsabilidad informática y corporativa, así como disminuye el riesgo de incumplimiento global.

---

<sup>35</sup> [www.oracle.com/global/es/products/technologies/soa/index.html](http://www.oracle.com/global/es/products/technologies/soa/index.html)

- Gestionar y controlar fácilmente los datos y aplicaciones críticos de la organización.
- Invertir la mayor productividad y velocidad conseguidas en beneficio del negocio y de las Tecnologías de Información.
- Conservar los valiosos sistemas heredados de la empresa.
- Agilizar la entrega de los servicios.
- Consolidar la complejidad subyacente a las TI y simplificar la integración permanente.
- Como arquitectura, propone un cambio a la empresa como un todo.
- El bajo acoplamiento de los servicios genera la independencia del uso de uno u otro servicio.
- A nivel operativo, por ejemplo, rapidez en la implementación de nuevos procesos, economía en el mantenimiento, agilidad en su diseño, minimizando los costos.
- No propone un cambio radical, o una revolución, sino que permite utilizar módulos previos, o sistemas antiguos haciendo la migración menos violenta.
- Es a la medida del cliente.
- Es absolutamente modular, más flexible, pero también tenemos la ventaja de que nos facilita la extensión de la estructura que existe hoy.

## Desventajas<sup>36</sup>

- Requiere un cambio en las organizaciones, un alto esfuerzo.
- En la mayoría de las empresas, la proliferación de aplicaciones lleva al surgimiento espontáneo de una arquitectura corporativa “accidental”, esto es, no planeada.
- Incrementalmente se hace difícil y costoso el ser capaz de cumplir con los protocolos y hablar con un servicio a menos que se implemente un Bus de Servicio Empresarial.
- El conocimiento de los servicios es necesario para poder usar el servicio que proporciona un directorio de servicios. Dado que la Web es ilimitada por naturaleza, es imposible mantener todos los servicios web en un único directorio.
- Implica conocer los procesos del negocio, clasificarlos, extraer las funciones que son comunes a ellos, estandarizarlas y formar con ellas capas de servicios que serán requeridas por cualquier proceso de negocio.

Al 2008, SOA será predominante en las prácticas de Ingeniería de Software, terminando con 10 años de dominio de las arquitecturas monolíticas.

### 3.1.17. Elementos de SOA que son importantes para su éxito

Algunos de los puntos a tener en cuenta son los siguientes:<sup>37</sup>

---

<sup>36</sup> [www.oracle.com/global/es/products/technologies/soa/index.html](http://www.oracle.com/global/es/products/technologies/soa/index.html)

- Como primer punto se encuentra la flexibilidad. SOA es la primera arquitectura de Tecnologías de Información que asume lo que los negocios han sabido desde hace mucho tiempo. Se trata esencialmente de un conjunto de servicios sueltos, donde cada uno es relativamente económico para construirlo o reemplazarlo si es necesario. Al ser independientes, el poder unirlos permite a SOA adaptar cambios, cuestión imposible para arquitecturas tradicionales.
- En la Arquitectura Orientada a Servicios, se puede reemplazar un servicio sin tener que preocuparse por la tecnología fundamental; la interfaz es lo que importa, y está definida en un estándar universal en servicios Web y XML. Esto es flexibilidad a través de la interoperabilidad. También es la habilidad de asegurar los activos existentes, aplicaciones y bases de datos legales y hacerlos parte de las soluciones empresariales extendiéndolos al SOA en vez de reemplazarlos. El resultado en la red es la habilidad de evolucionar rápida y eficientemente, en otras palabras, adaptarse "orgánicamente" de acuerdo a la demanda del negocio. Esto es realmente nuevo.
- En segundo lugar está la relevancia para el negocio. SOA es tecnología expresada a un nivel que tiene un significado importante para la colaboración del negocio y profesionales del área. Sus servicios actuales pueden coordinar unidades de trabajo muy cercanas a las actividades del

---

<sup>37</sup> <http://www.mastermagazine.info/articulo/3391.php>

negocio; piense, por ejemplo, en un servicio llamado "Actualización de órdenes de trabajo". Éstos son inmediatamente relevantes para los analistas de la empresa que participan en la creación y definición de nuevos procesos permitiendo el "Servicio Dirigido Empresarial".

- Los negocios y las TI se enfocan en la lógica del negocio y la comunicación; finalmente comparten el lenguaje de servicios. Esto también es relativamente nuevo y tendrá implicaciones en la entrega de servicios

### 3.1.18. Casos de Negocio para SOA

Los casos de negocio que son fuentes comunes de proyectos en la industria, para utilizar una arquitectura orientada a servicios son los siguientes:

38

- Compra, adquisición o fusión de empresas en la industria, en las cuales es requerido integrar uno o varios de los sistemas empresariales.
- Automatización de procesos de negocio en las empresas. Los procesos de negocio pueden abarcan todo el ecosistema de empresas participantes en un proceso, tales como: clientes, empresas proveedoras, aliados de negocios, proveedores de servicios de públicos, entidades reguladoras, bancos, centrales de riesgo entre otros.
- Creación de una nueva línea o unidad de negocios en una empresa. Este caso normalmente requiere que la unidad de negocios soporte parte de sus operaciones con sistemas de información, sin embargo,

---

<sup>38</sup> [lucasian.com/soa/2007/07/27/casos-de-negocio-para-soa-en-las-empresas/](http://lucasian.com/soa/2007/07/27/casos-de-negocio-para-soa-en-las-empresas/)

estos sistemas comúnmente requieren integrarse con otros sistemas para intercambio de información y servicios, para lo cual se puede utilizar una arquitectura SOA.

- Modernización de Mainframes o Sistemas con tecnologías legadas. Este caso típico requiere que las empresas incrementalmente modernizar los sistemas legados. Para este caso, se utilizan normalmente integradores que permiten exponer servicios de los Mainframes o sistemas legados, de tal forma, que cuando las aplicaciones nuevas de las empresas requieran datos o transacciones de los sistemas legados, lo realicen a través de las tecnologías estándares propuestas por SOA, evitando el acoplamiento de los sistemas modernos con los sistemas antiguos. A futuro, se puede establecer un plan incremental que permita reemplazar módulos funcionales del Mainframe, cambiando la implementación tecnológica, sin cambiar la interfaz de servicios.
- Inteligencia de negocios. Un caso muy común en las empresas es consolidar la información de diferentes sistemas y crear un repositorio sobre el cual se puedan aplicar herramientas de inteligencia de negocios para mejorar los procesos de toma de decisiones. La extracción, enriquecimiento y homologación de información de los diferentes sistemas empresariales, se puede realizar utilizando una arquitectura SOA, en la cual normalmente se utilizan: Buses Empresariales de Servicios (ESB) y adaptadores de integración.
- Consolidación de Información de Clientes.
- Creación de una nueva línea o unidad de negocios en una empresa.

### 3.1.19. Barreras a vencer para obtener el éxito de SOA

SOA es un nuevo horizonte para las TI. Como cualquier gran cambio, las principales barreras son organizacionales, no técnicas. A continuación ejemplificaremos algunas:<sup>39</sup>

- **Administración:** Servicios compartidos es lo principal para utilizar SOA. La habilidad para ensamblar rápidamente aplicaciones o procesos está basada en la disponibilidad de algunos servicios que pueden ser compartidos. Hacer esto, por definición, requiere administración.
- **Desarrollo Cultural:** Al utilizar SOA se requiere un cambio significativo en el estilo de programar. Muchos desarrolladores utilizan equipos diferentes para resolver problemas de manera independiente para cada aplicación. En SOA necesitarán escribir aplicaciones para ser re-utilizadas en mente, usando códigos existentes, a los cuales se podrá tener acceso constantemente.

### 3.1.20. Lo que puede lograrse con una Arquitectura SOA

SOA surge de la necesidad de hacer que los sistemas de negocios de IT sean más ágiles con respecto a los cambios en la empresa. Al permitir relaciones fuertemente definidas, si bien implementaciones específicas flexibles, los sistemas pueden obtener las ventajas de los otros sistemas

---

<sup>39</sup> [www.mastermagazine.info/articulo/3391.php](http://www.mastermagazine.info/articulo/3391.php)

existentes y, no obstante, estar lista para cambios futuros en sus interacciones.<sup>40</sup>

Para dar un ejemplo específico, una organización minorista de indumentaria que posee una cadena internacional de 500 tiendas necesita cambiar frecuentemente sus diseños para mantenerlos a la moda. Esto podría significar no sólo cambios en estilos y colores, sino también en materiales, fabricación y entrega. Cambiar de un proveedor a otro puede ser un proceso de software complicado si los sistemas que hay entre el minorista y el fabricante son incompatibles. La flexibilidad de una interfaz de WSDL para las operaciones puede permitir que cada compañía mantenga sus sistemas existentes tal como son. En vez de ello, pueden sólo hacer coincidir las interfaces de WSDL y establecer nuevos convenios de nivel de servicio en vez de reconstruir totalmente sus aplicaciones de software. Este es un cambio horizontal para la empresa, es decir, están cambiando asociados al tiempo que esencialmente todas las operaciones de negocios permanecen siendo las mismas en su mayoría. Aquí las interfaces de negocios pueden cambiar de una manera mínima y las operaciones internas pueden permanecer sin cambios, al tiempo que las interfaces de negocios todavía pueden funcionar juntas externamente.

Otra forma es el cambio interno, en la que la organización minorista ahora decide que también alquilará espacio dentro de la cadena de tiendas minoristas a vendedores de boutiques, tal como el modelo de negocios de tienda a tienda. Aquí la mayoría de las operaciones de negocios de la

---

<sup>40</sup> [www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

compañía permanecen sin cambios, pero ahora se necesita un nuevo software interno para manejar ese convenio de alquiler. Internamente, es posible que el sistema de software sea objeto de una revisión, pero necesita hacer esto sin afectar seriamente las interacciones con los sistemas existentes de sus proveedores. El modelo SOA en este caso permanece intacto, al tiempo que la implementación interna cambia. Se pueden agregar nuevos aspectos al modelo SOA para agregar las nuevas responsabilidades de los acuerdos de alquiler, mientras que el sistema regular de administración minorista continúa siendo el usual.<sup>41</sup>

Para continuar aún más con la idea del cambio interno, puede que el gerente de tecnología, encuentre que se puede usar la nueva configuración de otras maneras, tal como rentar también espacio de “poster” para publicidad. Aquí surge una nueva propuesta de negocios del modelo SOA flexible re aplicado en un nuevo diseño. Este es un nuevo resultado de un modelo SOA y una nueva oportunidad que podría no ser posible anteriormente.

También son posibles los cambios verticales, donde el minorista pasa totalmente de vender sus propias ropas a alquilar exclusivamente espacio a través del modelo tienda-en-tienda. Si se toma completamente desde una base cero, un cambio vertical comprendería una reestructuración significativa del modelo SOA, tal vez con nuevos sistemas, software, procesos y relaciones. La ventaja del modelo SOA en este caso es que funciona desde la perspectiva de las operaciones y de los procesos de negocios, en vez de aplicaciones programas, permitiéndole a la administración de los negocios identificar

---

<sup>41</sup> [www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

claramente que necesidades deben ser agregadas, modificadas o eliminadas sobre la base del funcionamiento de la empresa. Luego se pueden estructurar los sistemas de software para ajustar los procesos de negocios, en vez de ser lo opuesto, según se ve comúnmente en muchas plataformas de software.

Tal como se puede ver, el cambio y la capacidad del sistema SOA para adaptar a él son los elementos más importantes aquí. Para los desarrolladores, tales cambios pueden ocurrir dentro de su contexto de trabajo o fuera de él, según si hubiera cambios que son necesarios en cuanto a cómo las interfaces son definidas y cómo interactúan entre sí. En vez de ser el rol del desarrollador, es el del arquitecto engendrar la mayoría de los cambios en un modelo SOA. Esta división de trabajo, por la cual el desarrollador se concentra en crear unidades funcionales definidas como servicios, y el arquitecto y el modelador se concentran en la forma en que las unidades se ajustan entre sí, ha existido por más de una década, comúnmente representado en el *Universal Modeling Language* (UML) Lenguaje Universal de Modelado

### 3.1.21. Como puede beneficiar SOA a los negocios

El establecimiento de una arquitectura orientada a servicios puede ayudar a preparar tanto a las tecnologías de información, como a los procesos de negocios para un cambio rápido. Aún en las etapas tempranas de adopción de una SOA, su organización puede beneficiarse.<sup>42</sup>

---

<sup>42</sup> [www-306.ibm.com/software/ar/info/topic/openenvironment/soa/](http://www-306.ibm.com/software/ar/info/topic/openenvironment/soa/)

- Aumenta los ingresos, crea nuevos caminos al mercado; crea nuevos valores a partir de sistemas existentes.
- Provee un modelo de negocios flexible, busca el reaccionar más rápidamente a los cambios del mercado.
- Reduce a mediano plazo los costos, elimina los sistemas duplicados; construye una vez y potencia; mejore el tiempo de salida al mercado.
- Reduce los riesgos y la exposición, mejora la visibilidad en las operaciones de negocio.

El enfoque SOA puede colocar un puente entre lo que usted quiere que su negocio cumpla y las herramientas de infraestructura que tiene que tener para llegar allí.

- Disminuya los tiempos de los ciclos de desarrollo e implementación usando bloques de construcción de servicios reutilizables, prefabricados.
- Integre por toda la empresa, incluso los sistemas históricamente separados y facilite las fusiones y adquisiciones de empresas.
- Reduzca tiempos de ciclo y costos pasando de transacciones manuales a automáticas.
- Facilite la realización de negocios con los asociados de negocios aumentando su flexibilidad.

### 3.1.22. Cómo SOA puede afectar los resultados de los negocios

Dependiendo de las prioridades de su industria, SOA puede ayudar a reducir el tiempo y los costos de entrega de nuevos servicios<sup>43</sup>. Ayude a su empresa a responder más rápido a las exigencias del cliente. Proporcione a los clientes una experiencia de usuario unificada. Fortalezca los recursos de seguridad y reduzca los riesgos.

- Problema: La industria automotriz enfrenta problemas de calidad, los costos de garantías promedian US\$ 700 por vehículo en los Estados Unidos.
  - Solución: Trabajar hacia multi-proveedor en sistemas embarcados e integración de software con SOA.
- Problema: Las empresas de atención de salud deben abordar la subida de costos, tiempos de respuesta lentos y la calidad inconsistente de los registros de pacientes.
  - Solución: Usar SOA para integrar a los pagadores, proveedores y hospitales.
- Problema: La industria electrónica está cambiando de la fabricación tradicional a la configuración bajo pedido.
  - Solución: Construir SOA que facilite la producción en masa con personalización de último minuto.

---

<sup>43</sup> [www-306.ibm.com/software/ar/info/topic/openenvironment/soa/](http://www-306.ibm.com/software/ar/info/topic/openenvironment/soa/)

- Problema: La industria de bancos debe tratar con silos de información, redundancia y reutilización de datos, mientras está bajo una presión constante para crecer.
  - Solución: Responder con SOA para acelerar el desarrollo y la entrega de nuevos productos y servicios.
- Problema: Los minoristas se enfrentan al crecimiento exponencial de los datos (por ejemplo, RFID) que no son potenciados eficazmente.
  - Solución: Entregar información casi en tiempo real para optimizar la cadena de abastecimiento con SOA.
- Problema: Las empresas de telecomunicaciones se enfrentan con “islas” de infraestructuras, múltiples sistemas legados y entornos heterogéneos.
  - Solución: Diseñar SOA que ofrezca una única vista del cliente, desde la activación de cuenta al autoservicio, facturación y atención al cliente.

### 3.1.23. Modelado de Procesos de Negocio

El análisis de los modelos de procesos de negocios está enfocado principalmente a la completitud de éstos, es decir, qué elementos de la realidad a ser modelada pueden ser representados. El ámbito del Modelamiento considera aspectos inherentes a los procesos de negocios y aspectos de la relación proceso de negocio- sistemas informáticos.

Un proceso de negocio es un conjunto estructurado de actividades, diseñado para producir una salida determinada o lograr un objetivo. Los procesos describen cómo es realizado el trabajo en la empresa y se caracterizan por ser observables, medibles, mejorables y repetitivos<sup>44</sup>.

Estructuralmente, un proceso de negocio está constituido por un conjunto de actividades. Así, la actividad, como elemento básico, mediante relaciones o dependencias con otras actividades conforma la estructura de un proceso de negocio. Por otro lado, un sistema informático es una colección de componentes organizados para cumplir con funciones específicas, mediante la tecnología informática.

Un proceso de negocio es una colección de actividades estructurales relacionadas que producen un valor para la organización, sus inversores o sus clientes. Es, por ejemplo, el proceso a través del que una organización realiza sus servicios a sus clientes.

Un proceso de negocio puede ser parte de un proceso mayor que lo abarque o bien puede incluir otros procesos de negocio que deban ser incluidos en su función. En este contexto un proceso de negocio puede ser visto a varios niveles de granularidad. El enlace entre procesos de negocio y generación de valor lleva a algunos practicantes a ver los procesos de negocio como los flujos de trabajo que se efectúan las tareas de una organización.

---

<sup>44</sup> [www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf](http://www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf)

Los procesos de negocio pueden ser vistos como un recetario para hacer funcionar un negocio y alcanzar las metas definidas en la estrategia de negocio de la empresa.<sup>45</sup>

Los procesos de negocio consisten en *subprocesos*, *decisiones* y *actividades*.

Un *subproceso* es parte de un proceso de mayor nivel que tiene su propia meta, propietario, entradas y salidas.

Las *actividades* son partes de los procesos de negocio que no incluyen ninguna toma de decisión ni vale la pena descomponer (aunque ello sea posible). Por ejemplo, “Responder al teléfono”, “Hacer una factura”

Un sistema informático puede también ser visto, en su operación, como un conjunto de funcionalidades operativas. Este concepto se relaciona con el conjunto de funciones de información que están siendo requeridas por las actividades de un proceso de negocio. Es decir, los procesos de negocios requieren funciones de información, y los sistemas informáticos entregan funcionalidades operativas, que están apoyando a las actividades de los procesos. Se define funcionalidad operativa a una función de información que se encuentra activa o forma parte del apoyo que los sistemas informáticos entregan a los procesos.

Lo anterior, conforma la base conceptual, que permite identificar los elementos que necesitan ser representados en los modelos de procesos de negocios que consideren explícitamente su relación con sistemas informáticos.

---

<sup>45</sup> es.wikipedia.org/wiki/Proceso\_de\_negocio

## Elementos representables de los procesos de negocios- sistemas informáticos

Es esencial poder identificar a cada uno de los procesos de negocios que están siendo modelados<sup>46</sup>, de tal manera que son elementos básicos el nombre del proceso y su tipo. La figura 23 describe elementos básicos que permiten conocer y entender cuál es el rol de cada actividad dentro de un proceso de negocio

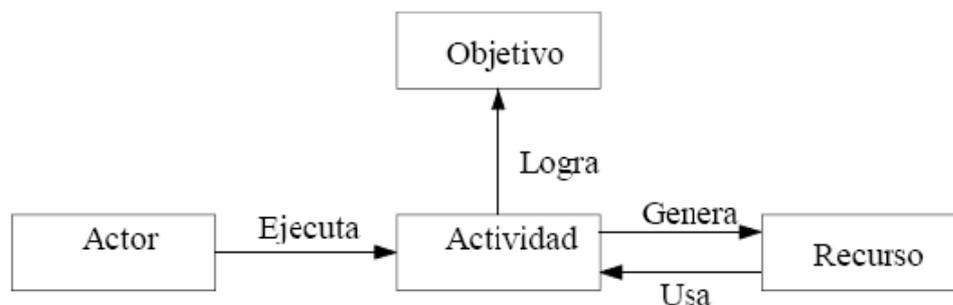


Figura 23. Elementos básicos relacionados con las actividades

El actor es el elemento encargado de realizar la actividad. Pueden ser individuos, grupos de personas o departamentos organizacionales. El objetivo de una actividad es una característica propia que indica el propósito de su existencia dentro del proceso de negocio al que pertenece. En general un recurso es todo aquello que es usado o afectado por las actividades. A menudo son considerados como entradas, salidas o resultados, o herramientas. La dependencia entre actividades y recursos, en un modelo simple de acciones:

<sup>46</sup> [www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf](http://www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf)

las actividades tienen condiciones previas (entradas a una actividad) y efectos (salidas de una actividad). Las condiciones previas o entradas son recursos requeridos, consumidos o utilizados por las actividades. Las herramientas o maquinarias necesarias para ejecutar una actividad, son recursos que no son consumidos pero si son usados para llevar a cabo la actividad. Los efectos son recursos creados por una actividad. Un tipo particular de recurso lo constituyen los sistemas informáticos. Estos se caracterizan por ser no-consumibles, reutilizables y compartibles.

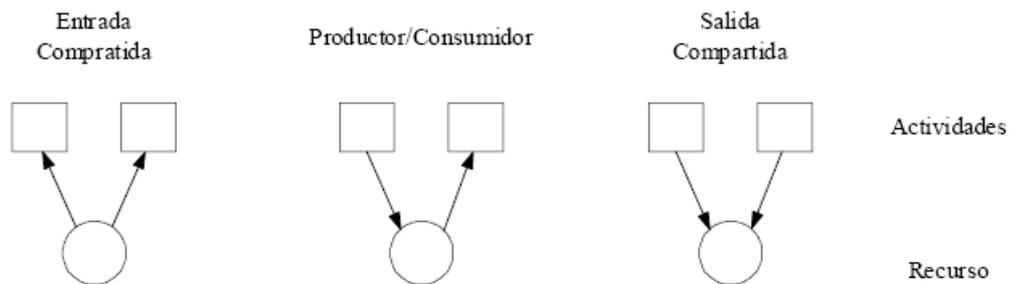
### **Relación entre actividades y recursos**

Es necesario conocer cómo se relacionan las actividades de los procesos de negocio con los sistemas informáticos, ya que de esta manera es posible identificar qué elementos son eventualmente representables mediante un modelo.

### **Relación entre múltiples actividades y recursos**

Para simplificar la relación entre varias actividades y recursos, se considera sólo dos actividades y un recurso común. Existen tres maneras en que dos actividades pueden relacionarse entre ellas, dependiendo de la existencia de un recurso en común: compartiendo el recurso como entrada; el recurso es la salida de una actividad y la entrada de otra; o el recurso es la salida de ambas actividades.

En la figura 24 se muestra las dependencias posibles entre dos actividades y un recurso que es común a ambas.



*Figura 24. Dependencia entre múltiples actividades y un recurso*

### **Entrada compartida**

Dos actividades son interdependientes si ambas necesitan como entrada un mismo recurso, que puede requerir de una actividad adicional para lograr ser compartido. Claramente la naturaleza del recurso determina la actividad adicional que es necesaria. Se consideran en particular dos dimensiones diferentes dentro de las entradas compartidas: Compatibilidad y Reusabilidad.<sup>47</sup>

La **Compatibilidad** describe cuántas actividades pueden usar un recurso simultáneamente. La mayoría de los recursos como materias primas o herramientas no son compartibles. La información es una excepción importante, ya que puede ser utilizada por múltiples actividades mientras no sea alterada por alguna de ellas. Hay que notar que un recurso que es compatible no es consumible.

La **Reusabilidad** describe la cantidad de actividades que pueden usar un recurso ya utilizado. Algunos recursos, tal como herramientas o información, se pueden usar y re-usar; en cambio otros, tal como materias primas, pueden

<sup>47</sup> [www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf](http://www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf)

usarse sólo una vez. Por supuesto que las herramientas eventualmente podrían gastarse y necesariamente ser reemplazadas, pero la distinción importante es identificar si otra actividad podrá utilizar el recurso si espera, como es el caso de las herramientas y las máquinas, o si el recurso simplemente ha sido consumido, como ocurre con las materias primas.

Si el recurso común es compartible, como es el caso de la información, entonces no hay conflicto entre dos actividades para usarlo al mismo tiempo. Por ejemplo, el área de ventas y producción pueden usar información simultáneamente sin ningún problema, aunque hay que considerar la ocurrencia de conflictos, es decir, si el sistema informático asociado tiene la propiedad de permitir el que dicha información sea compartida.

### **Productor / consumidor**

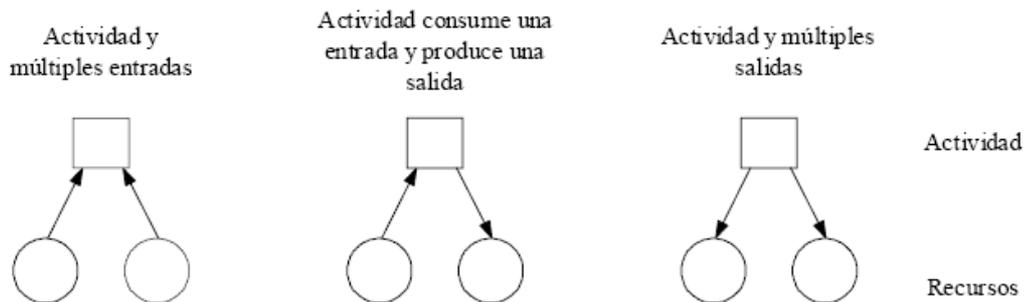
Si el resultado de una actividad es la entrada para otra, entonces hay una dependencia de precedencia entre las dos actividades, que requiere la ejecución de las actividades en el orden correcto. Esta relación frecuentemente ocurre entre las actividades de un proceso de negocio, y es también llamada relación de orden.

### **Salida compartida**

Esta situación ocurre cuando dos actividades generan el mismo recurso o salida. Dentro de este caso existe la posibilidad que el recurso resultante de las actividades sea el mismo o que las actividades produzcan colaborativamente una salida.

## Relación entre una actividad y múltiples recursos

La dependencia entre una actividad y varios recursos puede ocurrir cuando: una actividad consume múltiples recursos, cuando consume un recurso y produce otro, o cuando produce múltiples recursos. La figura 25 resume lo anterior



*Figura 25. Relaciones entre una actividad y múltiples recursos*

## Actividad y múltiples entradas

En este caso debe existir una sincronización en la disponibilidad de los múltiples recursos para que la actividad pueda realizarse. Cuando una actividad se relaciona con otras, a través de los recursos generados por estas últimas, debe esperar que dichas actividades terminen de generar los recursos. Esto se conoce como relación de convergencia, y se llama punto de convergencia al momento en que la actividad puede comenzar a utilizar los múltiples recursos.

Los siguientes dos tipos de relaciones (la actividad consume un recurso y produce otro, o cuando produce múltiples recursos) están orientados a la

naturaleza intrínseca de las actividades, es decir, la generación de uno o más recursos.

Existen otros elementos característicos de las actividades de un proceso de negocio, que son descritas a continuación.

### **Características estáticas de las actividades<sup>48</sup>**

**Nivel de Importancia:** No todas las actividades son igual de importantes. El nivel de importancia estará dado por los tipos de relaciones que tiene la actividad con otras, que componen un mismo proceso.

**Capacidad de Automatización:** Una actividad puede clasificarse en: No automatizada, Semi-automatizada y automatizada, dependiendo de su naturaleza. El término automatización está visto desde la perspectiva de sistemas informáticos, excluyendo otro tipo de tecnologías.

Una actividad es automatizable, si es repetitiva en el tiempo y estructurada. Al disminuir la estructuración, la actividad tiende a ser menos automatizable.

Existen otras características inherentes a las actividades: tal como la capacidad, la re-ejecución y la ubicación, que se consideran menos relevantes, en relación a los elementos informáticos participantes. La capacidad se define como la cantidad de producto que puede ser obtenido por una actividad durante un cierto periodo de tiempo. Cuando una actividad produce algo que no es lo adecuado, que no cumple los requisitos de calidad o que no está

---

<sup>48</sup> [www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf](http://www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf)

estipulado como una salida aceptable en la definición de la actividad, la actividad en algunos casos deberá ejecutarse nuevamente para producir la salida adecuada. Esta propiedad de la actividad para poder ejecutarse de nuevo, recibe el nombre de re-ejecución. La ubicación de una actividad está ligada a alguna locación geográfica, por ejemplo, algún tipo particular de oficina, o un lugar específico como la bodega o el sector B de la planta de producción, e indica dónde debe ser realizada dicha actividad.

Las actividades también poseen características obtenidas en su ejecución, que permiten planificar su ejecución y monitorearlas. Estas características que son medibles, varían su valor o su estado automáticamente mientras es efectuada la actividad.

### **Elementos representables considerados en el análisis de los modelos de procesos de negocios<sup>49</sup>**

Los elementos representables se han clasificado en básicos y avanzados. Los básicos son identificados en la mayoría de los modelos de procesos de negocio. Los elementos avanzados son aquellos que permiten representar la dimensión informática.

El cuadro 11, muestra un resumen de los elementos básicos inherentes a los procesos de negocios - sistemas informáticos.

---

<sup>49</sup> [www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf](http://www.inf.udec.cl/revista/ediciones/edicion9/cjimenez.pdf)

<b>Elementos Básicos</b>
(1) Identificación de elementos: Nombre proceso Nombre Sistema Informático Nombre de actividades
(2) Actividad
(3) Actores
(4) Recursos
(5) Decisiones asociadas a las actividades <sup>2</sup>
(6) Entradas y Salidas por actividad
(7) Flujos entre actividades

*Cuadro 11. Elementos representables básicos de un modelado de procesos de negocio*

El cuadro 12 muestra los elementos inherentes a los procesos de negocios - sistemas informáticos, a nivel avanzado.

<b>Elementos Avanzados</b>
(1) Identificación de elementos: Nombre de Módulos Nombre de Funcionalidad Operativa
(2) Objetivos de las actividades
(3) Capacidad de Automatización
(4) Nivel de Importancia
(5) Entrada Compartida
(6) Salida Compartida
(7) Productor/Consumidor
(8) Puntos de Convergencia de Actividades
(9) Tipo de Proceso de Negocio
(10) Funcionalidad Operativa
(11) Tipo de Funcionalidad Operativa
(12) Tipo de Sistema Informático

*Cuadro 12. Elementos representables avanzados de un modelado de procesos de negocio*

3.1.24. Metodología para elaboración de Arquitecturas de Sistemas Microsoft

La estrategia de arquitectura de Microsoft define, en consonancia con las conceptualizaciones más generalizadas, cuatro vistas, ocasionalmente llamadas también arquitecturas: Negocios, Aplicación, Información y Tecnología<sup>50</sup>. Cada arquitectura, a su vez, se articula en vistas también familiares que son: la Vista Conceptual, cercana a la semántica de negocios y a la percepción de los usuarios no técnicos; la Vista Lógica, que define los componentes funcionales y su relación en el interior de un sistema, en base a la cual los arquitectos construyen modelos de aplicación que representan la perspectiva lógica de la arquitectura de una aplicación; y la Vista Física, que es la menos abstracta y que ilustra los componentes específicos de una implementación y sus relaciones., mediante protocolos de comunicación y sistemas de hardware. A continuación se grafica lo expuesto en el cuadro 13.

	Negocio	Aplicación	Tecnología	Información
Conceptual	Modelo de negocio	Sistemas de servicios	SOA WS	KM

<sup>50</sup> [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/intro.msp](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp)

Lógica	B2B, B2C etc.	Diagramas de aplicación	Capas	Información y esquemas
Física	Business Txs	Código	Protocolos y HW	Datos

*Cuadro 13. Metodología Microsoft para el desarrollo de Arquitecturas de sistemas*

La arquitectura se puede ver desde distintos enfoques dentro de la organización, algunas de las perspectivas son:

**Negocio:** Describe el funcionamiento interno del negocio central de la organización.

**Aplicación:** Muestra las aplicaciones de la organización, su funcionalidad y relaciones.

**Información:** Describe la información que maneja la organización y cómo está ligada a los circuitos de trabajo.

**Tecnología:** Describe la estructura de hardware y software de base que da soporte informático a la organización.

### 3.2. Definición de Términos Básicos

#### 3.2.1. Servicio Web:

Es una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma,

pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet

### 3.2.2. UDDI:

Son las siglas del catálogo de negocios de Internet denominado Universal Description, Discovery and Integration; es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros

### 3.2.3. HTTP: (HyperText Transfer Protocol)

Es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con campos de texto

### 3.2.4. WSDL:

Son las siglas de Web Services Description Language; describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

#### 3.2.5. XML:

Sigla en inglés de eXtensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores

#### 3.2.6. SOAP:

Siglas en inglés de Simple Object Access Protocol; es un protocolo estándar creado por Microsoft, IBM y otros; define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

#### 3.2.7. Granularidad (Objetos de Grano Fino y Grueso):

La granularidad es indirectamente proporcional a la riqueza funcional de un servicio. Esto significa que entre más específica es la funcionalidad del servicio, mayor o más fina es su granularidad. Se dice que un servicio tiene granularidad gruesa, cuando tiene una mayor riqueza funcional y un servicio tiene una granularidad fina cuando hace una función específica.

Por ejemplo, un retiro es una función específica por lo que se le clasifica como una granularidad fina mientras que una función de compra de deuda tiene una granularidad gruesa o mayor porque no solo realiza un retiro o un cargo a una línea de crédito, sino también uno o varios depósitos e incluso operaciones de reversión en caso de falla.

El nivel de granularidad puede variar de acuerdo al contexto, en los siguientes puntos definiremos la taxonomía de servicios entre los que encontraremos clasificaciones de acuerdo a su nivel de granularidad.

#### 3.2.8. .NET:

Proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones. Basado en esta plataforma, Microsoft intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el Sistema Operativo hasta las herramientas de mercado.

#### 3.2.9. DCOM:

Siglas en inglés de Distributed Component Object Model ( Modelo de Objetos de Componentes Distribuidos ), es una tecnología propietaria de Microsoft para desarrollar componentes software distribuidos sobre varios ordenadores y que se comunican entre sí. Extiende el modelo COM de Microsoft y proporciona el sustrato de comunicación entre la infraestructura del servidor de aplicaciones COM+ de Microsoft.

### 3.2.10. Middleware:

Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

### 3.2.11. CORBA:

Siglas en inglés de Common Object Request Broker Architecture (arquitectura común de intermediarios en peticiones a objetos), es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

## **4. METODOLOGIA DE LA INVESTIGACION**

### **4.1. Estado del Arte: Análisis de Arquitecturas y Soluciones Existentes**

#### 4.1.1. Comparativa entre arquitecturas (BenchMarking)

Desde los inicios de los sistemas de información hace ya más de 3 décadas han existido diversas evoluciones de las arquitecturas de software, cada una proveía las herramientas y la base para desarrollar los sistemas de información que cumplían los requerimientos y exigencias que se iban presentando, desde las arquitecturas monolíticas, estructuradas pasando por la más conocida que es la cliente / servidor y que tuvo un gran auge en su momento, luego la gran explosión de las tecnologías de información y el aumento vertiginoso de consumidores de estas tecnologías en las organizaciones obligaron a cambiar el paradigma y llegó el modelo de n capas el cual fue el soporte para las subsiguientes arquitecturas distribuidas ya que con la evolución de las tecnologías de información, se hizo necesario integrar la gran variedad de aplicaciones existentes, en la figura 26 se puede observar la evolución en el tiempo de las mismas.

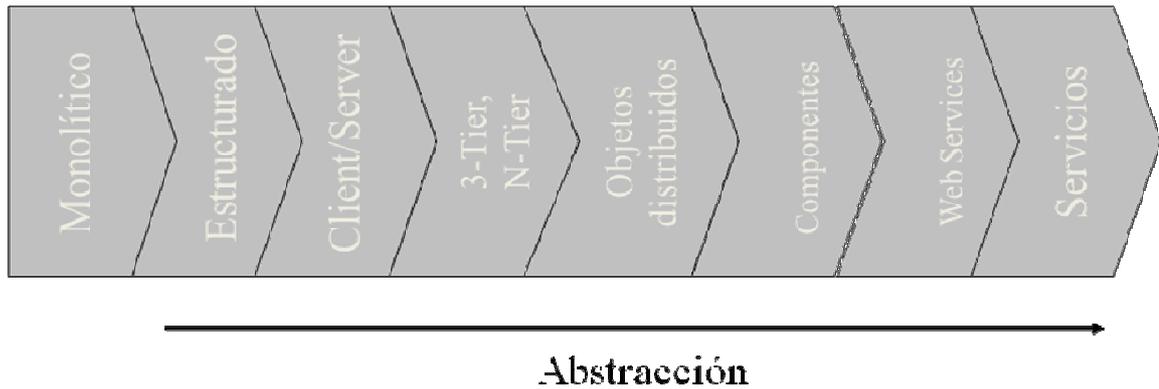


Figura 26. Evolución de las abstracciones arquitectónicas

La arquitectura orientada a servicios como ya sabemos es solo una de las diversas variantes que existen actualmente al afrontar el tema de encontrar una arquitectura de software que permita la interoperabilidad del software en las organizaciones de hoy en día. Dada la problemática de nuestro trabajo de investigación nos enfrentamos a la decisión de sopesar los pro y los contras de cada una de las posibilidades que se ofrecen tanto en el mercado nacional como con las experiencias de otras empresas que pasan por el mismo tipo de problema, a continuación en el cuadro 14 describiremos los criterios que llevaron a la elección de la arquitectura de software propuesta

Arquitectura	Cliente / Servidor 2 capas	N-capas basados en componentes	N-capas orientada a Servicios
Característica			
Perspectiva	Elementos <b>internos</b> de un	Elementos <b>internos</b> de un	Elementos que se ven <b>desde fuera</b>

<b>arquitectónica</b>	componente de software	sistema	del sistema
<b>Interfaz</b>	Aplicaciones para ejecutarse desde formularios	Aplicaciones para ejecutarse desde formularios y/o en la web	Aplicaciones para ejecutarse desde formularios y/o en la web
<b>Modelo de despliegue</b>	Junto con la aplicación o el componente	Despliegue “físico” <b>(instalar-y-usar)</b>	El software “existe” en algún lado <b>(conectar-y-usar)</b>
<b>Ámbito de intercambio de información</b>	<b>Dentro</b> de la empresa	Mayoritariamente <b>dentro</b> de la empresa	Mayoritariamente <b>entre</b> empresas
<b>Niveles de acoplamiento</b>	Fuerte	Intermedia	Débil
<b>Comunicación</b>	Interfaces	Middleware	Basada en la web (SOAP/XML sobre HTTP)
<b>Granularidad</b>	Fina	Fina	Gruesa
<b>Escalabilidad</b>	Muy pobre	Buena	Buena

<b>Reusabilidad</b>	Baja	Intermedia	Muy Alta
<b>Integración , InterOperabilidad</b>	Limitada	Dentro de la organización	Dentro y fuera de la organización
<b>Adecuación a los requerimientos del negocio</b>	Muy Lenta	Lenta	Rápida
<b>Nivel de Madurez</b>	Amplia (80)	Difundida (90)	Reciente (90+)
<b>Personalización</b>	Totalmente en función a la tecnología	En función a la tecnología y en menor grado al negocio	Mayormente en función al negocio
<b>Administración y Configuración</b>	Muy costosa	Costosa, control de versiones	Simple, Catalogo de servicios
<b>Tipo de Contrato</b>	Privado/Publico	Publico	Publicado

*Cuadro 14. Comparación entre Arquitecturas de Sistemas*

### **Perspectiva Arquitectónica**

En una arquitectura de 2 capas los aplicativos están diseñados para trabajar directamente contra el sistema de base de datos, manteniendo una comunicación directa y totalmente conectada con él, desde este punto se puede decir que no existe mayormente una visión arquitectónica mas allá que el de la aplicación y la de los datos, los “componentes” de software no tienen frontera definida, son internos al programa del cliente y se entremezclan pudiendo estar la lógica del negocio en un lado como en el otro, en una arquitectura de típica basada en componentes esto es claramente mejorado ya que la lógica de negocio reside en su propio espacio pudiendo ser un servidor de componentes y separando así la presentación, la lógica y los datos en 3 capas distintas que se comunican entres si para dar soporte a los procesos de negocio, los elementos si bien se encuentran separados, son propios de la organización pero se tiene una visión más ordenada de la arquitectura.

Aun así se dice que en este modelo los componentes son internos es decir pertenecen a un sistema y no pueden verse desde afuera de la organización, su ámbito está limitado mayormente a la organización y si se requiere interoperabilidad entre organizaciones es necesario de implementaciones tipo puente que comuniquen los sistemas entre sí. En una arquitectura orientada a servicios, el software es tratado como unidades independientes capaces de brindar un servicio de forma más autónoma, los elementos son publicables y vistos desde fuera de la frontera de la organización pudiendo ser empleados sin necesidad de ser dueño de los mismos.

## **Interfaz**

Una arquitectura tradicional cliente / servidor ya sea 2 capas o basada en componentes esta comúnmente apoyada en tecnología propietaria que repercute en los costos de implementación ya que encarece el mantenimiento de los sistemas, cada uno de ellos con sus propias interfaces de cara al cliente, normalmente en las entidades financieras encontramos software desarrollado en sistemas transaccionales que usan formularios (Visual Basic, Power Builder etc) o sistemas de tipo mainframe que usan interfaces tipo texto en las computadoras cliente (AS/400), esto para un entorno web de interoperabilidad es por sus propias características incompatible.

En una arquitectura orientada a servicios utilizando Servicios Web, el navegador es la herramienta por excelencia del cliente, garantizando que no existe otro requerimiento adicional para el manejo de las aplicaciones, basta con algún cliente de navegación (actualmente todas las computadoras disponen de ellos).Inclusive el mercado objetivo de este tipo de aplicativos puede ir mas allá porque muchas soluciones basadas en arquitecturas orientadas a servicios, incorporan integración con otros dispositivos y plataformas con sus correspondientes herramientas de usuario, por ejemplo podríamos tener operaciones a través de dispositivos móviles, handhelds, pdas, etc. Con esto ya no estamos supeditados a aplicaciones que requieran de un computador conectado a una red alambica para realizar operaciones de negocios. Ejemplo de esto pueden ser las consultas de cuotas pendientes de un crédito otorgado por nuestra institución a través de un celular o dispositivo móvil que cuente con navegación por Internet.

## **Modelo de despliegue**

En una arquitectura cliente / servidor de 2 capas, cada cambio en la aplicación requiere de un proceso de distribución, instalación y configuración en cada una de las PCs conectadas al sistema, esto representa un gasto para la organización y una altísima dependencia del departamento de tecnología en la configuración de cada cambio, Por ejemplo en nuestro caso particular en la Caja Metropolitana usando un sistema bajo esta configuración se requiere ante cada cambio que el área de Producción, envíe un correo electrónico a las agencias para que se dejen encendidos las computadoras y los equipos de comunicación para proceder a ;a actualización remota de la aplicación en cada una de las estaciones. Esto conlleva tiempo y riesgos de no tener la seguridad de que todas las maquinas cuentan con la última versión del aplicativo.

En una arquitectura basada en componentes, este problema no desaparece pero se controla mejor ya que se confina al control de las versiones de los componentes de lógica de negocio en el servidor de componentes, las actualizaciones son mas rápidas y por tanto el costo asociado es menor, sin embargo esto puede funcionar muy bien cuando el cambio es interno a la organización mas no cuando hablamos de un entorno de inter-operabilidad entre empresas, en estos casos cada cambio impacta en ambas organizaciones ya que los sistemas de ambas organizaciones deben adecuarse al software que las mantiene unidas, esto se refiere a los *middleware* que transportan los datos de una organización a otra. En estos casos se requiere de un despliegue físico de los componentes en el servidor cuidando las versiones de los mismos (versionamiento), luego de la instalación se hace disponibles los cambios. Una arquitectura SOA no requiere de un

despliegue físico ni referencias en las estaciones de los clientes, simplemente se invocan y se usan es decir no se requiere de un despliegue por cada una de los consumidores de los servicios.

### **Ámbito de intercambio de la información**

Mientras que en una arquitectura de 2 capas la información fluye siempre dentro de la empresa ya que se hace costoso implementar un proyecto que comunique a las organizaciones basándose en este tipo de arquitecturas, en una arquitectura basada en componentes esto se torna más realizable no sin contar con una estrecha coordinación previa sobre los formatos, protocolos, canales de comunicación, requerimientos técnicos de comunicación etc. (por ejemplo que puertos debe de abrir el administrador de red para permitir la comunicación).

La arquitectura orientada a servicios permite con mucha mayor simplicidad la comunicación entre empresas basándose en estándares como el de los esquemas XML, las organizaciones no necesitan mayor coordinación, simplemente se torna en un mecanismo de descubrimiento de los servicios disponibles y de consumo de los mismos, esto es ideal para comunicarse entre empresas sin repercutir en grandes proyectos de implementación.

### **Niveles de acoplamiento**

Una de las características más relevantes de los Servicios Web es la del llamado acoplamiento débil. No solo son independientes de la infraestructura en la que corren, de hecho no hacen ninguna diferencia (en contraste con la

Arquitectura basada en componentes) si un servicio corre utilizando a otro que no esté en la misma computadora o no usa el mismo lenguaje / sistema operativo. Los servicios Web desarrollados en .NET pueden fácilmente acoplarse con servicios Web desarrollados en Java y viceversa. Adicionalmente, estos pueden ser expuestos en Internet para que así cualquiera pueda usar estos servicios sin ser dueños de ellos, o necesitar copiárselos. Los Servicios Web son diseñados para ser publicados tan ampliamente como lo permite Internet. Deben ser como direcciones Web para las computadoras.

Un ejemplo popular de implementación de Servicio Web es el de implementación de uno que combine los servicios de Amazon y Google, mientras se realiza la búsqueda de un libro en Amazon, este provee de más información del libro en Google. Este ejemplo muestra una característica propia de SOA: Los servicios se pueden combinar aun no siendo diseñados para ello. Los servicios que fueron desarrollados por una razón o función específica y que fueron lanzados a la Web, pueden ser empleados para otros propósitos de los que originalmente se les había planeado. Este hecho de que los servicios puedan ser usados para otros propósitos distintos puede ser una ventaja como una desventaja. Si el servicio es usada nuevo forma, innovadora, probablemente ayude al proveedor del mismo a alcanzar nuevos clientes o nuevos negocios.

Por otro lado se vuelve una desventaja si el servicio es empleado para, de alguna manera perjudicar al proveedor del mismo, por eso no todo debe ser publicable, y su uso debe de ser regulado mediante condiciones o ciertos términos generales. El uso de los servicios para otros propósitos de los que

fueron inicialmente diseñados solo se hace posible debido al débil acoplamiento. El servicio de Google no conoce nada acerca del servicio de Amazon y viceversa, ambos son autosuficientes. Ahora bien, con Enterprise JavaBeans solo puede haber dos componentes en un container en el que uno no sabe nada acerca del otro y entonces sus interfaces se hacen utilizables. Pero el acoplamiento débil no es tal, ya que no hay frontera para determinarlo.

Muchas tecnologías son débilmente acopladas pero cada una tiene una diferente versión de lo que es acoplamiento débil. Por ejemplo ¿alguna vez se ha podido ver a los desarrolladores de Enterprise JavaBeans, usar un EJB de otra compañía, sin tener que entrar en contacto con esta? Esto solo se hace posible usando Servicios Web.

## **Comunicación**

En una arquitectura de 2 capas, la información de la aplicación fluye dentro de las aplicaciones y se limita a esta, no es posible el intercambio con otros sistemas dentro de la organización que estén desarrollados usando otro tipo de tecnología de desarrollo de software, esto se puede solventar con cierto grado de restricción usando interfaces específicamente desarrolladas ya que la información debe ser convertida a un formato común acordado previamente por los nodos que la intercambian.

En una arquitectura basada en componentes, esto es hasta cierto punto tarea de todos los días en empresas que cuentan con diversos sistemas de información, las famosas islas de información, pero se torna complicado pues no se puede extrapolar a la comunicación entre organizaciones con tanta

facilidad, aquí entran nuevamente los famosos middlewares para transportar y convertir la data a un formato común.

La arquitectura orientada a servicios se orienta más hacia internet, a la comunicación entre empresas aprovechando protocolos (HTTP), los estándares abiertos de comunicación (SOAP) y de descripción de los datos (XML). Esto transparenta la comunicación y permite el consumo de servicios web sin necesidad de interfaces de transformación ad-hoc.

## **Granularidad**

En una arquitectura orientada a servicios la atención se centra en generar los servicios necesarios a partir de las aplicaciones disponibles en lo posible o nuevas de ser necesario. La creación de servicios puede ser de grano fino, es decir un servicio individual que se corresponde con un proceso de negocio individual, (por ejemplo “insertar código de crédito”), o de grano grueso es decir múltiples servicios que van juntos para realizar una serie de funciones de negocio relacionadas entre sí, (ejm. usar varios servicios de grano fino para “procesar una solicitud de crédito”). En arquitecturas tradicionales cliente servidor no se puede hablar de granularidad gruesa ya que hasta cierto punto los componentes físicos están fuertemente ligados con la lógica de implementación, por lo que usualmente son de grano fino pues contienen procesos de negocio individual y combinarlos libremente requiere de una implementación específica. Es decir no se pueden combinar servicios preexistentes ya que no hay una orientación a servicios en este tipo de arquitecturas

La granularidad es indirectamente proporcional a la riqueza funcional de un servicio. Esto significa que entre más específica es la funcionalidad del servicio, mayor o más fina es su granularidad. Se dice que un servicio tiene granularidad gruesa, cuando tiene una mayor riqueza funcional y un servicio tiene una granularidad fina cuando hace una función específica.

Por ejemplo, un retiro es una función específica por lo que se le clasifica como una granularidad fina mientras que una función de compra de deuda tiene una granularidad gruesa o mayor porque no solo realiza un retiro o un cargo a una línea de crédito, sino también uno o varios depósitos e incluso operaciones de reversión en caso de falla.

## **Escalabilidad**

En una arquitectura de 2 capas los sistemas pueden llegar a atender a un máximo de 100 usuarios, normalmente ese es el límite de números de usuarios, después de este número la capacidad de una buena performance es excedida, esto se debe a que el cliente y el servidor “mantienen viva” la conexión de manera continua, aun si no se está realizando transacción u operación alguna, consecuentemente la red se satura.

Implementar lógica de negocio por ejemplo en stored procedures puede limitar la escalabilidad esto debido a que mas y mas lógica es movida al servidor de administración de base de datos, el poder de procesamiento del servidor debe crecer para atender a más usuarios. Cada cliente usa al servidor para ejecutar

parte de su código de aplicación y esto por ultimo reduce la cantidad de usuarios que pueden conectarse.

En la arquitectura basada en componentes la implementación de lógica de negocio se mueve a la capa intermedia y físicamente reside en un servidor de componentes, esto mejora dramáticamente la escalabilidad de las aplicaciones. Una arquitectura orientada a servicios, no sólo proporciona idoneidad de soluciones, sino que hace posible el incremento de velocidad de respuestas a cambios organizativos y estrategia dirigidas a mejorar los servicios públicos, mediante la modificación de la composición de servicios y la ampliación de los sistemas.

## **Reusabilidad**

En arquitecturas de 2 capas la reusabilidad es muy pobre ya que resulta extremadamente complicada reutilizar lógica de negocio ya implementada, esto debido a que se encuentra embebida ya sea en la capa de interfaz de la aplicación o en stored procedures de la base de datos, por lo general en las aplicaciones desarrolladas bajo este modelo, el código lamentablemente termina duplicándose en distintas partes de la aplicación y muchas veces se pierde cantidades considerables de tiempo en automatizar procesos de negocio ya que es necesario rehacer porciones de código que no se encuentran fácilmente disponibles para los nuevos requerimientos. Esto conlleva costos e ineficiencia en el desarrollo de aplicaciones.

Las arquitecturas basadas en componentes de software aportan reutilización de lógica de negocio siempre y cuando estas se encuentren bajo la

misma plataforma o lenguaje, es decir los componentes pueden ser llamados desde distintos procesos de negocio pero estos deben de ser fuertemente compatibles y esto solo es posible cuando la plataforma tecnológica (sistema operativo, lenguaje de programación, motor de base de datos) sea la misma.

En una arquitectura orientada a servicios, los servicios de grano fino son normalmente reutilizables para componer servicios de grano grueso, la reusabilidad es uno de los pilares de la arquitectura SOA y a medida en que cada proceso de negocio, incorpore repetitivamente el uso de uno o varios servicio(s) de negocio(s), se garantiza la reusabilidad del mismo. Permitiendo disminuir los costos de implementación y los tiempos de desarrollo. Los servicios no tienen dependencia de plataforma tecnológica al usar estándares como XML para comunicarse son independientes, débilmente acoplados y altamente escalables.

Los servicios deben de construirse pensando en todos los posibles escenarios de uso y no deben estar acoplados a un consumidor específico, es decir, que deben de poder ser llamados por cualquier aplicación, servicio o canal solicitante. La aplicación de la reutilización debe apoyar la visión de simplicidad para lograr la reducción de la complejidad tecnológica. A través de la reutilización se contribuye a la reducción de costos y a la agilidad del negocio.

## **Integración e Interoperabilidad**

La arquitectura de dos capas limita la interoperabilidad usando store procedures para implementar lógica compleja, esto debido a que los store

procedures normalmente se desarrollan usando un lenguaje propietario del sistema de administración de la base de datos. Esto implica que el cambiar o interoperar con más de un sistema de administración de base datos implica que la aplicación tenga que ser re-escrita en el peor de los casos. En una arquitectura tradicional cliente/servidor aparece además la figura de los sistemas propietarios, que dificultan la interoperabilidad entre sistemas de múltiples organizaciones que quieran brindar una solución de negocio conjunta o simplemente en el caso que nuestra organización requiera sacar al mercado un producto crediticio por ejemplo que necesite de la integración de sus aplicaciones tecnológicas que normalmente son islas dentro de la organización, y que deben trabajar juntas para completar el flujo de negocio.

Se necesita frecuentemente de un tiempo de análisis y desarrollo considerable lo que fácilmente (con una mala estimación) puede impactar en lo que se denomina "time to market" es decir el tiempo que se demora la empresa para sacar al mercado una solución de negocio. En nuestro caso si deseamos ofrecer la posibilidad de que otras Cajas Municipales puedan brindar productos financieros a través nuestra empresa requeriríamos de conocer bien las plataformas de cada uno de nuestros socios de negocio, desarrollar interfaces, o incluso depender de terceros (como ya ha sido el caso en el pasado) para que desarrollen estas interfaces por medio de software que enlacen ambos sistemas , normalmente conocidos como *middlewares* pero que representan un alto costo de inversión para una entidad de la envergadura de la Caja Metropolitana.

Con una arquitectura SOA, la integración se basa en utilizar estándares abiertos tales como XML, SOAP, WSDL, no se requiere de conocer las

plataformas de los consumidores de los servicios, ni en que lenguaje se encuentren desarrollados sus sistemas. Incluso enfocándonos en el entorno interno de la empresa podemos llevar el enfoque hacia las islas de software que existen dentro de la organización, ya que no es necesario empezar de cero para conseguir que estas interactúen para dar soporte a un nuevo proceso de negocio, bastara con adecuarlas y reutilizar las funciones existentes encapsulándolas de tal manera que puedan ser convertidas en servicios que puedan ser consumidas desde por ejemplo la intranet.

### **Adecuación a los requerimientos del negocio**

Normalmente en los sistemas de información basados en arquitecturas tradicionales, se produce una demora sistemática para atender las necesidades de los usuarios, consecuencia de la excesiva dependencia del personal de tecnología para realizar las personalizaciones que éstos demandan, los requerimientos deben pasar por una serie de pasos a través de un flujo de atención que frecuentemente genera colas para que el personal de sistemas pueda atenderlas, esto conlleva a mayor tiempo y muchas veces a que cuando se tiene lista la modificación del sistema y esta ya cuenta con el visto bueno del área de control de calidad, es muy tarde para el negocio, y por tanto se perdió la posibilidad de negocio que existía y que fue la que origino el cambio.

Mientras en una solución basada en arquitectura orientada a servicios la adecuación es mucho más rápida gracias a que la personalización puede ser realizada por los responsables de negocio que ya conocen los servicios que pueden conjugarse para dar origen a una solución, por ejemplo en el caso de la necesidad de crear un nuevo producto crediticio cada parte constituyente

puede estar dentro de un Servicio Web, que luego puede unirse con otros para añadir complejidad al proceso, con lo que no existe demasiada demora para satisfacer las necesidades de los usuarios.

## **Madurez**

Se considera que el software tradicional tiene una más amplia y profunda funcionalidad que es debida en gran parte a su larga historia (hay que considerar que por esta arquitectura han desfilado cuatro generaciones de lenguajes de desarrollo de sistemas). Las arquitecturas de 3 capas basadas en componentes por ejemplo han sido exitosamente usadas desde los inicios de los 90s en cientos de sistemas de diversos tipos en la industria comercial, donde la distribución de información en entornos heterogéneos se hace necesaria, estas arquitecturas han sido ampliamente aplicadas en algunos de los más grandes servidores de Internet por ejemplo.

Ahora bien, las frecuentes actualizaciones que se pueden conseguir al involucrarse en una arquitectura orientada a servicios ayudan a conseguir en mucho menor tiempo una funcionalidad equiparable a la del software tradicional que lleva más tiempo establecido, ya que esta arquitectura continua evolucionando debido al auge de las comunicaciones y el acortamiento de las distancias entre empresas gracias a Internet. La funcionalidad que pueda brindar un sistema basada en una arquitectura orientada a servicios se limita únicamente por la gama de procesos articulados que la empresa planea exponer, como ya hemos mencionado anteriormente no se está ligado fuertemente a un sistema propietario que limite la funcionalidad.

## **Personalización**

En una aplicación basada en arquitectura tradicional cliente / servidor, la personalización esta estrechamente ligada a la plataforma tecnológica bajo la cual esta implementada y ha de ser realizada por personal con conocimientos profundos de los mecanismos de configuración de la aplicación y muchas veces con conocimientos del lenguaje de programación para la aplicación.

Mientras que una arquitectura orientada a servicios está diseñada hacia el negocio, no hay olvidar que lo que da origen a usar este tipo de arquitectura es la necesidad que tiene la organización de dar soluciones rápidas de negocio, y que ante cualquier cambio que pueda ser realizada por personal sin conocimientos de programación mediante mecanismos de configuración fáciles de aprender, es decir que no se requiera de cambios considerables en el código de la aplicación.

## **Administración y Configuración**

Las arquitecturas de dos capas pueden ser difíciles de administrar y mantener debido a que cuando las aplicaciones residen del lado del cliente (cliente grueso), cada actualización del sistema debe ser distribuida, instalada, e incluso testeada en cada cliente. La carencia de uniformidad den las configuraciones de las estaciones clientes y la falta de control sobre los subsecuentes cambios de configuración pueden incrementar la carga de trabajo administrativa.

En el caso de las arquitecturas basadas en tradicionales componentes cliente servidor, el costo es menor debido a que un cambio se realiza solo en el componente específico que contiene la lógica de negocio pero aparece al mismo tiempo el problema del versionamiento de los componentes, los cuales si se carece de un eficiente control pueden terminar siendo un dolor de cabeza para los administradores de los servidores de componentes. En el caso de las arquitecturas orientadas a servicios estos son más fáciles de manejar pues se catalogan y se identifican claramente, aunque también se requiere de una buena administración para evitar la proliferación desordenada de los mismos

### **Tipo de Contrato**

Los contratos definen los mensajes de entrada y salida entre los servicios, dentro de las fronteras donde reside el servicio los datos pueden estar estructurados de manera distinta. En las arquitecturas cliente servidor de dos capas los contratos entre componentes de software son privados pues al no tener una orientación a servicios (no tienen que ser públicos, sino solo parte de la aplicación) es potestad de los desarrolladores de software el definir los mensajes de entrada y de salida, este conocimiento es interno de quien(es) implementan la aplicación de la empresa y no existe mayor detalle en él.

Al evolucionar hacia componentes las implementaciones se tornan más ordenadas y se define mejor cuáles son las interacciones entre los componentes y como por ejemplo debe enviarse información y devolverse entre una aplicación y sus componentes, esto implica un conocimiento público de los mismo pero normalmente interno a la organización pero pocas veces

documentado. Mientras que en una arquitectura orientada a servicios los contratos se vuelven fundamentales incluso se realizan a través de un lenguaje como WSDL que describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Así, WSDL se usa a menudo en combinación con SOAP y Esquemas XML. Un programa cliente que se conecta a un servicio Web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de Esquema XML. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL.

Básicamente el contrato de servicios típico debe contener:

- Encabezado
- Nombre del servicio.
- Versión del contrato de servicio.
- Propietario: Persona o equipo encargados del servicio
- RACI
  - Responsable de la entrega del servicio.
  - Accountable (Dueño) del servicio.
  - Consultado: Quien debe ser consultado antes de tomar cualquier acción sobre el presente contrato/servicio.

- Informado: quien debe ser informado sobre cualquier decisión o acción que se vaya a tomar sobre el presente contrato/servicio.
- Tipo: Es el tipo de servicio, ayuda a distinguir en que capa residirá.
 

Diferentes implementaciones pueden tener diferentes tipos de servicios

  - Datos
  - Procesos
  - Negocio
  - Presentación
  - Integración
- Funcionalidad
  - Requerimientos funcionales: Indica específicamente la funcionalidad que este servicio debe proveer.
  - Operación del servicio: Se debe definir en términos de que parte de la funcionalidad se provee (métodos usados, acciones etc.).
  - Invocación: Indica los medios como se invoca o llama al servicio, esto puede ser URL, interfaz, triggers etc. Pueden existir múltiples caminos para invocar un mismo servicio.
- Requerimientos no funcionales, tales como
  - Seguridad: Roles, mecanismos de invocación etc.
  - Calidad del servicio: Determina el nivel de tolerancia de fallos del servicio.
  - Transaccionales: Es este servicio capaz de realizar gran cantidad de transacciones? cuantas? bajo qué condiciones?
  - SLA: Determina la máxima latencia permitida para el servicio bajo la cual se pueden realizar acciones.

- Semántica: Define el glosario de términos usadas en la descripción e interfaces del servicio.
- Procesos: Describe los procesos (si existen) relacionados con el servicio contratado.

## **SOA VS CORBA, Comparativa entre arquitecturas Interoperables**

En este apartado compararemos las arquitecturas más ampliamente difundida, que ofrecen soluciones de interoperabilidad sobre entornos heterogéneos ya que no se puede comprar SOA con arquitecturas propietarias como el modelo de componentes de Microsoft DCOM o el modelo RMI de invocación remota de Java, ya que estas últimas trabajan sobre entornos homogéneos únicamente.

Ambas usan mecanismos de comunicación estándar. CORBA usa IIOP (Internet Inter ORB Protocol), los Servicios Web de SOA usan SOAP. CORBA y los SOA tienen lenguajes de definición de Interfaces, el de CORBA es el IDL (Lenguaje de Definición de Interfaces), y en el mundo de los SOA es el WSDL (Lenguaje de Definición de Servicios Web).

<b>Item</b>	<b>SOA (Web Services)</b>	<b>CORBA</b>
Protocolos	SOAP, HTTP, XML Schema	IIOP, GIOP
Identificadores de	URLs	IORs, URLs

Localización		
Especificación de Interfaces	WSDL	IDL
Directorio	UDDI	Naming Service, Interface Repository, Trader service

*Cuadro 14. Comparación entre SOA y CORBA*

### **Protocolo de Transporte**

La comparativa es complicada de de hacer ya que los ambos, envían y reciben peticiones, pero SOAP destaca al ser el único que puede pasar de forma “natural” por los firewalls y así quitar el problema de la invocación de módulos remotos, las conexiones no requieren de abrir un puerto especial ya que se realizan por el conocido puerto 80 de conexión a Internet pues está basado en http. Esto simplifica la Gestión de puertos ya que no se necesita abrirlos específicamente por cada proyecto de implementación fuera de la frontera de la empresa como sucede con otras tecnologías.

### **Grado de Complejidad**

Cuando se estudian las características de una plataforma, muchas veces se pasa por alto el grado de complejidad, en su aprendizaje, CORBA

resulta ser un protocolo considerablemente difícil de aprender, complejo por su IDL y su incompatibilidad entre aplicaciones.

En lo que concierne a SOAP es un asunto diferente, es un protocolo de comunicación relativamente fácil ya que basado en XML, (recordemos que este lenguaje de programación peca de ser sumamente ordenado) que permite la comunicación entre componentes mediante HTTP.

### **Adaptación a los lenguajes de programación**

Si bien en la actualidad “adaptación” es la palabra clave para muchas cosas, esta no pasa inadvertida para los protocolos de comunicación. Para que un protocolo pueda ser aceptado de manera global por los diferentes lenguajes y plataformas debe ser abierto y aceptar programas u aplicaciones en distintos lenguajes.

Para CORBA esto no es difícil pero peca en los estándares que maneja ya que al no ser estándares conocidos se limita, , SOAP por otra parte no está peleado con ninguna plataforma de desarrollo, así que no tiene problema y se considera un protocolo abierto y adaptable.

### **Lenguaje de especificación de Servicios.**

IDL es el lenguaje específico de Servicios que se ha creado para implementar las especificaciones en CORBA, IDL se ofrece la sintaxis necesaria para la invocación de los métodos que se requieren de manera remota. SOAP utiliza WSDL que describe los tipos de datos, las funciones

exportadas y sus mensajes de petición y respuesta de manera si no sencilla, entendible y bien estructura (recordemos que WSDL también utiliza XML).

### **Invocación de Métodos.**

Los protocolos de comunicación obviamente tienen que invocar métodos remotos pero hay algunas diferencias entre ellos y la manera de hacerlo. SOAP sobresale por su manera de invocación de sus métodos remotos, ya que lo hace de manera ordenada y estructurada (utiliza HTTP y XML, HTTP+XML= SOAP).

### **Capacidad de comunicación entre sus componentes.**

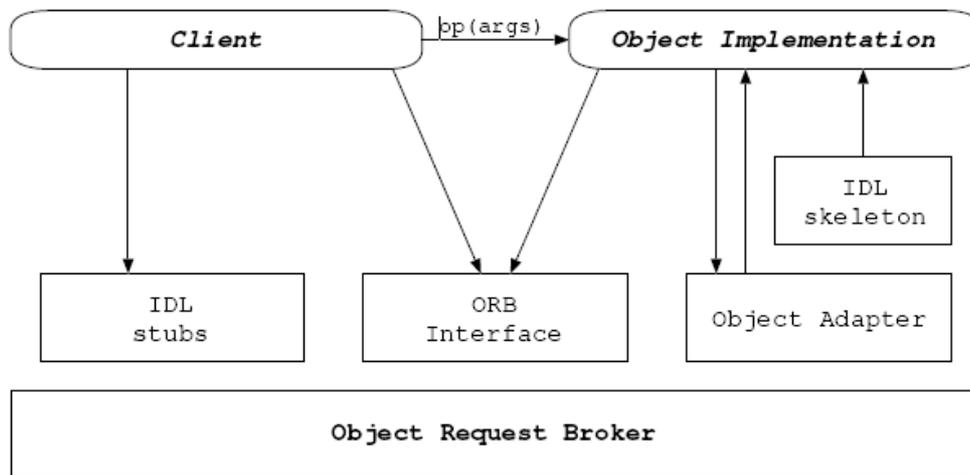
En este punto se analiza la capacidad de adaptación entre sus componentes es decir en la estructura misma, CORBA aunque tiene la capacidad de ser libre entre sus componentes, es muy problemático y complejo para permitir la interoperabilidad entre distintos lenguajes, otras arquitecturas y sistemas operativos.

SOAP, por definición de estructura con lleva la interoperabilidad entre distintos lenguajes en sus raíces, si bien es parecido a CORBA, lo supera en la facilidad de interoperabilidad con la que trabaja.

### **CORBA:**

Desventajas fuertes: Firewall, falta de una implementación que cumpla el estándar correctamente, esto produce problemas de integración con otras implementaciones, etc. Y el problema es que CORBA en si es complejo de

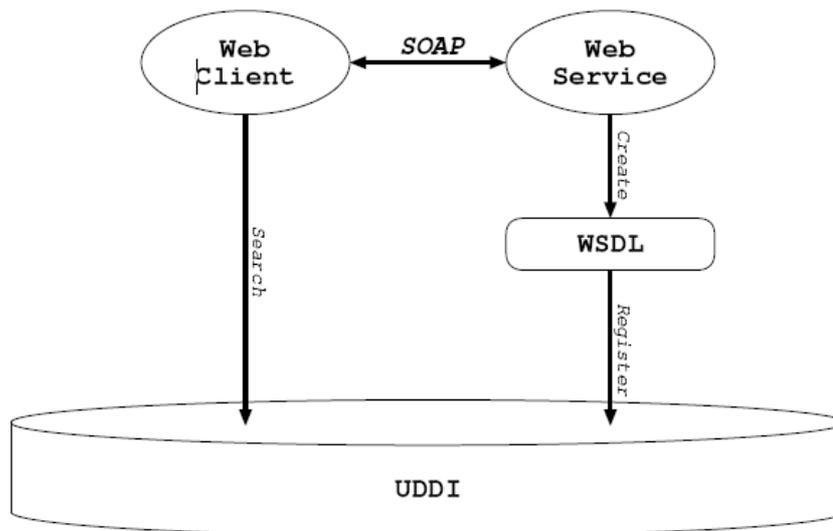
implementar. La figura 26 muestra los componentes principales de esta tecnología



*Figura 26: Componentes en el Modelo de Referencia CORBA 2.x*

## SOA

Los Servicios Web en SOA, son realmente independientes de las plataformas, lenguajes, etc. Su implementación no es compleja, para integrarse con terceros es muy efectiva. Y además de esto funciona sobre XML heredando así sus tan buenas propiedades que le han dado tanto éxito. La figura 26 muestra la interacción entre los componentes de un Servicio Web



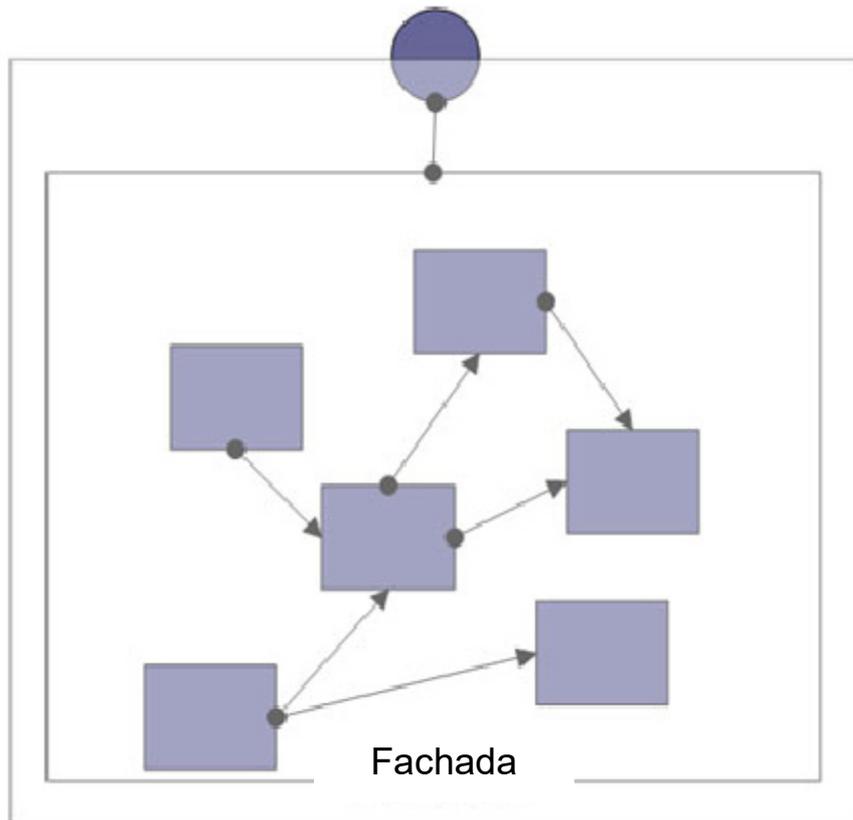
*Figura 27: Componentes en un Servicio Web.*

### **Conclusión de la comparación**

Las arquitecturas basadas en componentes y las arquitecturas orientadas a servicios parecen tener el mismo objetivo: Proveer una modelo para servicios de negocios altamente ínter-operables y débilmente acoplados. Las últimas evoluciones recientes han tenido la misma intención: El desarrollar un tipo de arquitectura que permita débil acoplamiento y alta reutilización de componentes, deben permitir más eficiencia, rapidez y producción de software libre de errores. En términos más abstractos son pasos evolutivos que mejoran el paso anterior y ayudan a acercarse más a los objetivos.

Es por eso que el paso de la Arquitectura basada en componentes a la Arquitectura orientada a servicios es más una evolución que una revolución.

Un ejemplo de porque este es un paso evolutivo y porque no existe una frontera absolutamente clara entre la Arquitectura basada en componentes y la Arquitectura orientada a servicios es el siguiente:



*Figura 26: Fachada para ofrecer servicios de Grano Grueso*

A pesar que el software basado en componentes da salida a muchas tecnologías de objetos distribuidos, no es posible distribuir objetos llamados “de grano fino” sin causar un gran impacto en al menos algún requerimiento no funcional. Como el método de invocación local es mucho más rápido que el de una invocación remota, solo pueden exponerse a la red objetos de “grano grueso”, por lo que los componentes reutilizables de lógica de negocio deben de permanecer de “grano fino”.

Una solución a este problema es usar patrones “de fachada”, tal como se ilustra en la figura 26: No podemos publicar entidades de grano fino a los clientes, así que debemos proveerles de una vista de grano grueso. Por otro lado, se hace muy costos cambiar las interfaces de las entidades, así que

debemos proveer un elemento adicional que a su vez provea de una vista distribuida del sistema.

Esta fachada provee a los clientes de una vista del sistema con la que pueden trabajar. Se pueden diseñar varias fachadas según las distintas demandas del sistema. Esta idea es similar a la que se encuentra detrás de los Servicios Web: Abrir el sistema en puntos específicos y definidos de forma precisa, a los clientes o a los socios de negocio.

#### 4.1.2. Soluciones existentes de Interoperabilidad en la Caja Metropolitana

Actualmente en el ámbito financiero nacional no se tiene un conocimiento de casos de éxito aplicando orientación a servicios y dado el carácter confidencial muchas veces de las instituciones financieras –como es natural-, este tipo de información resulta inaccesible. Por otro lado, en la organización materia de estudio, es decir la Caja Metropolitana, no existe ninguna solución de tecnologías de información que se aproxime a una orientación de servicios. Si bien en el ámbito de la interoperabilidad con otras instituciones, existen soluciones implementadas y que se mantienen en funcionamiento en la organización. A continuación presentamos dos claros ejemplos de cómo se viene trabajando la interoperabilidad como estado del arte desde dentro de la organización:

#### **Proyecto de Interconexión entre la Caja Metropolitana y otras Cajas Municipales, usando software middleware**

*Duración del proyecto: 5 meses*

*Periodo de Implementación:* Julio 2005 – Noviembre 2005

*Objetivo:* Permitir intercomunicar los sistemas informáticos de la Caja Municipal del Cusco y la Caja Municipal de Paita con los de la Caja Metropolitana de Lima, para permitir a los clientes de las primeras realizar operaciones de diversa índole (pagos de servicios, retiros/depósitos en cuenta, etc.) en la red de agencias de la Caja Metropolitana.

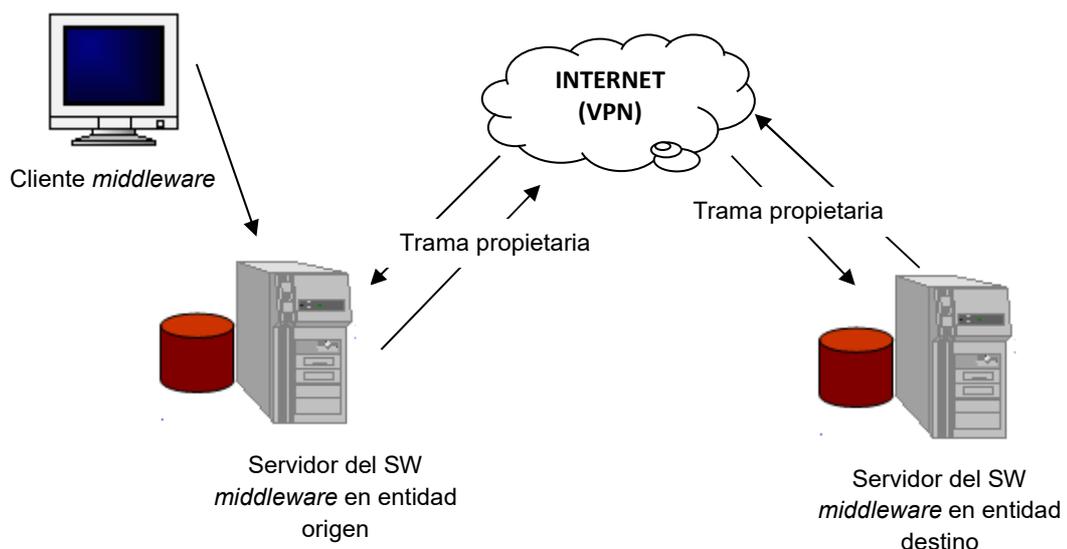
*Situación:* Esta solución se encuentra actualmente operativa en la empresa

Este proyecto contemplo la implementación de una red de comunicación e intercambio de tramas de mensajes entre las instituciones financieras que deseaban realizar un convenio con nuestra institución, empleando para ello conexiones de tipo punto a punto a través de una Red Privada Virtual (VPN)

Para esto se debía considerar un medio de intercambio entre los aplicativos de cada una de las Cajas Municipales en cuestión. Actualmente ambas Cajas Municipales -tanto Cusco como la de Paita - cuentan con un sistema informático similar al de la Caja de Lima que soporta sus procesos de negocio desarrollado en lenguaje Visual Basic y Visual Basic. NET bajo plataforma Windows, esto debido a que este sistema core de negocio fue adquirido en conjunto hace ya varios años, si bien cada entidad financiera presenta modificaciones con respecto al sistema original y nuevos desarrollos propios de cada uno, los procesos principales de negocio se mantienen similares.

Así era claro que para poder implementar este proyecto se necesitaba de un canal común que permitiera comunicar los sistemas involucrados.

Debido a que esta propuesta de negocio, tuvo su origen en la necesidad estratégica del negocio de las Cajas Municipales involucradas (Cusco, Paita, Lima) fueron estas en común acuerdo las que –en base a experiencias previas que todas tenían con un proveedor de software común – establecieron un modelo de interconexión mediante el uso de un software de tipo middleware que eliminara la barrera que presentaba la implementación de una adecuada comunicación de las aplicaciones de sus sistemas con el nuestro. Esta solución se basaba en el intercambio de tramas de mensajes los cuales contenían los datos necesarios de las operaciones que se registrarán en las ventanillas de nuestra red de agencias, llegaran en línea a los servidores de cada una de las Cajas Municipales (Cusco y Paita), para que a su vez estas reflejen los movimientos en sus propios sistemas. Es decir se maneja un esquema donde los datos registrados en un lado, se deben reflejar (“mirror”) en otro lugar para así mantener la consistencia de las operaciones registradas, tal como se puede apreciar en el siguiente grafico:



*Figura 27: Modelo de sistema de interconexión entre entidades usando middleware*

Esta solución si bien es válida y se mantiene en funcionamiento, ha presentado a lo largo de su funcionamiento y presenta hasta la actualidad, varios puntos desfavorables, en grandes rasgos están los siguientes:

- Dependencia de un tercero en el aspecto tecnológico, en este caso, las tres Cajas Municipales dependen del proveedor de software middleware, debido a que es este el que maneja la sincronización del envío y recepción de los mensajes (tramas) que viajan a través de la red. Si este aplicativo fallase se hace necesaria la intervención del proveedor del software intermediario ya que la lógica de funcionamiento es propia de este, siendo independiente y cerrada. Esto acarrea costos ocultos derivados de la pérdida de capacidad de atender operaciones durante el periodo en el que el proveedor toma en diagnosticar y solucionar los posibles inconvenientes.
- Existencia de redes privadas virtuales para la comunicación, debido a los requerimientos del software del proveedor, se hizo necesario mantener conexiones de tipo VPN (red privada virtual) de tipo punto a punto para conectar los servidores de las organizaciones involucradas. Los servidores VPN poseen un vínculo permanente a Internet y aceptan las conexiones vía Internet provenientes de los servidores remotos. Las Cajas Municipales se conectan a Internet usando los servicios de su proveedor local de Internet, típicamente mediante conexiones de banda ancha. Esto requiere de un tratamiento específico para cada caso y el

uso de protocolos de red como IPX/SPX para la conectividad, los cuales ya han sido reemplazados por el TCP/IP

- Costo de licenciamiento del software middleware, Ya que cada servidor requiere contar con la misma versión de este aplicativo para poder operar, es decir si el proveedor decide realizar modificaciones a su producto, este deberá ser actualizado en cada una de las entidades que la utiliza de otro modo se corre el riesgo que determinado nodo quede desactualizado y no operativo. Asimismo se requería de una coordinación para establecer una ventana de tiempo común a todas las entidades para que el proveedor pueda actualizar la versión del middleware, lo cual de por sí ya implica complicaciones operativas.
- Reconstrucción de partes de cada uno de los sistemas involucrados, para poder adecuarse al esquema de tramas de información y serialización de los datos, es decir se debió reescribir el código para operaciones de Retiros, depósitos, pagos de créditos, pagos de servicios, etc. En general de toda transacción involucrada para que generara las tramas correspondientes bajo el esquema que el proveedor del servicio proponga de otro modo no se hacen compatibles.

Alternativa bajo una Arquitectura Orientada a Servicios:

Pero esta solución pudo reemplazarse por una que implique menor costo, menor tiempo de implementación y sin la dependencia de terceros, esto

bajo un enfoque orientado a servicios, de haber existido una arquitectura SOA en la organización y un acuerdo previo entre las partes involucradas, exponiendo servicios a través de un lenguaje común y estándar como XML. También la transferencia de los datos se realizaría sobre Internet, con las adecuadas técnicas de encriptación de los datos, pero con la ventaja de no tener que volver a codificar nuevamente partes de código de las aplicaciones, sino reutilizando sus funcionalidades y publicándolas vía Servicios Web de modo que fueran rápidamente publicables y reutilizables no solo por este proyecto sino dejando sentada la oportunidad de poder emplearlos en otros proyectos similares.

Asimismo al usar protocolos estándares y abiertos como SOAP esto permitiría la interoperabilidad de las aplicaciones de las tres entidades, sin la necesidad de usar ningún software intermedio que transforme los datos mediante una interfaz rígida. Esto repercute directamente en los costos de implementación

Un ejemplo de cómo se manejaría una operación mediante una arquitectura orientada a servicios podría ser el retiro de una cuenta de ahorros por parte de un cliente de la Caja Municipal del Cusco en una agencia de Lima, el sistema de la Caja Metropolitana de Lima invocaría un servicio Web provisto por la Caja de destino, mediante el cual enviaría la información de la operación en lenguaje XML a través de SOAP y debidamente encriptado, directamente al servidor remoto el cual de manera transparente a nuestro sistema realizaría el cargo en la cuenta de ahorros del cliente, luego una vez completada esta parte de la operación, el propio servicio invocaría a un segundo servicio Web (este expuesto por la Caja Metropolitana de Lima), el cual realizaría de manera

independiente el cargo en una cuenta de ahorros previamente abierta perteneciente a la Caja de Cusco en nuestra institución para poder entregar el monto correspondiente al cliente . Así ya se cuenta con dos servicios Web de cada institución que fácilmente pueden reutilizarse y acoplarse con otros, para otra operación que tal vez no tena que ver con un retiro sino por ejemplo una transferencia entre cuentas, sin necesidad de reescribir el código de la aplicación.

### **Proyecto de transferencias interbancarias entre la Caja Metropolitana y otras entidades financieras**

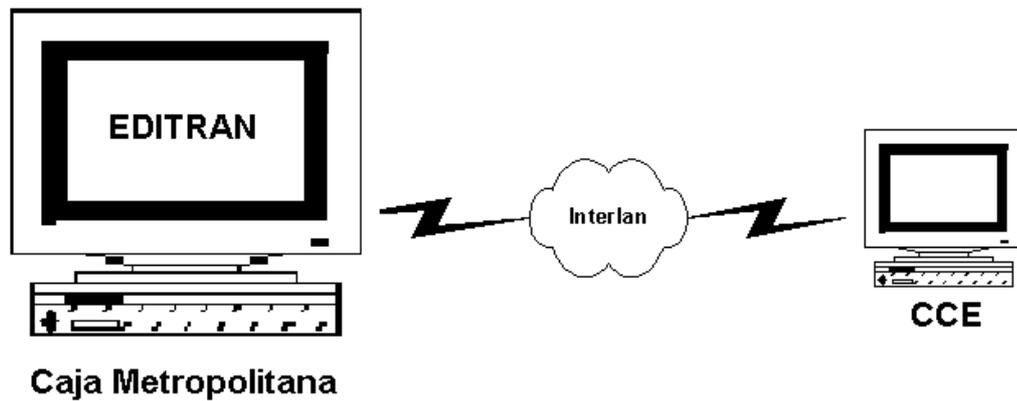
Para el caso de las Transferencias Interbancarias el sistema se basa en la compensación electrónica de la información correspondiente a transferencias, lo cual implica el intercambio electrónico de los datos entre las entidades Participantes Originantes y las Entidades Participantes Receptoras de transacciones a través de la Cámara de Compensación Electrónica; este sistema es abierto, en el sentido que un cliente (cliente originante) con cuenta o sin cuenta que desee realizar transferencias de fondos entre distintas Entidades usando el Sistema de Transferencias de Crédito pueda realizar estas transacciones a través de abonos en las cuentas de los clientes o pago a los clientes sin cuenta ( Cliente Receptor) de Entidades ( Entidades Receptoras) distintas a donde se ordena la transferencia ( Entidad Originante).

El sistema consta de una computadora ubicada en la Caja Metropolitana, donde reside un software propietario denominado EDITRAN que se encarga del Envío, Recepcion y Sincronizacion de los Archivos de Transferencias que

se envían y se reciben; en la Cámara de Compensación Electrónica poseen un servidor que se encarga también de recibir y enviar la información de las transferencias de todas las entidades participantes, previo proceso de consolidación y clasificación de los archivos recibidos. Una vez hecho ello se envía al BCRP el retiro de las transferencias presentadas por un monto suficiente para adecuar la Posición Neta multilateral a los recursos disponibles de las Entidades participantes en el BCRP, también se procede al envío de la información estadística y de control al la SBS y el mismo BCRP.

Entre las desventajas que tenemos de esta solución tenemos:

- El costo del contrato en que se incurre para usar esta solución
- Si en caso fallara el envío de los archivos desde al Cámara de Compensación Electrónica, se debe solicitar al Operador de dicha entidad el envío de la información que depende de 2 factores: si se está dentro de la ventana horaria de envío y recepción de archivos se envía a través del software EDITRAN caso contrario se envía a través de Correo Electrónico
- Si en caso la PC de la Caja Metropolitana fallara, se necesitaría del soporte Tecnológico proporcionado por la Cámara de Compensación Electrónica.
- LA información de las transferencias no está en Línea.



*Figura 28: Modelo de sistema de interconexión entre la Caja Metropolitana y el CCE*

Para evitar este engorroso proceso se propone el Desarrollo de Servicios Web, donde cada entidad participante consume Servicios Web de cualquier otra entidad participante para la ejecución de transferencias interbancarias, ello trae consigo de que las transferencias estarán en línea. Asimismo en los reportes que se envían a la SBS y BCRP cada entidad pone a disposición de dichas entidades un Servicio Web que se encarga de proveer aquella información todo a través de XML.

## **4.2. Arquitectura Orientada a Servicios Propuesta**

### **4.2.1. Objetivos de la Arquitectura**

Así, tenemos que los principales objetivos de la Arquitectura orientada a servicios que se buscan alcanzar son:

- Estabilidad Operativa

Los servicios deben de garantizar realizar correctamente las tareas de las cuales son responsables con el nivel de rendimiento determinado.

- Reducción de Costos

Los servicios deben permitir ser reutilizados, servicios no deben de estar acoplados a sus consumidores, es decir, que deben de poder ser llamados por cualquier aplicación, servicio o canal solicitante.

- "Time to Market"

Los servicios deben ser diseñados para que sean flexibles. En el estado óptimo debe permitir adaptarse a los cambios de negocio dinámicamente, evitando cambiar el código fuente.

- Autonomía tecnológica

Los servicios serán administrados por la empresa y así se evitara la dependencia tecnológica de un proveedor de software de interoperabilidad, como es el caso actual con los middlewares.

#### 4.2.2. Criterio de elección entre arquitecturas SOA

Actualmente se viene trabajando en la organización que es materia de estudio con un enfoque de arquitectura de tipo cliente/servidor, pero este esquema no es el apropiado cuando hablamos de una orientación Web, es decir resulta altamente costoso utilizar una arquitectura cliente / servidor ya que por naturaleza la comunicación debe ser estándar y como sabemos en la mayoría de casos el uso de esta arquitectura requiere de una fuerte

dependencia de la tecnología de implementación de los sistemas, por ejemplo en el diseño de sistemas transaccionales tradicionales tenemos que cada instalación de un cliente requiere de un compilado de código propio de cada cliente, es decir de un instalador en cada máquina que lo requiere y que además depende de la versión y plataforma tecnológica que la organización haya implementado.

Tampoco existe mayor grado modificación de la elección en la herramienta de software que se utilice para desarrollar los sistemas, se tiene una gran dependencia de la tecnología, el lenguaje, la infraestructura, los protocolos de comunicación etc., y aunque se tiene una fuerte ligazón ya que se implementan interfaces para hacer específicas para hacer “conversar” a los sistemas se requiere de grandes esfuerzos de programación cuando se producen cambios en las reglas de negocios lo que es consecuencia del alto grado de rigidez de estos sistemas.

Por otro lado en una arquitectura orientada a servicios se elimina la figura de la instalación de código compilado que resida en el cliente ya que lo único que se requiere es de un navegador para acceder al software expuesto como servicio, asimismo se corta la dependencia tecnológica, es decir los sistemas pueden interactuar sin necesidad de establecer interfaces duras que requieren ser modificadas cada vez que ocurre un cambio, simplemente se establece un contrato de utilización del servicio Web y mediante la parametrización se accede a la funcionalidad, cada componente de software que se encuentre detrás del servicio puede ser implementado en el lenguaje más adecuado para cada caso específico o según el perfil del personal con que

cuenta el área de sistemas, al final el servicio es una caja negra para el consumidor del servicio.

Esto es especialmente útil cuando hablamos de exponer partes del sistema a las demás entidades financieras que lo requieran porque no necesitamos conocer como esta implementado su cliente ni ellos necesitan saber cómo esta implementado nuestro sistema, se establece un medio único un canal único estandarizado que a través de XML permite la conversación de los sistemas. En la figura 34 mostramos el diagrama de secuencia de cómo se origina, publica y consume un servicio Web, esto es solo posible a través de una arquitectura tipo Web

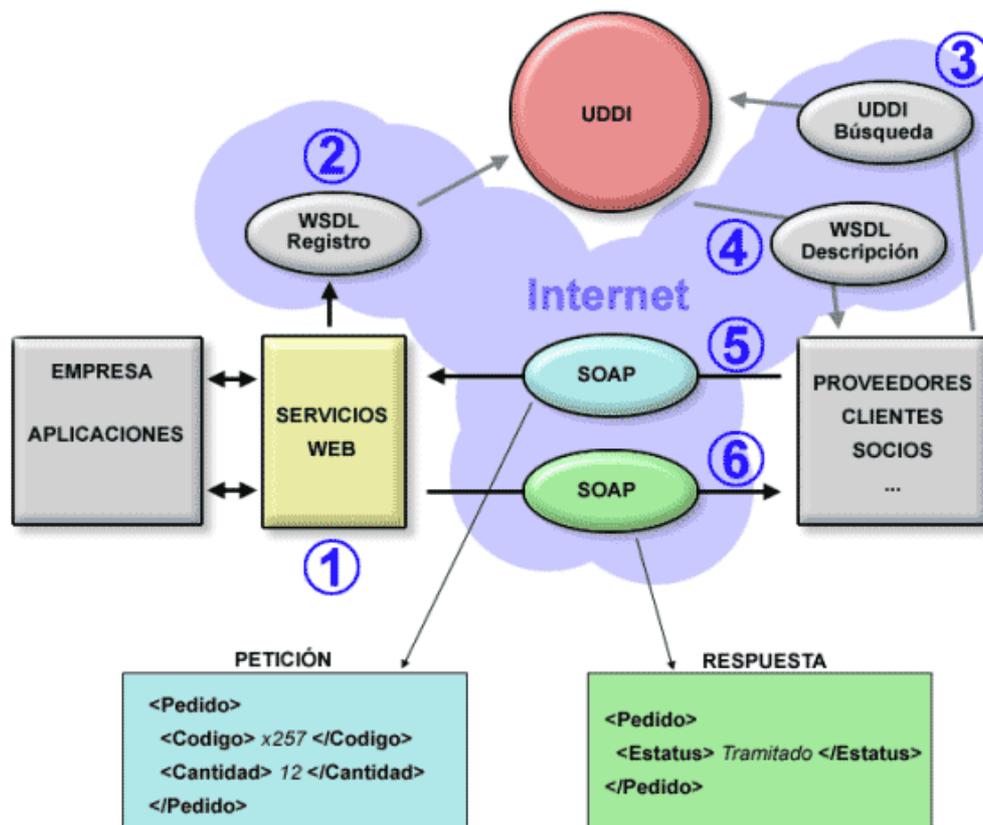


Figura 29: Diagrama de Secuencia de utilización de un Servicio Web

En una arquitectura cliente / servidor es innegable por la misma razón de la dependencia tecnológica que las nuevas versiones del software desarrollado requieren tener en cuenta las distintas posibilidades de los clientes en cuanto a la plataforma tecnológica que poseen, es decir se requiere una plataforma siempre homogénea. Al realizar la actualización de la versión del software todo esto debe tenerse en cuenta, si los requerimientos del nuevo sistema cambian también deben de hacerlo los clientes, por ejemplo si la empresa decide en un futuro próximo por motivos estratégicos de mercado aliarse con otra institución para brindar una serie de servicios a los clientes de la Caja Metropolitana, esto requerirá de toda una serie de pasos para adecuar los sistemas de ambas (si la cantidad de entidades es mas de dos, se multiplica la complejidad) para conversar entre ellas.

Esto como es sabido representa una gran cantidad de tiempo, consumo de recursos tanto humanos como tecnológicos, un alto grado de coordinación sobre todo si las plataformas de ambas organizaciones son poco homogéneas y por supuesto un alto grado de riesgo en las implementaciones debido a descoordinaciones o interpretaciones erróneas de requerimientos tecnológicos. Con una arquitectura orientada a servicios se obtiene el beneficio inmediato de simplemente conocer dónde encontrar un servicio (UDDI) y como consumirlo según el contrato del Servicio Web expuesto, esto mediante el lenguaje WSDL que describe la interfaz pública a los servicios Web, está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje. Así, si usamos WSDL en combinación con SOAP y esquemas XML un programa cliente que se conecta a un Servicio Web puede leer el WSDL para determinar que funciones están disponibles en el servidor. Los tipos de datos especiales se incluyen en el archivo WSDL en forma de esquemas XML. El cliente puede usar SOAP para hacer la llamada a una de las funciones listadas en el WSDL. El ciclo evolutivo del software es mucho más rápido al tener servicios débilmente acoplados independientes de la plataforma tecnológica de la organización con la que decidamos aliarnos o de realizar operaciones entre ellas.

### **Casos de Uso vs Escenarios de Arquitecturas SOA**

Teniendo en cuenta los cinco escenarios de arquitectura SOA, se puede confeccionar un cuadro de doble entrada que detalla la relación entre escenarios y casos de uso genéricos que nos ayudaran a identificar bajo que escenario nos encontramos en nuestro problema.

*U1: Creación o reutilización de lógica existente de aplicaciones como un servicio dentro de la empresa*

Existe funcionalidad que se desea exponer como servicio para consumo interno. Esta funcionalidad puede residir en una aplicación existente o en una nueva. Exponer funcionalidad de una aplicación como servicio provee una aplicación basada en estándares e interfaces bien definidas. Otras aplicaciones internas pueden usar estas interfaces para interactuar con la funcionalidad de

la aplicación, lo que promueve el reuso de la funcionalidad de la aplicación en la empresa, puede reducir el costo promedio de mantenimiento y hace disponible la funcionalidad a un amplio rango de aplicaciones.

Un ejemplo de este caso de uso puede ser la exposición de una aplicación .NET como servicio web (proveedor del servicio). Las aplicaciones de la empresa utilizan un cliente proxy para invocar el servicio web (consumidor del servicio).

*U2: Creación o reutilización de lógica existente de aplicaciones como un servicio fuera de la empresa*

Existe funcionalidad que se desea exponer como servicio para ser accedida por una aplicación externa en una organización extendida. Esta funcionalidad puede residir en una aplicación existente o en una nueva. Los mismos beneficios que provee una aplicación basada en estándares e interfaces bien definidas aplican a este caso de uso. Adicionalmente este caso de uso acarrea el beneficio de ofrecer esta funcionalidad a una gran variedad de aplicaciones externas.

La exposición de esta funcionalidad introduce un requerimiento no funcional a la empresa, Por ejemplo el acceso restringido a la aplicación y las medidas de seguridad correspondientes para proteger el flujo de datos hacia adentro y fuera de la organización.

Un ejemplo común de este caso de uso sería el de la exposición de una aplicación .NET como servicio web, con la aplicación de configuración WS-Security para la confidencialidad e integridad de los datos. Las aplicaciones

fuera de la frontera de la organización invocarían al servicio web usando la figura de las firmas digitales y una llave pública.

*U3: Integración punto a punto de aplicaciones mediante el uso de servicios entre empresas homogéneas*

Se necesita integrar un servicio con una aplicación interna como un sistema de información empresarial, La aplicación interna puede no estar definida como un servicio y por tanto puede no ofrecer interfaces basadas en estándares. La interacción entre el servicio y la aplicación interna es directa a través de una conexión punto a punto. Por lo tanto no se requiere de algún tipo de ruteo inteligente, la aplicación interna puede estar en cualquiera de los lados de la conexión: bien solicitando la interacción o respondiendo a ella. Al tratarse de plataformas homogéneas no se necesita capacidad de transformación a nivel de transporte y mensajería.

Un ejemplo de este caso de uso sería la interacción entre dos empresas que comparten una plataforma tecnológica común (como en nuestro caso) y que desean exponer funcionalidad de sus aplicativos para comunicarse e invocar servicios entre ellas.

*U4: Integración punto a punto de los servicios dentro de la empresa*

De manera similar al caso de uso anterior se necesita que dos servicios existentes dentro de la empresa se comuniquen e interactúen para dar soporte a un proceso de negocio, pero en este caso los servicios exponen interfaces bien definidas y están basados en estándares por lo que la comunicación es directa, punto a punto y transparente para el resto del proceso de negocio.

Un ejemplo de este caso de uso sería la interacción entre dos aplicaciones empresariales dentro de la misma organización. Un servicio de registro de información de cliente, puede invocar a otro servicio web que se encargue de obtener los datos de garantías relacionadas al cliente en el sistema de garantías.

*U5: Permitir que los usuarios invoquen servicios de manera simple*

Un aplicativo nuevo o existente en la organización ya existe expuesta como servicio y se requiere que esté disponible a los usuarios finales. Estos deben estar en capacidad de invocar al servicio directamente o indirectamente a través de loica de presentación.

Un ejemplo común de este caso de uso puede ser el de un servicio web invocado por una aplicación cliente por medio de una interfaz ASP.NET. Por ejemplo un usuario selecciona revisar ratio de mora del cliente, y se invoca a un servicio web para obtener la información de mora.

*U6: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo estático*

Un cliente requiere desacoplar las conexiones punto a punto entre consumidores de servicios y proveedores de servicios. Una capa intermedia se agrega para proveer ruteo entre consumidores y proveedores. La capa intermedia provee ruteo estático, donde una solicitud por un servicio particular de un consumidor es siempre direccionado al mismo proveedor de servicio. La capa intermedia provee de seguridad de conexión, protocolos básicos, y transformación de mensajes entre consumidores y proveedores.

*U7: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo dinámico basado en estándares y protocolos*

Un cliente requiere desacoplar las conexiones punto a punto entre consumidores de servicios y proveedores de servicios. Una capa intermedia se agrega para proveer ruteo entre consumidores y proveedores. La capa intermedia provee ruteo estático, el ruteo dinámico determina el proveedor del servicio a usar basado en reglas de ruteo aplicada a la solicitud recibida del consumidor del servicio. La capa intermedia provee seguridad de conexión, protocolos básicos, y transformación de mensajes entre consumidores y proveedores.

*U8: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo dinámico avanzado y diversos protocolos*

Se requiere extender el acoplamiento débil entre consumidores de servicios y proveedores de servicios para incluir la conversión del formato de mensajes y/o protocolos usados por el consumidor del servicio, a un formato y protocolo manejable por el proveedor del servicio. Esto requiere una capa intermedia entre el consumidor y el proveedor para realizar la conversión. La capa intermedia provee un avanzado grado de lógica de ruteo, permitiendo la invocación de múltiples proveedores de servicios a raíz de una sola solicitud del consumidor del servicio. En este caso la capa intermedia debe desagregar y agregar solicitudes y respuestas de los proveedores de servicios. La capa intermedia provee seguridad y transformación avanzada de entre un diverso conjunto de consumidores y proveedores.

*U9: Mejorar un el flujo existente de un proceso de negocio por procesos de negocio y políticas de Modelamiento y simulación*

En este caso de uso se tiene un proceso de negocio consistente y además una serie de tareas manuales involucradas. Y se requiere mejorar la eficiencia de este proceso de negocio. Para poder lograr este objetivo la empresa documenta el proceso de negocio y luego ejecuta una serie de simulaciones a través del proceso para identificar posibles cuellos de botella, ineficiencias y costos asociados. Esta información es usada para crear un nuevo y más eficiente proceso de negocio.

*U10: Implementar un nuevo flujo de proceso de negocio*

En este caso de uso se desea crear un nuevo proceso de negocio. Este está implementación puede estar basada en procesos completamente nuevos o en el refinamiento de procesos de negocio existentes que a sido previamente modelado y simulado.

La implementación del nuevo proceso de negocio es diseñada por el analista de negocio y es creada por el arquitecto empresarial y el equipo de desarrollo. La implementación del proceso de negocio puede incluir actividades automatizadas expuestas como servicios y actividades manuales. Estas pueden residir dentro o fuera de la organización.

*U11: Analizar el flujo de un proceso existente de negocio usando monitoreo*

En este caso de uso se tiene implementado un determinado proceso de negocio y se desea monitorear la ejecución de las instancias del proceso como de este proceso de negocio. La información capturada puede utilizarse como

indicadores claves de performance (KPI), análisis de dimensiones y eventos para analizar la data de esta instancia del proceso de negocio.

**U12: Habilitar el acceso “single-sign-on” para diferentes servicios**

Se requiere habilitar el acceso a diferentes servicios con la capacidad de un logueo único, es decir el usuario con una sola autenticación puede invocar a múltiples servicios. Una vez autenticado el usuario será capaz de acceder a otros servicios sin necesidad de autenticarse nuevamente dentro de la misma sesión.

En algunos casos, los requerimientos de seguridad serán mayores y el contexto de seguridad necesitara ser extendido incluyendo a otros sistemas habilitando mecanismo de no repudio.

***U13: Personalizar la información basado en el perfil del usuario***

En este caso de uso un consumidor requiere de información basada en el contenido personalizado. La personalización y agregación de contenido es obtenido invocando a múltiples servicios.

Por ejemplo un usuario se autentica en una aplicación de seguros basada en portales, usando su ID y password. El contenido de la página del portal esta personalizado, basado en el perfil del usuario. El agente observa una lista de sus clientes con reclamos pendientes en un portal y otro contenido personalizado en otras partes del portal. El contenido personalizado es recuperado invocando a los servicios correspondientes.

***U14: Habilitar a los usuarios para crear y administrar su contenido***

En este caso de uso se requiere brindar a los usuarios la posibilidad de colaborar fácilmente, compartir datos e información con otros usuarios de la red de manera interactiva. Además no se necesita saber de la ubicación física o la dirección exacta de los otros usuarios de la solución. Algunos ejemplos de este caso de uso son los sistemas de correo electrónico o un “chat room”.

*U15: Habilitar a los usuarios para acceder a los servicios a través de los dispositivos cliente*

En este caso de uso se requiere exponer servicios a un amplio rango de clientes incluyendo computadoras tradicionales, notebooks, PDAs y teléfonos móviles. La motivación de este caso de uso es extender los procesos de negocio a los usuarios móviles usando para ello una combinación de plataformas de software clientes, middlewares, y tecnologías inalámbricas.

*U16: Habilitar a los usuarios para realizar investigaciones / análisis de la información*

En este caso de uso se requiere que los usuarios puedan realizar consultas de información. Los datos solicitados residen en diferentes almacenes de datos de múltiples niveles. Los datos son recuperados invocando servicios. Un escenario común puede ser un sitio web que presenta información categorizada y que permite al usuario realizar búsquedas complejas. La información puede provenir de diferentes fuentes. Un datamart puede contener datos frecuentemente usados, mientras que en el warehouse residen los detalles.

*U17: Poblamiento de la información*

En este caso de uso se necesita distribuir información a través de diferentes bases de datos, para hacerla accesible a un amplio rango de usuarios. Además los cambios realizados en el sistema principal deben ser transmitidos a diversos sistemas relacionados, donde serán repartidos a diferentes ubicaciones geográficas.

Escenario de Arquitectura SOA	Creación de Servicios	Conectividad de Servicios	Servicios de Interacción y Colaboración	BPM – Administración de procesos de negocio	Información como servicio
Casos de Uso					
U1: Creación o reutilización de lógica existente de aplicaciones como un servicio dentro de la empresa	x				
U2: Creación o reutilización de lógica existente de aplicaciones como un servicio fuera de la empresa	x				
U3: Integración punto a punto de aplicaciones mediante el uso de servicios entre empresas con plataformas homogéneas	x				
U4: Integración punto a punto de los servicios dentro de la empresa	x				
U5: Permitir que los usuarios invoquen servicios de manera simple	x				
U6: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo estático		x			
U7: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo dinámico basado en estándares y protocolos		x			
U8: Habilitar el acoplamiento débil de los servicios de los consumidores y proveedores usando ruteo dinámico avanzado y diversos protocolos		x			
U9: Mejorar el flujo existente de un proceso de negocio por procesos de negocio y políticas de Modelamiento y simulación				x	
U10: Implementar un nuevo flujo de proceso de negocio				x	

U11: Analizar el flujo de un proceso existente de negocio usando monitoreo				x	
U12: Habilitar el acceso "single-sign-on" para diferentes servicios			x		
U13: Personalizar la información basado en el perfil del usuario			x		
U14: Habilitar a los usuarios para crear y administrar su contenido			x		
U15: Habilitar a los usuarios para acceder a los servicios a través de los dispositivos cliente		x	x		
U16: Habilitar a los usuarios para realizar investigaciones / análisis de la información					x
U17: Poblamiento de la información					x

*Cuadro 15: Casos de Uso genéricos versus escenarios de arquitectura SOA*

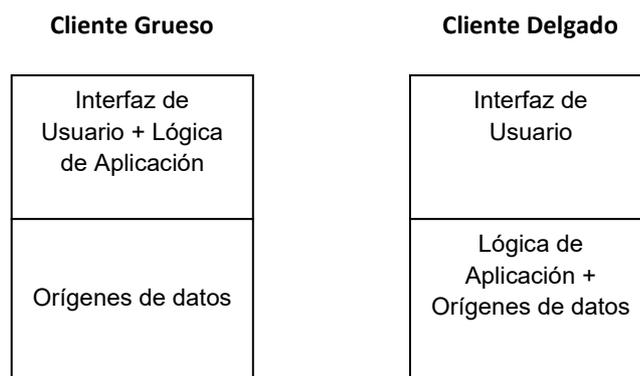
*Por tanto, en vista de la problemática presentada proponemos una Arquitectura Tecnológica empresarial Orientada a Servicios que se ubica en el tipo de arquitectura SOA de Nivel 1: Escenario de Creación de Servicios, esto se sustenta en lo siguiente:*

- Necesidad de publicación de funcionalidad existente en los aplicativos de la organización como servicios simples de utilizar tanto por usuarios finales como por aplicaciones desde fuera de la frontera de la organización
- Existencia de homogeneidad entre plataformas de las instituciones financieras destino, debido a que presentan en sistemas core de negocio similares.
- Necesidad de integración entre aplicativos desde dentro de la Caja Metropolitana para dar soporte a procesos de negocio que involucran a varias áreas funcionales.

- Necesidad de mayor autonomía tecnológica en el ámbito de la interoperabilidad de aplicaciones hacia fuera de la organización, es decir minimizar la dependencia con proveedores de tecnología middleware.

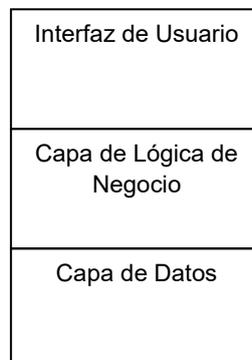
#### 4.2.3. Vista Lógica de la Arquitectura propuesta

En los inicios del desarrollo de los sistemas empresariales, no existía una clara separación de los datos, las lógicas de aplicación y las interfaces de usuario, básicamente los sistemas eran monolíticos y todo residía en el mismo espacio físico y lógico. A medida que se fueron desarrollando los nuevos sistemas se empezó a hablar de los llamados Clientes delgados (Thin Client) o Clientes Gruesos (Fat Client) los cuales eran representaciones figurativas de la distribución de las capas en los sistemas de información. Se les llamaba Clientes Delgados cuando en las máquinas de los clientes solo reside la interfaz de la aplicación y en el lado del servidor se encontraban tanto los datos como la lógica de la aplicación, es decir las reglas de negocio que frecuentemente se encontraban en store procedures o procedimientos almacenados creados en el motor de base de datos. Esto se ilustra en la figura 30.



*Figura 30: Variantes de distribución en la Arquitectura de 2 capas*

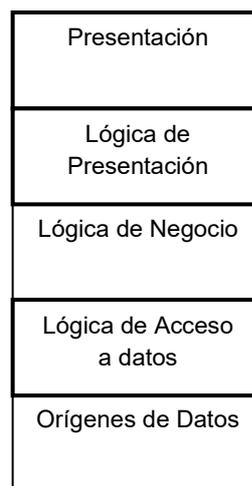
Con el paso de los años y el asentamiento de la arquitectura Cliente / Servidor el esquema fue cambiando y se hizo un estándar la separación en 3 capas, es decir la Lógica de aplicación se encontraba en el medio de las de datos y la interfaz, uno de los motivos que impulso este nuevo modo de ver la arquitectura fueron los problemas de escalabilidad que presentaban las aplicaciones, debido al vertiginoso crecimiento de las aplicaciones empresariales. Tanto en el caso de los clientes delgados como en el de los clientes gruesos, se suscitaba que a medida que el número de usuarios se incrementaba, se perdía performance con mayor rapidez y la administración de los cambios se complicaba de manera importante. Con las arquitecturas de tres capas nació a la par el concepto del Componente de negocio, los cuales fueron implementándose en diversas tecnologías según los proveedores de software (ejm. COM)



*Figura 31: Arquitectura tradicional de 3 capas*

Este mejoraba dramáticamente la escalabilidad de las aplicaciones, separando la lógica de la presentación y de los datos, tal como se muestra en la figura 31. Pero este esquema si bien ampliamente utilizado, resulta insuficiente para mostrar la arquitectura lógica en las aplicaciones empresariales contemporáneas. Por tanto podemos transformar este esquema, evolucionándolo hacia uno más preciso y consistente con la realidad actual.

Este nuevo esquema está compuesto por cinco capas lógicas.



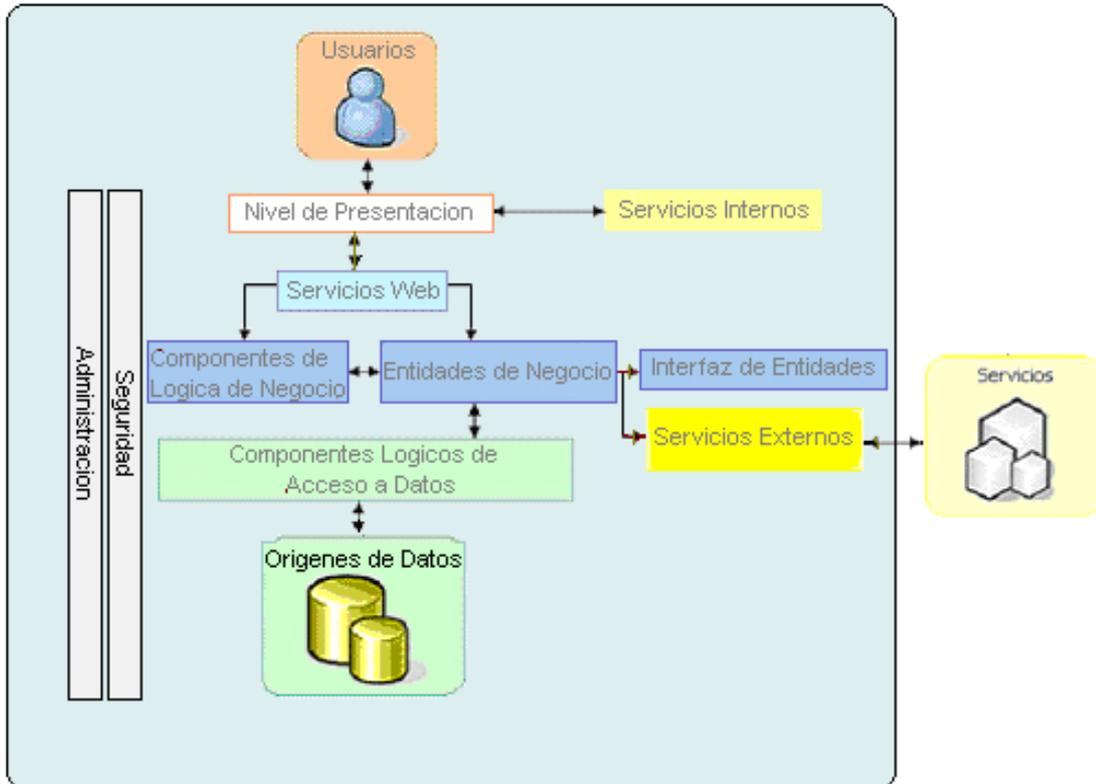
*Figura 32: Arquitectura de 5 capas*

Este modelo agrega capas intermedias que son la de Lógica de Acceso a datos y la Lógica de presentación, esto con el objetivo de obtener una mayor independencia entre capas así como permite la evolución y/o sustitución autónoma, minimizando considerablemente el impacto en otras partes del sistema. En la figura 32 podemos apreciar la vista lógica de esta arquitectura. Así tenemos que al separar las capas podemos implementar cada una con

distintos tipos de tecnología, cuidando que estas sean compatibles para la información. Ejemplos de los elementos de cada capa son: Bases de datos relacionales o archivos planos en la capa de Orígenes de datos, Clases o componentes COM en Componentes de Lógica de acceso a datos, Componentes COM, COM+, Enterprise Java Beans etc. en la capa de componentes de negocio, XML, Paginas Asp, php, aspx , Windows forms etc. en las capas de lógica de Presentación y presentación.

*Por tanto, en vista de la problemática presentada proponemos una Arquitectura Tecnológica empresarial Orientada a Servicios que se ubica en el tipo de arquitectura SOA de Nivel 1: Escenario de Creación de Servicios, y de tipo multi – capa:*

A continuación presentamos la Vista Lógica de la Arquitectura basada en servicios que deseamos aplicar para nuestro problema así como describimos cada una de las capas que se pueden apreciar en la figura 33



*Figura 33: Vista Lógica de la Arquitectura Propuesta*

#### 4.2.4. Capas de la Arquitectura

##### **Capa de Presentación**

La implementación de interfaces de usuario se puede llevar a cabo de varias formas. Por ejemplo, la aplicación comercial requiere una interfaz basada en el Web y otra basada en Windows. Otros tipos de interfaces de usuario pueden involucrar programas basados en documentos y aplicaciones de cliente móviles, entre otros. Los componentes de la interfaz de usuario administran la interacción con el usuario. Muestran los datos al usuario, obtienen los datos del mismo e interpretan los eventos generados por el

usuario para actuar en los datos empresariales, cambiar el estado de la interfaz o facilitar la tarea al usuario.

En un gran número de casos, la interacción del usuario con el sistema se realiza de acuerdo a un proceso predecible. Por ejemplo, cuando el usuario realiza una Consulta en línea del financiamiento, la interacción sigue un proceso predecible de recolección de datos por parte del usuario, por el cual éste en primer lugar proporciona datos de entrada tal como su Número de DNI en caso de personas Naturales o el RUC en caso de personas jurídicas.

## **Capas de Negocio**

### *Servicios Web*

La necesidad de comunicación de aplicación a aplicación y la interoperabilidad sobre la red creció cuando las compañías se dieron cuenta de la simplicidad de crear aplicaciones Web utilizando los Servicios Web, montándose sobre la infraestructura existente de Internet, de una manera efectiva, rápida y con un menor costo.

Un Servicio Web es una pieza de software auto disponible sobre Internet y utiliza un sistema estandarizado de mensajería XML, el Extensible Markup Language (XML) es el formato básico para representar datos en la plataforma de los Servicios Web. El XML fue elegido porque no es específico para una plataforma ni para un proveedor, además de ser simple de crear y analizar.

En esta capa la aplicación solicitante envía su petición a la aplicación proveedora del Servicio Web usando el protocolo SOAP, el cual está basado

en XML. La aplicación proveedora recibe la petición, procesa su lógica de negocio, y envía la respuesta usando el mismo protocolo SOAP.

La Capa de los Servicios Web es un conjunto de protocolos usados para definir, descubrir, e implementar dichos, tal como se describe en la Figura 34.

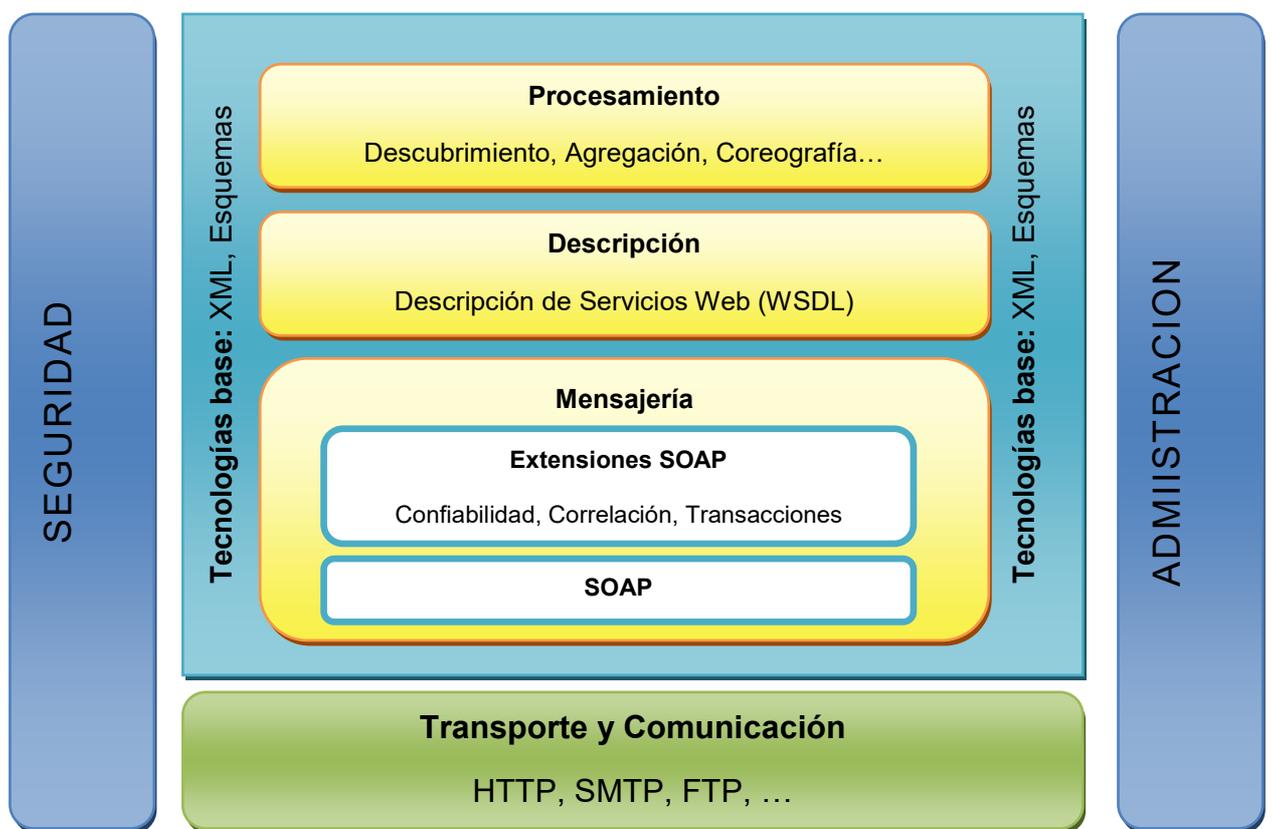


Figura 34: Componentes de un Servicio Web

El núcleo del Servicio de cuatro capas:

- **Capa de Servicio de Transporte:** Esto incluye el HTTP, el SMTP, el FTP.

Esta capa es la responsable de transportar mensajes entre las aplicaciones.

- **Capa de Mensajería XML:** Actualmente, esto incluye el XML-RPC (utiliza mensajes XML para realizar Llamadas a Procedimientos Remotos) y el SOAP - Protocolo de Acceso a Objetos Simples. Esta capa es responsable de codificar los mensajes en formato XML común para que dichos mensajes puedan ser comprendidos en cualquier extremo de la comunicación.
- **Capa de Descripción del Servicio:** La descripción del servicio se maneja vía WSDL - Web Service Description Language, Lenguaje de Descripción de los Web Services. Esta capa es responsable de describir la interfaz pública a un Servicio Web específico.
- **Capa de Recuperación del Servicio:** La recuperación del servicio se maneja vía UDDI - Universal Discovery Description and Integration, Descripción e Integración de Recuperación Universal. Esta capa es responsable de centralizar los servicios en un registro común, y de proporcionar una sencilla funcionalidad para publicación y localización de dichos servicios.

### *Componentes de Lógica de Negocio*

La parte más importante de la aplicación es la funcionalidad que proporciona. Una aplicación realiza un proceso empresarial que consta de una o varias tareas. En los casos más simples, cada tarea se puede encapsular en un método de un componente y llamar de forma sincrónica o asincrónica. Para los procesos empresariales más complejos que requieren varios pasos y transacciones de ejecución larga, la aplicación necesita disponer de un modo de organizar las tareas empresariales y almacenar el estado hasta que el proceso se haya completado. La parte principal de la lógica empresarial se

suele denominar lógica de dominio. Los componentes empresariales también pueden realizar solicitudes de servicios externos, en cuyo caso tal vez sea preciso implementar agentes de servicios para administrar la conversación requerida para la tarea empresarial específica realizada por cada uno de los servicios que necesita utilizar.

Los componentes empresariales pueden ser la raíz de las transacciones sencillas. Éstos implementan las reglas de negocio y aceptan y devuelven estructuras de datos simples o complejos. Los componentes empresariales deben exponer funcionalidad de modo que sea independiente de los almacenes de datos y los servicios necesarios para realizar la tarea, y se deben componer de forma coherente desde el punto de vista del significado y transaccional.

Los componentes de lógica de negocio, irán evolucionando y aumentando, proporcionando lógica y operaciones de mayor nivel que encapsulan la lógica preexistente. En un gran número de casos, necesitará componer funcionalidad empresarial preexistente con el fin de realizar la lógica empresarial requerida. Si el proceso de negocio invocará a otros procesos empresariales en el contexto de una transacción, todos los procesos invocados deben garantizar que sus operaciones participan en la transacción existente de modo que las operaciones se deshagan en caso de que la lógica de negocio que realiza las llamadas se interrumpa. Una técnica muy segura es volver a intentar una operación si ésta da error, sin miedo a que los datos pierdan su coherencia. Por ejemplo podemos considerar un límite de una transacción determinada como un límite de reintento. Las transacciones de los servidores

que ejecutan Windows se pueden administrar utilizando el Controlador de transacciones distribuidas (DTC), utilizado por el Enterprise Services (COM+).

Si no se pueden implementar transacciones, necesitará ofrecer métodos y procesos de compensación (extorno por ejemplo). Aunque debe tenerse en cuenta que una acción de compensación no devuelve necesariamente todos los datos de la aplicación a su estado anterior, sino que restaura los datos empresariales a un estado coherente. Por ejemplo, un proveedor puede exponer una interfaz de compra B2B (entre empresas) a sus socios. Una acción de compensación para cancelar un pedido en proceso puede conllevar la imposición de una cantidad por la cancelación de dicho pedido.

### *Entidad de Negocio*

Los componentes de entidad contienen datos de instantánea. Se trata casi de una caché de información local, por lo que la coherencia de los datos sólo se puede garantizar si se leen en el contexto de una transacción activa. Se recomienda no asignar una entidad empresarial a cada una de las tablas de la base de datos; normalmente una entidad empresarial dispondrá de una o más tablas de la base de datos.

Debido a que almacena los valores de los datos y los expone a través de sus propiedades, el componente proporciona acceso programático con estado a los datos empresariales y la funcionalidad relacionada. Se evitara el diseño de los componentes de entidad empresarial de modo que se obtenga acceso al almacén de datos cada vez que una propiedad cambia. Los componentes de entidad empresarial no deben tener acceso directo a la base de datos, pero

deben utilizar componentes lógicos de acceso a datos para realizar las tareas relacionadas con datos a medida que se invocan sus métodos. Las entidades empresariales no deben inicializar ningún tipo de transacciones ni utilizar API de acceso a datos; son meramente una representación de datos, potencialmente con comportamiento. Debido a que las entidades se pueden invocar desde componentes de negocio e interfaces de usuario, deberían permitir el flujo de transacciones de forma transparente y no deberían votar en transacciones salientes.

Es una buena idea diseñar componentes de entidad empresarial serializables, permitiéndole almacenar el estado actual. El diseño orientado a objetos es habitual en entornos con estado, como el diseño de interfaz de usuario, mientras que la funcionalidad y las transacciones empresariales se pueden expresar normalmente de forma más clara en términos de intercambios de documentos. La mayoría de las aplicaciones requieren el paso de datos entre distintos componentes.

Las entidades de negocio que se utilizan de forma interna en la aplicación suelen ser estructuras de datos, aunque también se pueden implementar utilizando clases orientadas a objetos personalizadas que representan entidades del mundo real necesarias para la aplicación, como productos o pedidos.

### *Interfaz de Entidades*

Exponen las siguientes características:

- Descriptores de acceso a propiedades (funciones Get y Set) para los atributos de la entidad.
- Descriptores de acceso a colecciones para sub-colecciones de datos relacionados. (Las colecciones no dan a lugar necesariamente a colecciones de entidades empresariales, por lo que se puede diseñar la entidad de servicio para exponer directamente conjuntos de datos o tablas de datos no relacionados con la transversal de modelo de objetos).
- Las funciones y propiedades de control que se utilizan normalmente en la administración de entidades, por ejemplo, Load, Save, IsDirty y Validate.
- Métodos de acceso a los metadatos de la entidad, que puede resultar útil para aumentar la facilidad del mantenimiento de la interfaz de usuario.
- Eventos para señalar los cambios en los datos involucrados.
- Métodos para realizar tareas empresariales u obtener datos para consultas complejas.

## **Capa de Acceso a Datos**

### *Componentes Lógicos de Acceso a Datos*

Independientemente del almacén de datos utilizado, la aplicación o el servicio utilizarán componentes lógicos de acceso a datos para obtener acceso

a los datos. Estos componentes abstraen la semántica del almacén de datos y la tecnología de acceso a datos y proporcionan una interfaz simple de programación para la recuperación y realización de operaciones con datos.

Los componentes lógicos de acceso a datos suelen implementar un patrón de diseño sin estado que separa el procesamiento empresarial de la lógica de acceso a datos. Cada uno de estos componentes suele proporcionar métodos para realizar operaciones Create, Read, Update y Delete (CRUD) relacionadas con una entidad empresarial determinada de la aplicación (por ejemplo, Order). Los procesos empresariales pueden utilizar estos métodos. La interfaz de usuario pueden utilizar las consultas específicas para procesar los datos de referencia.

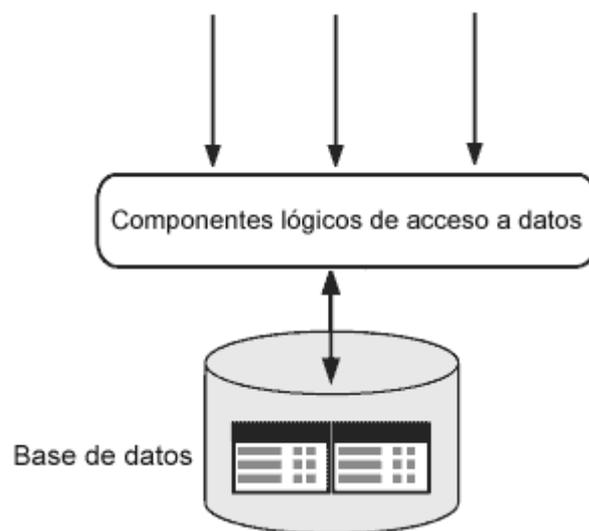
Cuando la aplicación contiene varios componentes lógicos de acceso a datos, puede resultar útil utilizar un componente de ayuda de acceso a datos genéricos para administrar las conexiones de las bases de datos, ejecutar comandos y almacenar parámetros en caché, entre otros. Los componentes lógicos de acceso a datos proporcionan la lógica necesaria para obtener acceso a datos empresariales específicos, mientras que el componente de ayuda para el acceso a datos centraliza el desarrollo de API de acceso a datos y la configuración de la conexión a éstos, permitiendo de esta forma la reducción de código duplicado. Un componente de ayuda de acceso a datos bien diseñado no debe repercutir negativamente en el rendimiento y proporciona una ubicación central para el ajuste y optimización del acceso a datos.

Los componentes lógicos de acceso a datos proporcionan acceso simple a funcionalidad de bases de datos (consultas y operaciones de datos), devolviendo estructuras de datos simples y complejas. Ocultan las idiosincrasias de la invocación y el formato del almacén de datos de los componentes empresariales y las interfaces de usuario que las consumen. La implementación de una lógica propia de acceso a datos en los componentes lógicos de acceso a datos permite encapsular toda la lógica de acceso a datos de la aplicación completa en una única ubicación central, lo que facilita el mantenimiento y la extensibilidad de la aplicación.

Una vez que se recupera la información de la Base de Datos, los datos se transfieren entre las capas y entre los distintos niveles de la aplicación y, finalmente, uno de los componentes los utilizará. Tal vez desee utilizar formatos de datos diferentes para recuperar, pasar y utilizar datos; por ejemplo, podemos solicitar una Consulta en línea del financiamiento y detalles de pago de los Aceptantes para ello procederemos a llenar las propiedades de un objeto de entidad personalizado. No obstante, se debería intentar mantener coherencia en cuanto al tipo de formato utilizado, ya que mejorará probablemente el rendimiento y la facilidad de mantenimiento de la aplicación para presentar sólo un conjunto limitado de formatos

Por ejemplo, la aplicación empresarial necesita recuperar los datos de los productos de una base de datos para mostrar al usuario los detalles de los mismos, así realizar el Ingreso de planillas y documentos en la base de datos cuando un usuario desea que dichos se usen en el proceso de Factoring.

Por tanto, es razonable abstraer la lógica necesaria para obtener acceso a los datos en una capa independiente de componentes lógicos de acceso a datos (Figura 35), ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.



*Figura 35: Componentes lógicos de acceso de datos y Origen de Datos*

### *Orígenes de Datos*

Es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y `archivos. Un **campo** es una pieza única de información; un **registro** es un sistema completo de campos; y un **archivo** es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de

registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

Sus características pueden ser ventajosas o desventajosas: pueden ayudarnos para almacenar, organizar, recuperar, comunicar y manejar información en formas que serían imposibles sin los computadores, pero también nos afecta de alguna manera ya que existen enormes cantidades de información en bases de datos de las que no se tiene control del acceso

Las bases de Datos tienen muchos usos: nos facilitan el almacenamiento de grandes cantidades de información; permiten la recuperación rápida y flexible de información, con ellas se puede organizar y reorganizar la información, así como imprimirla o distribuirla en formas diversas.

### **Capas de Servicios**

Cuando un componente empresarial requiere el uso de la funcionalidad proporcionada por un servicio externo, tal vez sea necesario hacer uso de código para comunicarnos con dicho servicio. Por ejemplo, los componentes de negocio de la aplicación comercial descrita anteriormente podrían utilizar un agente de servicios para administrar la comunicación con el servicio de autorización de tarjetas de crédito y utilizar un segundo agente de servicios para controlar las conversaciones con el servicio de mensajería. Los agentes de servicios permiten aislar las idiosincrasias de las llamadas a varios servicios desde la aplicación y pueden proporcionar servicios adicionales, como la

asignación básica del formato de los datos que expone el servicio al formato que requiere la aplicación.

Tenemos 2 tipos de Servicios:

### *Servicios Internos*

Está formado por el agente de impresión y agentes de transformación de datos hacia archivos PDF y archivos Excel.

### *Servicios Externos*

Son componentes de comunicación con entidades externas para obtención de datos. En algunos casos puede ser en línea.

## **Capas de Seguridad**

Al momento de tratar la seguridad en Servicios Web XML, es necesario considerar los siguientes puntos:

¿Cuál es nuestro objetivo? Limitar el acceso a Servicios XML a usuarios autorizados, evitar que los mensajes puedan ser visualizados por personas no autorizadas, etc.

¿Cómo vamos a cumplir con el objetivo deseado? Red, capa de transporte, sistema operativo, servicio o aplicación.

¿Qué nivel de interoperabilidad deseamos y necesitamos en nuestra solución?

Local o global.

¿Cómo se puede aumentar la seguridad de los Servicios XML actuales?

Esto se consigue respondiendo a las preguntas anteriores y, a continuación, aplicando las mismas técnicas de seguridad que utilizaríamos para cualquier otra aplicación Web: Crear conexiones seguras o Autenticar y autorizar la interacción

Como se puede comprobar estas técnicas ofrecen varias opciones que se pueden combinar para obtener beneficios adicionales. Por ejemplo, un servidor de seguridad puede funcionar con Servicios Web XML para limitar el acceso a ciertas funcionalidades (métodos) basándose en la identidad del cliente y las directivas definidas para cada cliente. Para completar la implementación con éxito, la planeación debe considerar lo siguiente:

- Tener una idea detallada de los riesgos potenciales del entorno (como virus, intrusos y desastres naturales).
- Realizar un análisis activo de la relación entre las consecuencias y contramedidas de una intrusión y los riesgos.
- Crear una estrategia de implementación planeada con detenimiento para integrar las medidas de seguridad en el conjunto de una red empresarial, basándose en los dos puntos anteriores.

Uno de los métodos más sencillos para aumentar la seguridad de los Servicios XML consiste en asegurarse de que la conexión entre el cliente de Servicios Web XML y el servidor es segura. Esto se puede llevar a cabo mediante varias técnicas, dependiendo de la extensión de la red y el perfil de actividades de las interacciones. Tres de las técnicas más comunes y accesibles son: reglas basadas en un servidor de seguridad, SSL (Secure Sockets Layer) y redes privadas virtuales (VPN).

Si sabe con exactitud qué equipos van a tener acceso a los Servicios Web XML, puede utilizar reglas de servidor de seguridad para restringir el acceso a equipos con direcciones IP conocidas. Esta técnica resulta particularmente útil si desea restringir el acceso a equipos en una red privada virtual (por ejemplo, una red LAN/WAN corporativa) y no desea mantener el contenido de los mensajes en secreto (cifrados). Esto resulta de gran utilidad cuando se espera que determinados clientes tengan acceso a funcionalidades (métodos) diferentes de los mismos Servicios Web XML.

Se puede utilizar SSL para establecer conexiones seguras en redes que no son de confianza (como Internet). SSL cifra y descifra mensajes enviados entre el cliente y el servidor. Al cifrar los datos, se evita que los mensajes sean leídos por terceros durante la transmisión. SSL cifra el mensaje del cliente y, a continuación, lo envía al servidor. Una vez que el servidor lo recibe, SSL lo descifra y comprueba que procede del remitente correcto (proceso que se denomina autenticación). El servidor, o el servidor y el cliente, pueden tener certificados que se utilizan como parte del proceso de autenticación y que proporcionan capacidades de autenticación además del cifrado de la conexión.

Aunque es un método muy eficaz para crear comunicaciones seguras, SSL presenta un coste en rendimiento que debe tenerse en cuenta.

## **Capa de Administración**

Esta capa se ocupa de la ejecución constante y diaria de la aplicación y abarca aspectos como la administración de excepciones, la supervisión, la supervisión empresarial, los metadatos, la configuración y la ubicación del servicio

- **Administración de excepciones:** La administración de excepciones incluye la detección y generación de excepciones, el diseño de éstas, el flujo de información de las mismas y la publicación de información de las excepciones a diversos usuarios. Todas las aplicaciones deben implementar algún tipo de control de las excepciones para detectar errores en tiempo de ejecución. Las excepciones se deben detectar y resolver si es posible. Si no se puede resolver un estado de error, la aplicación deberá mostrar un mensaje descriptivo para el usuario y proporcionar algún medio para el registro o publicación de la información de la excepción para la depuración.
- **Supervisión:** Se necesita instrumentar la aplicación de forma que proporcione al personal de operaciones información sobre el mantenimiento de la aplicación, compatibilidad con los acuerdos de nivel

de servicios , así como la administración de la escalabilidad y capacidad.

Su aplicación se puede beneficiar de los siguientes tipos de supervisión:

- Supervisión del mantenimiento: ¿se ejecutan correctamente los componentes? ¿Se producen bloqueos transitorios, cuelgues, salidas de procesos, colas bloqueadas y demás?
- Compatibilidad con SLA: ¿se ejecutan los procesos empresariales dentro de los parámetros esperados? ¿Cumplen las expectativas esperadas los servicios que integra? ¿Cumple la aplicación o servicio las expectativas de respuesta y rendimiento del llamador?
- Administración de la escala: ¿está correctamente diseñado el equipo, batería de servidores o red en los que se están implementando los componentes para la tarea que deben realizar? ¿Se puede predecir el rendimiento de los recursos disponibles?
- Supervisión empresarial: ¿puede mejorar la eficacia de los procesos empresariales? ¿Se pueden tomar decisiones fundamentales con tiempo? ¿Cuáles son los cuellos de botella en un proceso empresarial eficaz?
- **Supervisión empresarial:** Este tipo de supervisión tiene como objetivo proporcionar una capacidad reactiva para aquellos que dirigen la empresa en relación al mantenimiento de la empresa, compatibilidad con SLA a nivel de empresa y administración de la capacidad organizativa. En lugar de indicar los errores de la red, este tipo de supervisión ofrece

una perspectiva con respecto a la estructura empresarial y a la eficacia de los procesos. Por ejemplo, puede determinar que procesos empresariales se detengan durante días cada vez que un determinado socio forme parte del envío y control.

- La supervisión empresarial es un componente de la inteligencia empresarial, pero no sustituye otras técnicas como, por ejemplo el análisis OLAP y extracción de datos, que derivan sus datos de los procesos ETL (extraer, transformar, cargar) de los almacenes del servicio o aplicación para informar de decisiones activas basadas en tendencias de datos pasados. El principal factor diferenciador es que la métrica empresarial es transitoria y puede incluso que no se refleje en los datos de las aplicaciones.
- **Metadatos:** Para que la aplicación sea más flexible en relación a los cambios en las condiciones de tiempo de ejecución, puede que se desee proporcionar información sobre él mismo. El diseño de la aplicación para que utilice metadatos en determinados lugares puede facilitar el mantenimiento y permitirá su adaptación al cambio sin costosos procesos de implementación o modificaciones en el desarrollo. Hay dos momentos principales en los que puede utilizar metadatos en la aplicación:
- Tiempo de diseño: por ejemplo, puede utilizar información sobre la base de datos para generar código, procedimientos almacenados, clases o

incluso componentes de la interfaz de usuario para patrones que se repitan con frecuencia.

- **Tiempo de ejecución:** la aplicación puede ser más fácil de mantener si aprovecha la ventana de los metadatos adecuados para los aspectos de cambio más frecuentes. Por ejemplo, puede que se decida adoptar encabezados para una lista de IU o cuadrícula de metadatos, para que sean modificables en la aplicación. La aplicación también puede beneficiarse de los metadatos cuando establezca relaciones entre los componentes o cuando se procesen patrones predecibles, como reglas de validación. Sin embargo, el uso de metadatos en tiempo de ejecución suele ser más costoso en términos de rendimiento, por lo que debería probar y crear un perfil del diseño de la aplicación al principio del ciclo de vida de la aplicación.
- **Configuración:** Las aplicaciones requieren datos de configuración para funcionar técnicamente. Los valores que modifican el comportamiento de las directivas (seguridad, administración operativa y comunicaciones) se consideran datos de configuración.
- Los datos de configuración se conservan en los archivos de configuración a nivel de usuario, equipo y aplicación. La configuración personalizada almacenada aquí se puede definir con cualquier esquema y se puede tener fácil acceso. Es muy importante tener en cuenta la confidencialidad de seguridad de la conexión; por ejemplo, no se debe almacenar cadenas de conexión SQL en texto no cifrado en los archivos

de configuración XML, especialmente si contienen credenciales SQL. Se debe limitar el acceso a la información de seguridad a los operadores adecuados y a fin de disponer de una mayor seguridad, debería considerar la firma digital de la información para asegurarse de que los datos de configuración no se han modificado.

- **Ubicación de servicios:** En las llamadas a servicios remotos, es necesario determinar dónde están situados los objetos y servicios externos que pueden procesar su solicitud y cómo tener acceso a ellos. Esto resulta especialmente importante a la hora de utilizar servicios Web alojados por otras organizaciones o terceros.

#### 4.2.5. Vista Física de la Arquitectura propuesta

Las capas lógicas de la arquitectura se convierten en una arquitectura de una, dos o tres capas según como se complementen físicamente. Las soluciones Mono – posicionadas (solo una capa física, por ejemplo, una aplicación que corre en Access trabajando sobre una base de datos local) no son frecuentes en el entorno empresarial donde las aplicaciones usualmente son complejas, multi-posicionadas, a veces distribuidas y donde los datos están usualmente protegidos y requieren un considerable esfuerzo de administración.

Consecuentemente, las arquitecturas basadas en dos o más capas físicas son las más frecuentes en el entorno empresarial. Enfocándonos en estas últimas debemos de decir que se articulan básicamente en dos

configuraciones distintas, cada una con su propia distribución lógica, componentes, tecnologías etc.

Estas configuraciones son la Arquitectura Web y la Arquitectura Cliente / Servidor.

Ahora representemos gráficamente estas arquitecturas, asumiendo algunas premisas: Cada rectángulo representa un nodo que es una computadora. Este estará asociado a un rectángulo con bordes redondeados, donde están escritas las capas lógicas si es que tuviera, así como las diferentes tecnologías asociadas que pueden usarse para su implementación. Las flechas delgadas representan las conexiones entre los nodos y también están escritos junto a sus protocolos o sustitutos que puedan ser usados en este tipo de comunicaciones.

### **Vista Física de la Arquitectura Web**

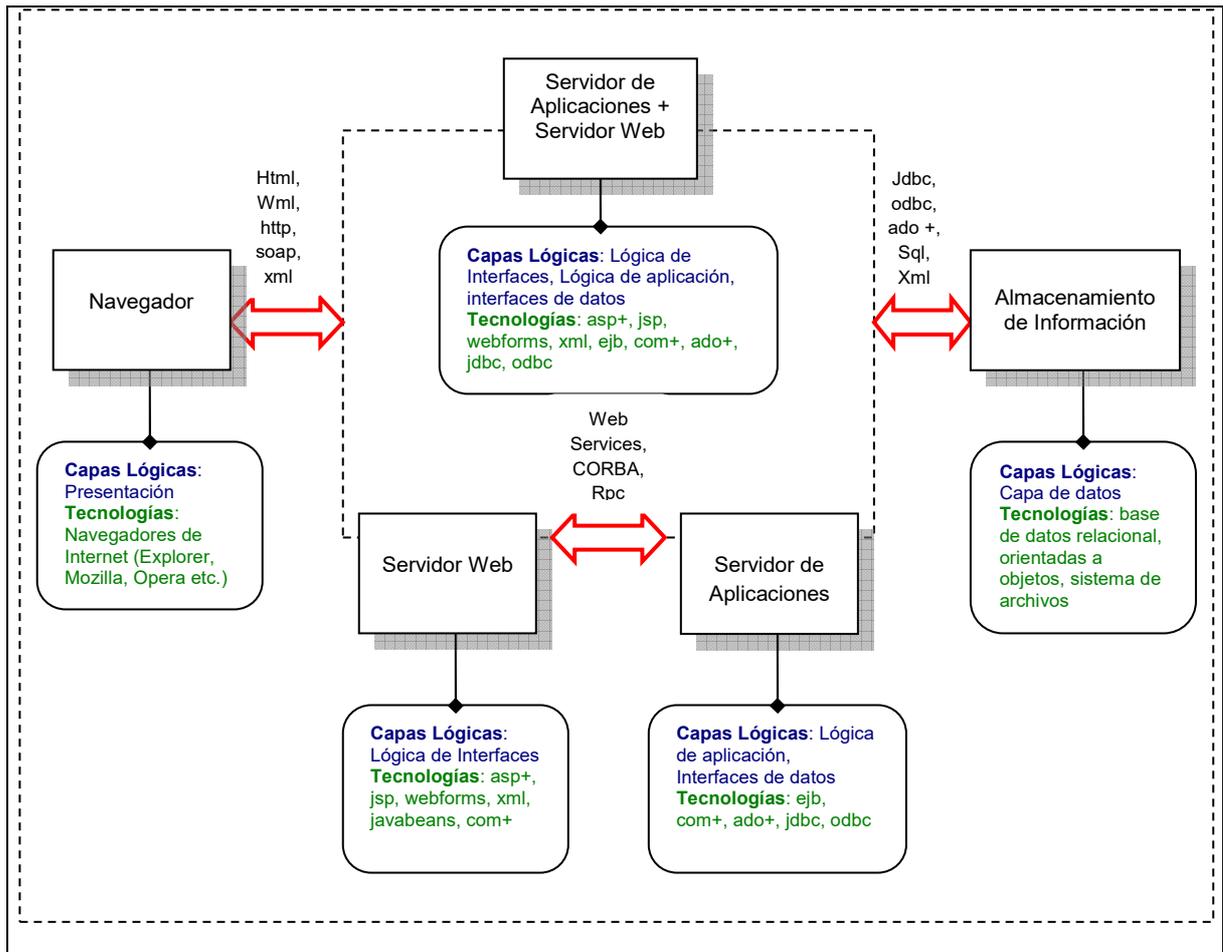


Figura 36: Vista Física de una Arquitectura tipo Web

En la figura 36 se muestran las dos diferentes variables de una arquitectura Web: una, formada por el navegador, el servidor Web, y la base de datos; y el otro, donde el servidor de aplicaciones es independiente del servidor Web.

Las arquitecturas Web representan uno de los mayores cambios en las aplicaciones empresariales durante la última década. Estas han traído una gran cantidad de beneficios como: el acceso universal por medio de los navegadores de Internet (IE y Mozilla), estandarización de protocolos de comunicación (http, Soap, etc.), estandarización de los lenguajes de

representación grafica de interfaces (html), reducción de costos asociados al mantenimiento del software en la PC.

### Vista Física Arquitectura Cliente / Servidor Tradicional.

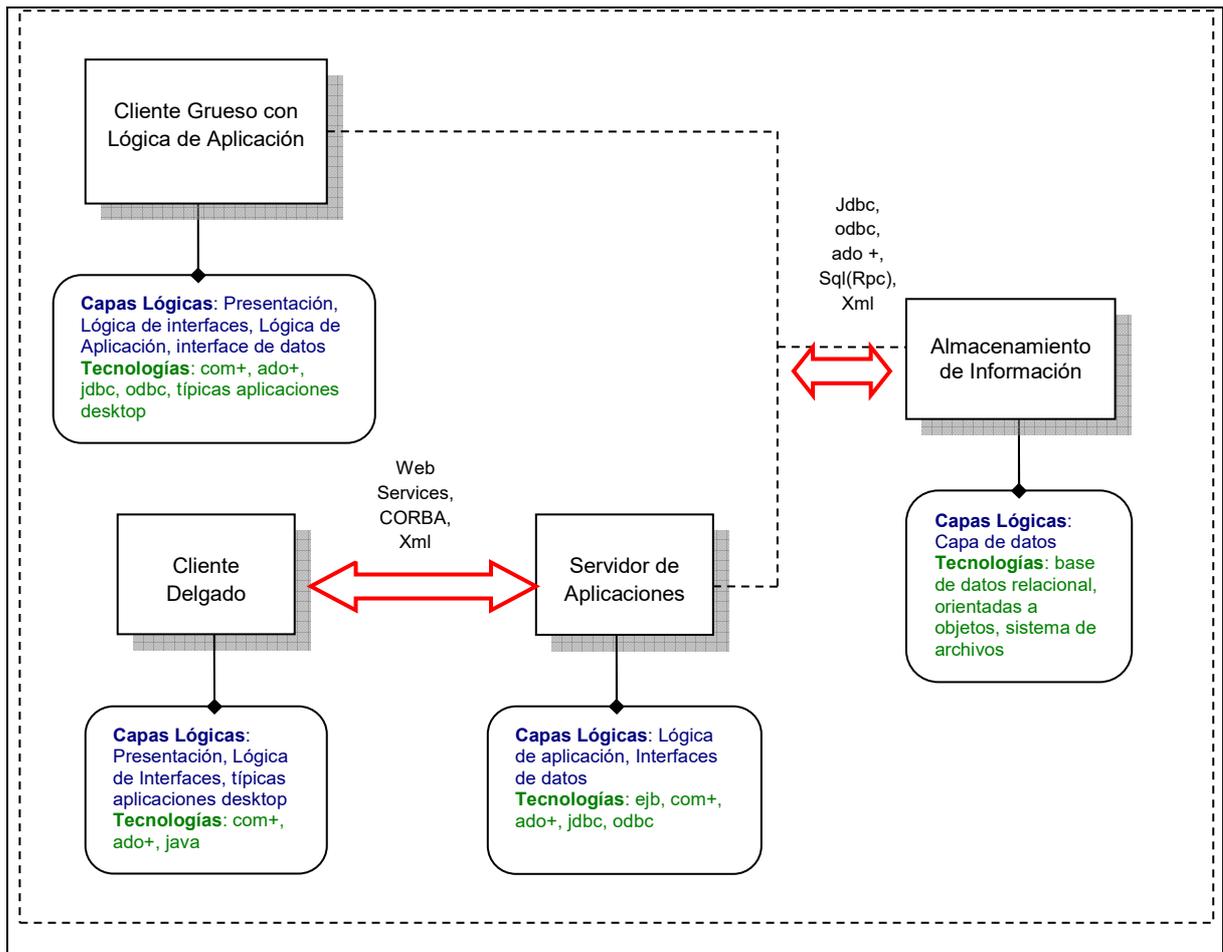


Figura 37: Vista Física de una Arquitectura típica Cliente / Servidor

En la figura 37 podemos observar la vista física de una arquitectura cliente/servidor típica para aplicaciones desktop. Clientes y servidores se supone que pertenecen a la misma web corporativa en este tipo de arquitectura. Desde este punto de vista esta arquitectura presenta algunas

ventajas. Por ejemplo proporciona la oportunidad de crear interfaces muy versátiles y ricas aprovechando la capacidad de procesamiento de las computadoras clientes y además ofrecen una mayor agilidad a las aplicaciones, lo que requiere un mayor ancho de banda para cada cliente.

Pero también este esquema presenta muchas desventajas, si lo comparamos con las arquitecturas Web. Hay un costo mucho mayor de mantenimiento de las aplicaciones ya que es más costoso construir y distribuir las actualizaciones en las estaciones de los clientes

*Por lo tanto bajo lo observado en estas vistas tanto físicas como lógicas decidimos que la configuración que mejor se adapta a la solución de nuestro problema de negocio en la organización definitivamente es la Arquitectura Web Multi-Capas basada en el nivel 1 SOA: Escenario de Creación de Servicios.*

#### **4.3. Metodología de Generación de Servicios**

Toda implementación de proyectos sobre todo los relacionados a las Tecnologías de información y en especial cuando se habla de tecnologías que hacen su aparición por primera vez en una organización, requieren de una cuidadosa metodología que determine cuales son los pasos a seguir para sus construcción. En este caso la Arquitectura SOA plantea una visión orientada a los Servicios y estos deben desarrollarse siguiendo una metodología que permita el adecuado desarrollo, mantenimiento y catalogamiento de los mismos para permitir un adecuado crecimiento ya que como se deduce de lo anterior la organización empezara a enfocar sus soluciones de negocio apoyadas en esta nueva arquitectura. Es así como planteamos los siguientes pasos para el

proceso de generación de servicios: Identificación, Planificación, Realización y Producción. Los cuales como se observa son iterativos, es decir la creación de servicios (primero de grano fino) ira derivando posteriormente en la agregación de los mismos y a cambios o mejoras que a medida que se vayan utilizando y re-utilizando serán la entrada para una nueva versión del servicio que soporte nuevas características

Antes de que un desarrollador escriba la primera línea de código es imprescindible identificar cuáles son los principales elementos motrices de la empresa de cara al proyecto SOA y las dependencias existentes entre el propio negocio y las tecnologías que lo soportan. Ignorar el contexto empresarial puede dar origen a un proyecto donde la infraestructura SOA se implante sin motivo o donde las inversiones realizadas no tengan un correlato adecuado con las necesidades y prioridades de la propia empresa.

Se aplican generalmente dos tipos de estrategia para implementar SOA: la denominada “descendente” (“top-down” en inglés) y la “ascendente” (o “botton-up”). Ambas tienen sus propios puntos débiles que pueden poner en riesgo el éxito del proyecto. Muchas organizaciones que han intentado poner en marcha una infraestructura SOA aplicando el enfoque top-down han descubierto después que cuando la infraestructura por fin se ha puesto en servicio, está desconectada de las necesidades reales del negocio. Y a la inversa, un enfoque ascendente puede también fracasar porque puede originar una implementación caótica de servicios creados sin tener en cuenta los objetivos de la organización.

Existe una metodología híbrida, denominada de “término medio” (“middle-out”) que es una síntesis equilibrada de estas dos anteriores. Los condicionantes principales de la empresa y la visión estratégica se emplean en primer lugar para establecer con claridad el rumbo y las prioridades del proyecto. Basándose en ellos, se inicia un proceso iterativo de múltiples pasos orientados a crear pequeños fragmentos de funcionalidades de alto nivel, y en cada iteración se entrega a la organización una nueva aplicación dinámica que se utiliza para generar retorno de la inversión.

Esta estrategia tiene la ventaja de una rápida puesta en valor y genera resultados de negocio en todos sus pasos incrementales e iterativos, facilitando un correcto alineamiento de los recursos de IT con las condiciones de negocio

### **Identificación de Servicios:**

Etapa en la que se inicia el ciclo iterativo, comienza por la evaluación del servicio, la forma en que se identifica (fino, grueso), hacer el mapeo inicial y hacer el registro inicial en el portafolio, esto se ilustra en la figura 38

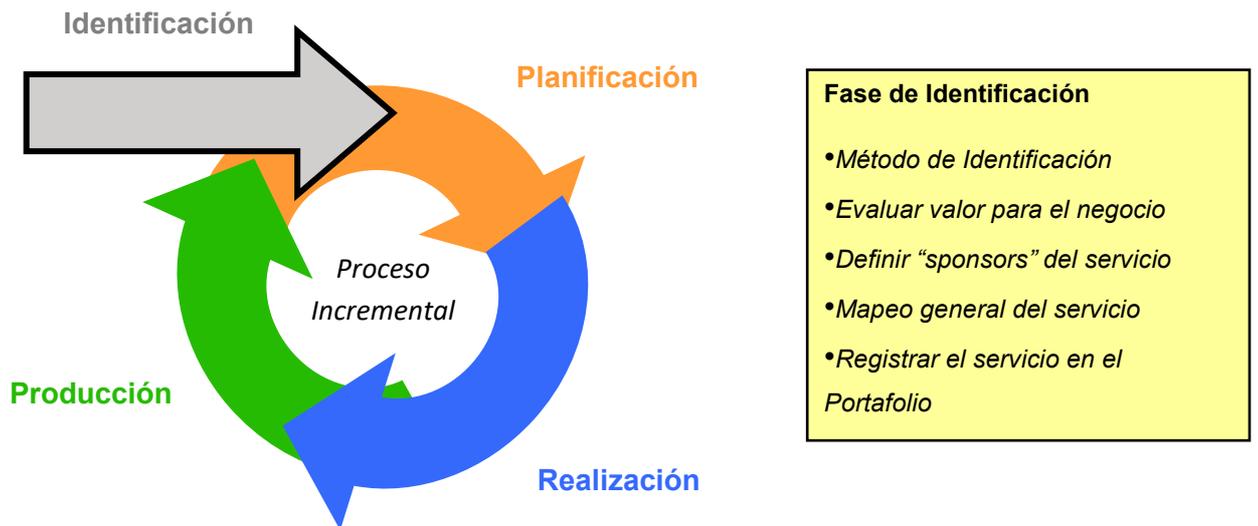


Figura 38: Fase Identificación de Servicios

*Método de identificación de servicios:*

- Hay servicios que pueden ser ricos en funcionalidad, pero para que su implementación es necesario construir primero los servicios de funcionalidad más granular (puntual) que los alimente. Esto nos da un parámetro de priorización en la construcción de servicios.
- Hay servicios que son descubiertos fácilmente por que su funcionalidad es obvia en los diferentes procesos y áreas de la organización. Por lo que deben ser los primeros en ser generados y de los cuales se espera una alta reutilización.
- Existirán servicios cuyo nivel de reutilización sea baja, pero de los cuales se requiere una alta flexibilidad para que se pueda adaptar rápidamente

a los cambios de negocio. En este sentido, estos servicios también requieren de una prioridad de atención.

- Hay servicios que para descubrirlos necesitaran mayor análisis y tiempo. Estos generalmente serán los que son específicos para ciertas áreas y tendrán una menor prioridad.
- Revisar los servicios disponibles: Antes de decidir si crear un nuevo servicio, hay que verificar si existe un servicio que cubre o que puede cubrir la funcionalidad que se requiere. El objetivo de este paso es fomentar la reutilización de los servicios existentes y evitar la construcción de servicios que ofrecen funciones similares. En caso no se encuentre un servicio que ofrezca la funcionalidad deseada, proceder con los siguientes pasos

### *Evaluar valor para el negocio*

- Definir el valor de negocio.
  - Definir la responsabilidad y beneficios del servicio.
  - Identificar los potenciales usuarios del servicio en términos de procesos, de servicios consumidores y de otras aplicaciones.
  - Realizar el análisis costo beneficio que justifique la creación del servicio y que sirva como criterio del por qué construirlo.
- Si la evaluación previa a sido positiva:

- Ponerlo en la lista de servicios a construir en el portafolio de servicios.
- Clasificarlo dentro del modelo de negocio y dentro de la clasificación de servicios.

#### *Definir el “sponsor” del servicio*

- Quién será el dueño del servicio (en términos de áreas del negocio)
- El área usuario que definirá la funcionalidad futura del servicio.
- Qué equipo construirá el servicio y en qué momento.
- Quienes serán los consumidores y cuándo lo irán adoptando.

#### *Mapeo general del servicio*

- En esta etapa se definirá en términos de niveles de rendimiento, el volumen transaccional que debe de soportar y los canales que lo consumirán.
- Generar un mapeo del servicio. La construcción del servicio se puede dividir en fases en las cuales se irá enriqueciendo funcional y/o técnicamente generando así el mapeo del servicio, el cual debe de estar registrado en el portafolio de servicios.

#### **Planificación de Servicios:**

Fase en la que se define los pasos previos a la realización, es decir cómo definir prioridad y alcance del servicio, responsables y plataforma de desarrollo. Esto se grafica en la figura 39

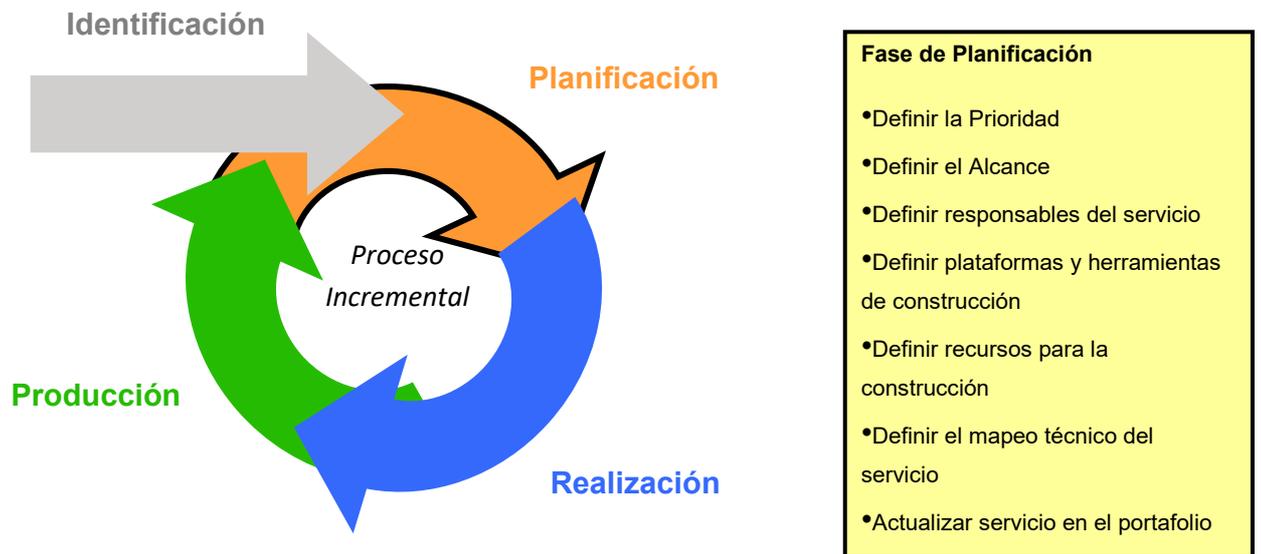


Figura 39: Fase Planeamiento de Servicios

#### *Definir la prioridad*

De acuerdo a todos los servicios que se encuentran en cartera, definir la prioridad de ejecución de los servicios de acuerdo a:

- La disponibilidad de las dependencias de otras piezas técnicas o funcionales para la construcción del servicio.
- Las necesidades del negocio en el corto y mediano plazo.
- Los recursos disponibles para la atención del requerimiento

#### *Definir el alcance*

Definir el alcance del servicio, detallando la implementación tal que quede clara la definición de cómo se construirá. En este caso se definirá si el servicio

- Se construye totalmente nuevo.
- Se reutilizará alguna funcionalidad existente, exponiéndola a través de una interfaz estándar.
- Se actualizará la versión de un servicio existente en base a su mapeo definido.

*Definir los responsables del servicio:*

Todos los servicios deben de tener los siguientes responsables:

- De negocio: Quien determina la funcionalidad que debe de cumplir y definir su mapeo de acuerdo a las nuevas estrategias. Es probable tener mayor dificultad en asignar un dueño de negocio de los servicios de los servicios de granularidad fina y altamente reutilizados por las diferentes áreas.
- Arquitectura: Quien determina cómo el servicio encaja y evoluciona dentro del contexto tecnológico de la Caja Metropolitana.
- Técnico: Quien determina cómo el servicio es construido internamente. En este caso, también es posible definir un mapeo de mejoras técnicas.

Los servicios de granularidad fina son más estables en el tiempo, por lo que es posible que no tengan tantas variantes como servicios compuestos o de granularidad más gruesa.

### *Definir la plataforma y herramientas de construcción*

Esta es una función que debe ser desarrollada por el Área de Sistemas. Los servicios que residan en alguna pieza tecnológica especial como el (Interconexión con Terceros, Message Broker, Biztalk Server) u otra que se defina, deberá de seguir los procedimientos y lineamientos asociados a cada una de estas piezas a lo largo de su desarrollo.

### *Definir los recursos para la construcción*

A partir de los pasos anteriores, definir recursos que se requerirán para la construcción de los servicios en términos de:

- Estimar tiempo.
- Personas.
- Hardware.
- Software.

Dependiendo del tipo de servicio, se definirá los lineamientos que se aplicarán teniendo en cuenta que

- Hay lineamientos que no son negociables y que deben de cumplir en la construcción del servicio, y
- Hay sugerencias y recomendaciones que pueden tomar en el diseño del servicio.

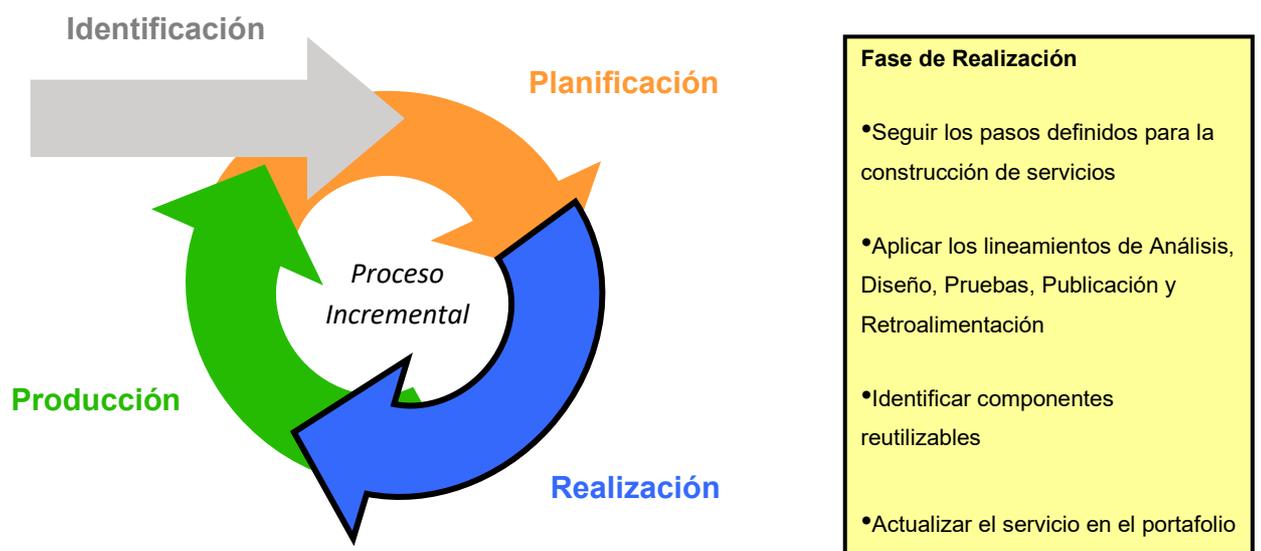
### *Definir un mapeo del servicio:*

- Definir un mapeo para la construcción del servicio y posibles versiones posteriores.
- Registrar la definición del Servicio en el Repositorio de Servicios
- Documentar el detalle de cómo será construido el servicio

*Actualizar servicio en el Portafolio:*

- Actualizar los cambios en el Portafolio de Servicios

## Realización



*Figura 40: Fase de Realización de Servicios*

*Seguir los pasos definidos para la construcción de servicios*

- Definir funcionalmente y a detalle cómo se construirá. Aplicar los pasos para la construcción de un servicio. Aplicar los lineamientos de servicios.

- Seguir los lineamientos para el proceso y realizar el seguimiento en caso de ser necesario.
- Seguir los pasos para la construcción de servicios, definidos en el Reporte de Especificaciones Funcionales

#### *Realizar las pruebas y el aseguramiento de la calidad del servicio*

- Crear casos de prueba para el servicio: De acuerdo a todos los posibles escenarios de uso del servicio, se debe de crear los casos de prueba para el servicio. Se recomienda crear casos de prueba considerando las características de negocio y plataforma de cada uno de los potenciales consumidores del servicio.
- Someter a pruebas de estrés al servicio: Para determinar los niveles de servicio que ofrece, se debe de someter al menos a las siguientes 3 pruebas de estrés.
  - Latencia
  - Concurrencia
  - Volumen de transacciones
- Someter a pruebas de estrés en el contexto de los procesos que lo utilizan.
- Realizar un plan de implementación y/o migración hacia el servicio.

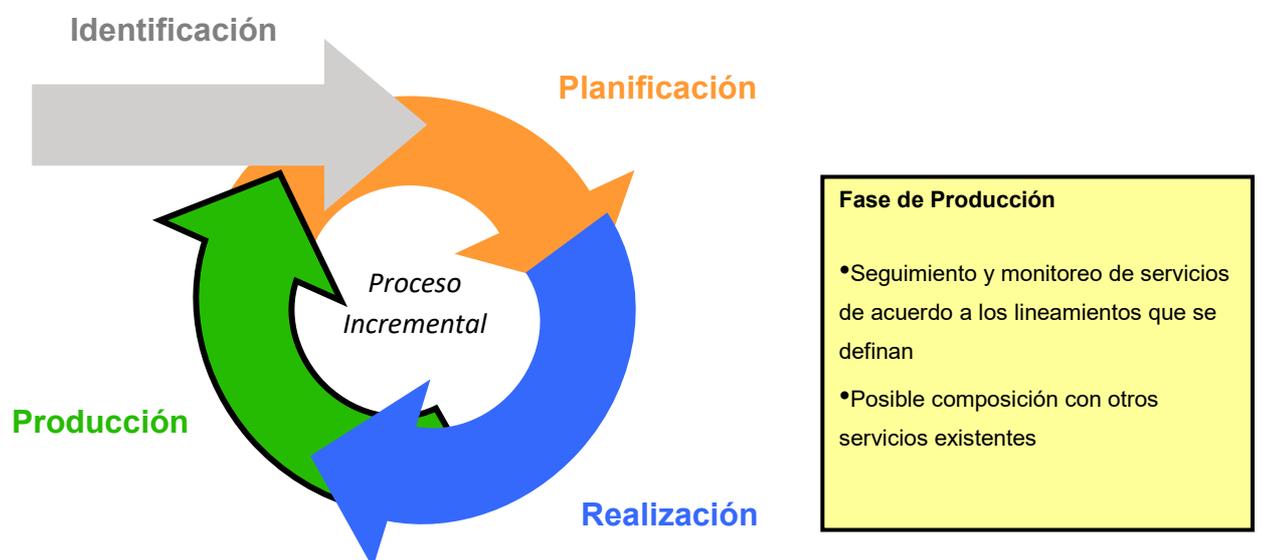
#### *Identificación de componentes reutilizables*

En el proceso de realización de los servicios pueden identificarse si estos pueden ser reutilizados y bajo que características, esto permite alimentar

el catalogo de servicios indicando que el servicio desarrollado puede servir para futuras soluciones de negocio y bajo que puede reutilizarse por otros servicios.

*Actualización del estado del servicio en el portafolio*

## Producción



*Figura 41: Puesta en producción de Servicios*

- Realizar seguimiento y monitoreo a servicios en Producción de acuerdo a los lineamientos que se definan.
- Cuando los servicios ya están creados se pueden combinar en servicios de mayor nivel de complejidad, aplicaciones o procesos de negocio

multifuncionales. Puesto que los servicios son entidades independientes entre sí y también con respecto a la infraestructura de IT en la cual se basan, pueden combinarse y reutilizarse con la máxima flexibilidad. Y según van evolucionando los procesos de negocio, las reglas y prácticas internas pueden ajustarse sin las restricciones impuestas por las limitaciones que afectan a las aplicaciones de base.

Después de crear una nueva aplicación o proceso de negocio, la funcionalidad resultante se pone a disposición (consumo) por parte de usuarios finales o de otros sistemas de IT. Al crear aplicaciones compuestas que consumen estos servicios y procesos, la organización dispone ahora de aplicaciones dinámicas que permiten mejorar la productividad y la visión interna del rendimiento de la empresa. Los usuarios pueden consumir los servicios compuestos utilizando distintos medios, como pueden ser portales Web, clientes avanzados, otras aplicaciones y dispositivos móviles.

#### **4.4. Lineamientos del proceso de Generación de Servicios**

A continuación planteamos los lineamientos que deben tenerse en cuenta en el proceso de generación de servicios: (cuadro 16)

Categoría	Lineamiento	Consideraciones
-----------	-------------	-----------------

<b>Orientación a Servicios</b>	<p>Reutilización: Puede ser utilizado por múltiples solicitantes y en diversos escenarios, sin necesidad de realizar cambios o ajustes.</p>	<ul style="list-style-type: none"> <li>• Generar la lista de solicitantes potenciales del servicio</li> <li>• Identificar las posibles variaciones de las funciones de negocios.</li> <li>• Finalmente deberá poder atender diferentes y nuevos procesos de negocios</li> </ul>
	<p>Manejo de versiones para que el servicio sea modificado, extendido o limitado sin mayor impacto en sus consumidores.</p>	<p>Considerar compatibilidad con sus consumidores, si es que cambia en sus interfaces o implementa nuevas funciones</p>
	<p>Desacoplamiento Interno: Cada unidad de programación debe pertenecer sólo a una pieza según la composición estándar de un servicio.</p>	<p>Las Capas del servicio son:</p> <ul style="list-style-type: none"> <li>• Lógica</li> <li>• Interfaces (solicitantes y proveedoras)</li> <li>• Administración</li> <li>• Seguridad</li> <li>• Operaciones</li> <li>• Datos (configuración y parametrización, log, info interna)</li> </ul>
	<p>Desacoplamiento externo: Debe estar desacoplado de otros elementos de la arquitectura organización</p>	<p>Debe consumir funciones de otros componentes a través interfaces</p> <p>No debe leer directamente archivos o bases de datos de otros aplicativos</p>

	<p>Debe Exponer sus funciones de acuerdo a los estándares definidos en la organización</p>	<p>Estándar actual: Web services (UDDI/WSDL/SOAP)</p>
	<p>Debe contar con documentación requerida para su uso y mantenimiento</p>	<p>Documentos:</p> <p>Manual de Uso: Como expone sus servicios (interfaces, métodos, estructura de datos, valores, etc) y como pueden ser utilizados</p> <p>Documentación técnica (como ha sido implementado técnicamente)</p>
	<p>Debe Encapsular la complejidad y expone interfaces simples y estándares</p>	<p>Los datos que los solicitantes le entregan para obtener su función:</p> <p>Deben ser los mínimos indispensables</p> <p>Deben ser siempre los mismos</p> <p>La complejidad de sus funciones no debe ser trasladada a sus solicitantes</p>
	<p>Si la funcionalidad del servicio ya existe en diferentes aplicaciones, el servicio deberá contener todas las variantes</p>	<p>Investigar si la función que brindará el servicio existe y está repetida, y sus variantes</p> <p>Incluir en la funcionalidad del servicio a construir todas las variantes de la funcionalidad</p>

	<p>Debe ser multimoneda, , multiproducto</p>	<p>Debe brindar la misma funcionalidad para todos los " multis "</p> <p>Debe contener campos que permitan identificar la moneda, institución, producto o segmento al que está atendiendo</p> <p>Puede contener restricciones por tipo, a través de parámetros</p>
	<p>El servicio se debe configurar a través de parámetros</p>	<p>Debe tener al menos parámetros para añadir o modificar moneda, institución, banca, segmento</p> <p>Analizar si el servicio requerirá otros parámetros</p>
	<p>Si es de "logica pura" las reglas de negocio que implementa deben ser únicas, genéricas, reutilizables y ningún otro elemento de la arquitectura las provee</p>	<p>Ningún otro elemento de la arquitectura contiene las reglas de negocio que brinda el servicio común</p> <p>Ej: .Simulador de cuotas</p>
	<p>Debe manejar integridad y transaccionalidad de funciones monetarias</p>	<p>El servicio debe asegurar que la función se complete al 100% o que se revierta la operación en caso de falla.</p> <p>Debe manejar integridad / transaccionalidad de forma automática o a través de compensación o extorno</p>

<b>Integración e Interfaces</b>	No duplicar interfaces proveedoras	<p>Contiene una única interfaz por funcionalidad</p> <p>No generar dos interfaces con la misma función para diferenciar:</p> <ul style="list-style-type: none"> <li>• Elementos solicitantes</li> <li>• Plataformas de elementos solicitantes</li> </ul>
	Las interfaces deben ser flexibles a las necesidades del negocio	Las interfaces deben contener la suficiente información para satisfacer las necesidades del negocio.
	Las interfaces proveedoras de extracción de información deben ser anchas	<ul style="list-style-type: none"> <li>• Deben contener todos los datos operacionales del servicio</li> <li>• De estas interfaces los aplicativos extraerán la información que requieran</li> <li>• No se debe generar interfaces específicas para un aplicativo solicitante</li> </ul>
	Debe reutilizar interfaces ya existentes (como solicitante)	<ul style="list-style-type: none"> <li>• Debe utilizar interfaces existentes para obtener información de otras aplicaciones</li> <li>• No debe leer directamente las bases de datos o archivos de otras aplicaciones</li> </ul>

	El servicio debe acceder a otros servicios o aplicaciones que están plataformas diferentes a través de los estándares de integración	En general a los sistemas legados o de externos de otras de manera transparente para el consumidor
	Los datos contenidos en las interfaces deben estar en formato XML	Se debe de respetar los elementos definidos para comunicación
<b>Seguridad</b>	Debe utilizar un servicio de autenticación, que identifica quien solicita información o sus funciones.	<ul style="list-style-type: none"> <li>• Utilizar elementos estándares de autenticación definidos por seguridad informática</li> <li>• El servicio debe conocer por qué medio y quien realiza la solicitud.</li> </ul>
	Debe Permitir el no repudio. Probar que un solicitante ha iniciado, participado o completado el uso de un servicio a través del uso de log (Log de Auditoría)	Se debe registrar el usuario fecha hora y operación realizada para facilitar la identificación de dolo
	Debe Asegurar la privacidad/confiabilidad/integridad de la información que fluye a través suyo. La información no podrá ser interceptada o visualizada	Utilizar si se requiere mecanismos de encriptación.

	El servicio debe operar con riesgo controlado	Recomendar a los responsables de los proyectos que se consulte al área de riesgos cuales son las condiciones para operar con riesgo controlado
<b>Información</b>	El servicio no debe guardar información de negocio (al menos, no persistentemente). A excepción de los parámetros	<ul style="list-style-type: none"> <li>• Almacena solo la información de negocios necesaria para su operación con límites temporales de persistencia (solo por un tiempo)</li> <li>• Cuenta con mecanismos de sincronización eficientes con las fuentes oficiales de información</li> </ul>
<b>Auditoría</b>	Debe manejar log de negocios estándar (en formato y estructura), para permitir su consolidación posterior	<ul style="list-style-type: none"> <li>• No podrá pasar a producción si no cuenta con el log en formato y estructura estándar</li> <li>• Al menos debe contener el usuario, tipo de operación, fecha y hora, cuentas, importe, moneda</li> <li>• Debe contener la estructura completa de la interfaz de entrada</li> </ul>
	Debe manejar un log técnico que permita rastrear errores o deshacer una operación "automáticamente"	<ul style="list-style-type: none"> <li>• Debe registrar datos que permitan efectuar el extorno de una función de actualización (monetaria).</li> </ul>

	<p>Debe contar con mecanismos automáticos para administración de logs</p>	<ul style="list-style-type: none"> <li>• Funciones estándares del software base para la registrar información en los logs</li> <li>• Mecanismos de depuración con tiempos parametrizables</li> <li>• Mecanismos de compresión de log para permitir su transporte</li> </ul>
<p><b>Administración</b></p>	<p>Debe ser configurable vía un módulo de administración que permita modificar parámetros para satisfacer las necesidades de uso de los diferentes consumidores del servicio.</p>	<ul style="list-style-type: none"> <li>• La configuración de parámetros se realiza exclusivamente a través de interfaces del módulo de administración</li> </ul>
<p><b>Infraestructura y Operaciones</b></p>	<p>Debe considerar mecanismos de soporte de alta disponibilidad y balanceo de carga</p>	<ul style="list-style-type: none"> <li>• Debe cumplir el Nivel de servicio definido con el negocio</li> <li>• Si el hardware y software base no soporta alta disponibilidad, se debe considerar la ejecución de más de una instancia del servicio, si éste servicio cae.</li> </ul>

	Manejo de errores	<ul style="list-style-type: none"> <li>• Registra eventos y alertas en caso de errores críticos, que faciliten el troubleshooting</li> <li>• Frente a un error debe contar con mecanismos de corrección (de preferencia automáticos) o manuales de operaciones</li> <li>• Prever los diferentes tipos de caídas que se puedan presentar</li> </ul>
	Durante el horario de mantenimiento, se debe contar con mecanismos para dejar de operar	<ul style="list-style-type: none"> <li>• Definir mecanismos para operar en offline si el servicio es 24 x 7</li> <li>• Definir horarios de mantenimiento para dejar de operar que no afecten la disponibilidad</li> </ul>
<b>Gestión del Servicio</b>	Debe estar registrado en el portafolio de servicios	Solicitar su registro al responsable
	Debe tener un dueño que define la evolución necesaria del servicio (nuevas funciones, versiones, etc.)	Al inicio del proyecto se definirá el dueño (la mayor parte de las veces es el equipo de solicita el servicio)

*Cuadro 16. Lineamientos generales del proceso de Generación de servicios*

#### **4.5. Aplicación de la Arquitectura**

##### 4.5.1. Análisis de la propuesta de solución de Factoring Electrónico

#### **Modelado de Procesos de Negocio**

Frecuentemente los sistemas (conjuntos de procesos y subprocesos integrados en una organización) son difíciles de comprender, amplios, complejos y confusos; con múltiples puntos de contacto entre sí y con un buen número de áreas funcionales, departamentos y puestos implicados. Un modelo puede dar la oportunidad de organizar y documentar la información sobre un sistema.

Pero ¿qué es un modelo? Un modelo es una representación de una realidad compleja. Modelar es desarrollar una descripción lo más exacta posible de un sistema y de las actividades llevadas a cabo en él.

Cuando un proceso es modelado, con ayuda de una representación gráfica (diagrama de proceso), pueden apreciarse con facilidad las interrelaciones existentes entre distintas actividades, analizar cada actividad, definir los puntos de contacto con otros procesos, así como identificar los subprocesos comprendidos. Al mismo tiempo, los problemas existentes pueden ponerse de manifiesto claramente dando la oportunidad al inicio de acciones de mejora.

Diagramar es establecer una representación visual de los procesos y subprocesos, lo que permite obtener una información preliminar sobre la amplitud de los mismos, sus tiempos y los de sus actividades.

La representación gráfica facilita el análisis, uno de cuyos objetivos es la descomposición de los procesos de trabajo en actividades discretas. También hace posible la distinción entre aquellas que aportan valor añadido de las que

no lo hacen, es decir que no proveen directamente nada al cliente del proceso o al resultado deseado. En este último sentido cabe hacer una precisión, ya que no todas las actividades que no proveen valor añadido han de ser innecesarias; éstas pueden ser actividades de apoyo y ser requeridas para hacer más eficaces las funciones de dirección y control, por razones de seguridad o por motivos normativos y de legislación.

Diagramar es una actividad íntimamente ligada al hecho de modelar un proceso, que es por sí mismo un componente esencial en la gestión de procesos.

A continuación presentamos el modelado de procesos de negocio realizado con la herramienta IBM Websphere Modeler (Para una explicación de la herramienta específica utilizada para modelar en esta investigación Ver ANEXO 1), siguiendo los pasos de la metodología BPM (Business Process Management, Administración de procesos de Negocio), así mismo las matrices de procesos y roles (cuadros 17,18 y 19) del producto de Factoring electrónico en la Caja Metropolitana de Lima. Estos procesos se dividen en tres, que son los principales en el ciclo de vida del otorgamiento de un crédito de Factoring, y a saber son:

- Proceso de Otorgamiento de Línea de Financiamiento de Factoring

Que a su vez está compuesto por los Sub-procesos:

- Solicitud de Línea de Factoring
- Evaluación de Línea de Factoring
- Garantización de Línea de Factoring
- Aprobación de Línea de Factoring

- Activación de Línea de Factoring
  
- Desembolso de Financiamiento de Factoring

Que a su vez está compuesto por los Sub-procesos:

- Coordinación Inicial
- Recepción de documentos valorados
- Remesa de documentos valorados
- Calificación de documentos valorados
- Desembolso del crédito

Y el proceso de

- Pago de Documentos Valorados

## A. Proceso de Otorgamiento de Líneas de Factoring

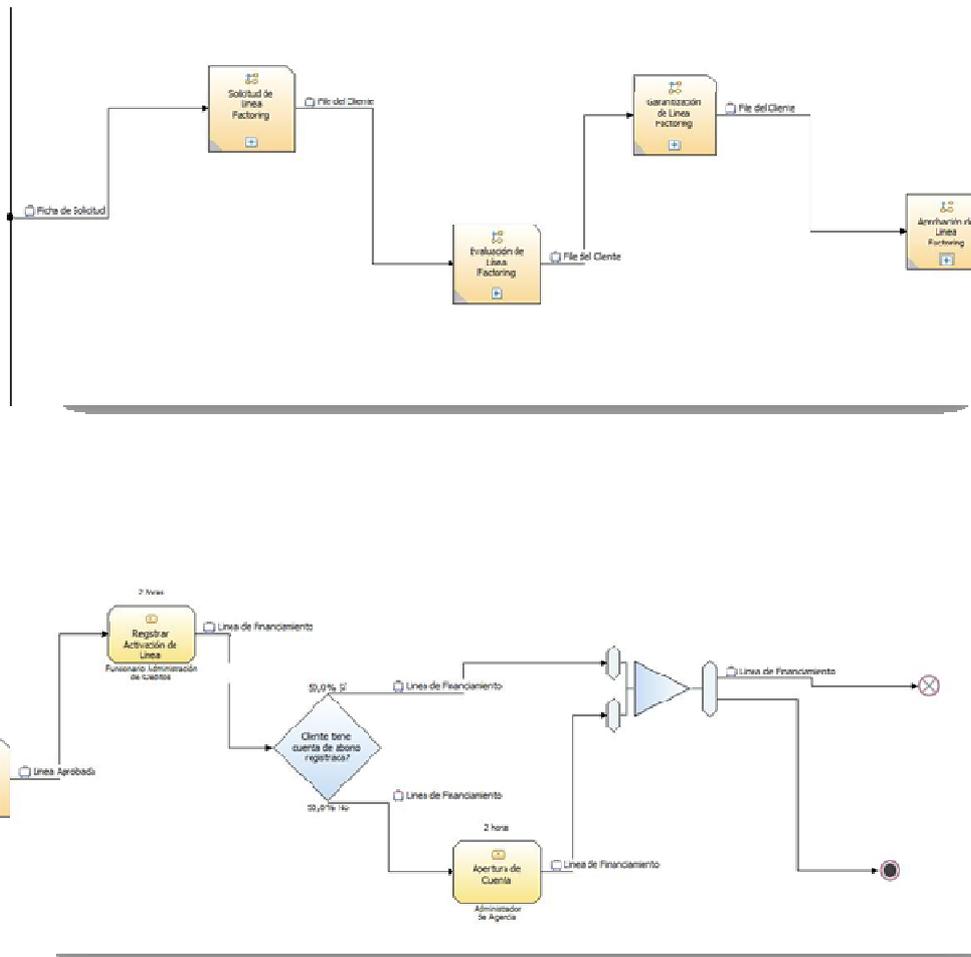


Figura 42. Modelado de Proceso de Negocio Otorgamiento de Líneas de Factoring

## Sub-proceso: Solicitud de Línea de Factoring

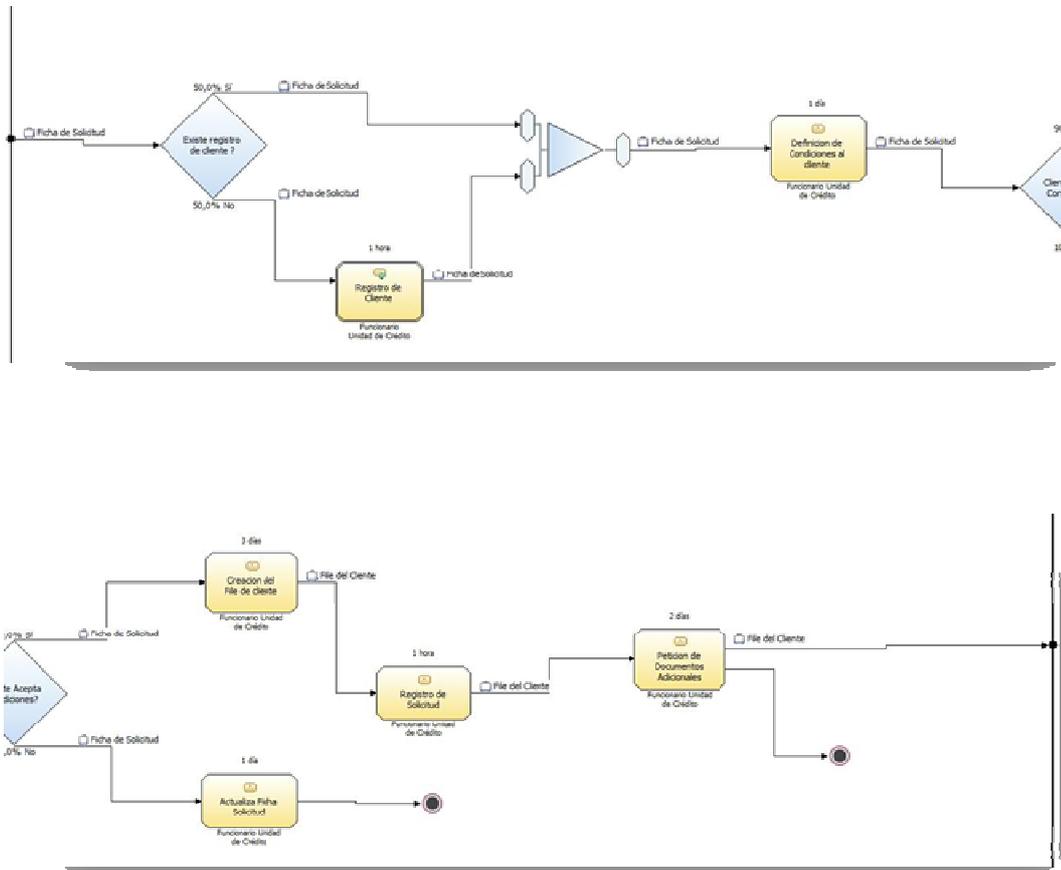


Figura 43. Modelado de Sub-Proceso de Negocio Solicitud de Línea de Factoring

## Sub-proceso: Evaluación de Línea de Factoring

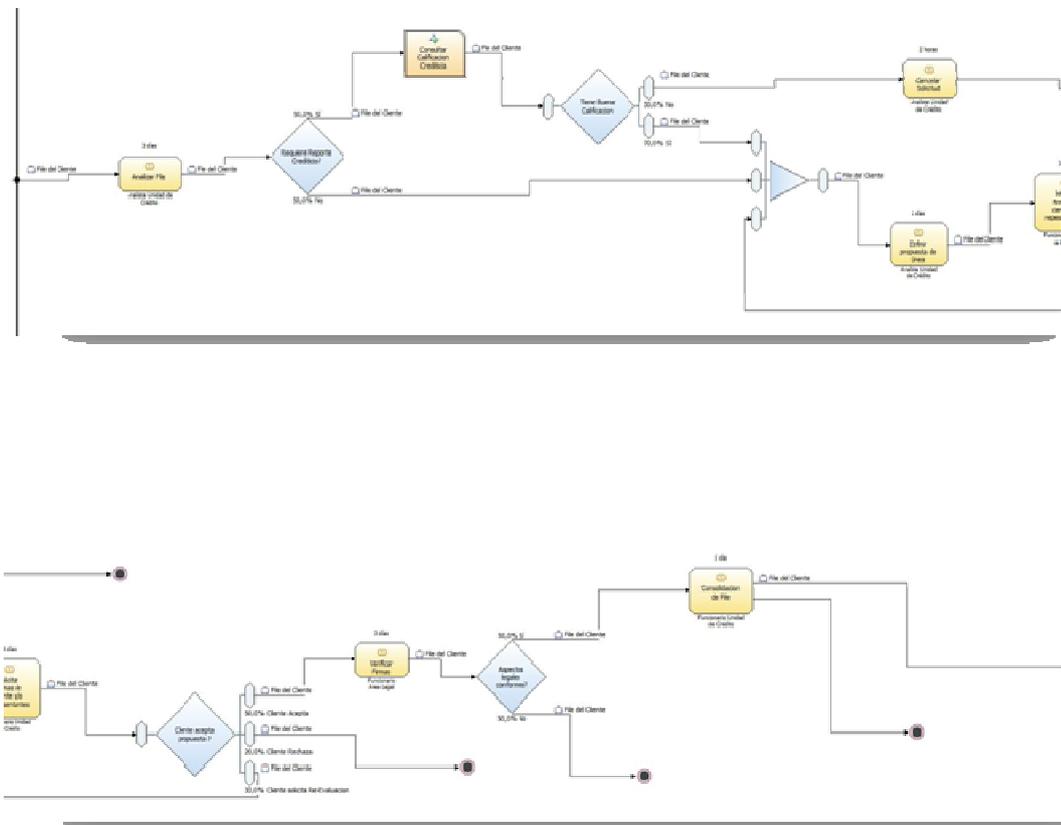


Figura 44. Modelado de Sub-Proceso de Negocio Evaluación de Línea de Factoring

## Sub-proceso: Garantización de Línea de Factoring

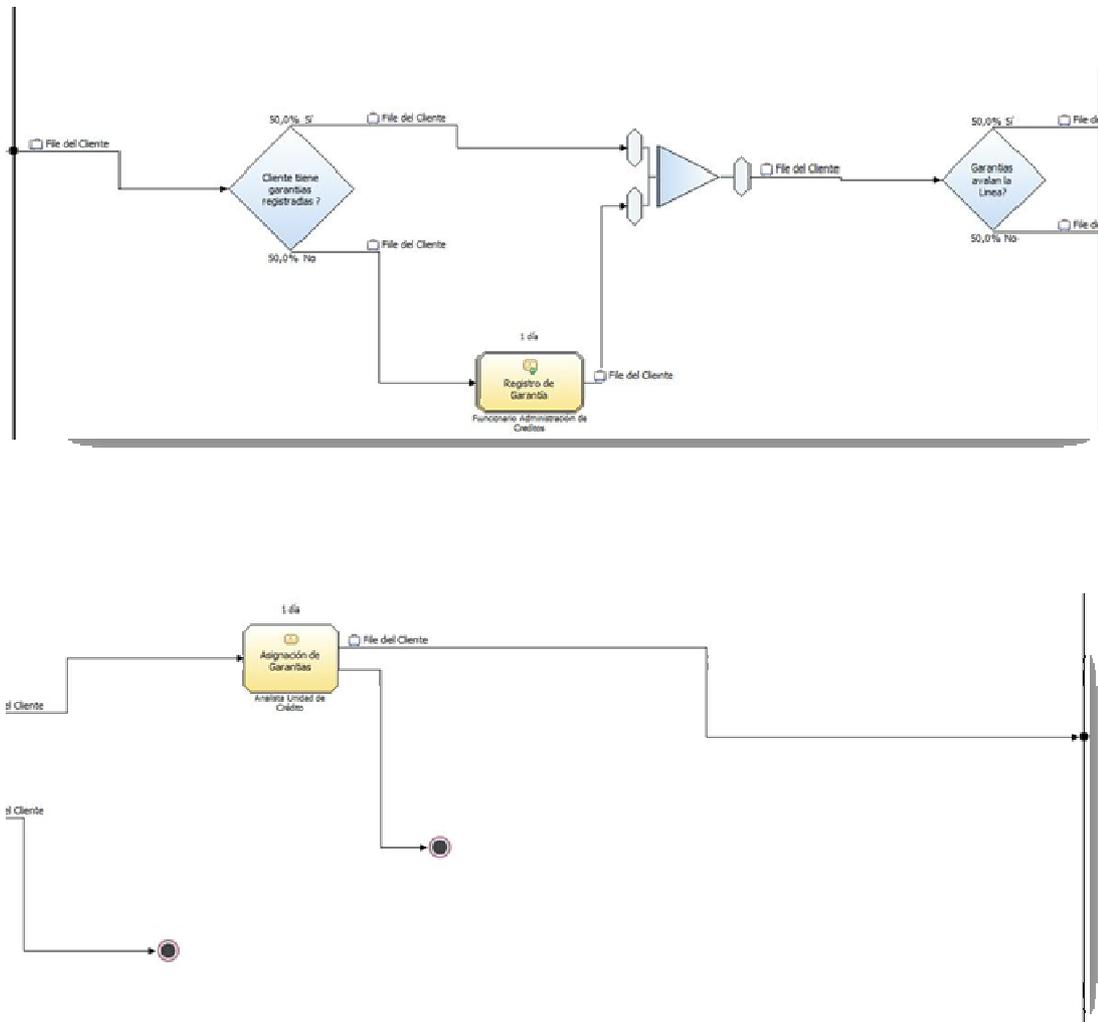


Figura 45. Modelado de Sub-Proceso de Negocio Garantización de Línea de Factoring

## Sub-proceso: Aprobación de Línea de Factoring

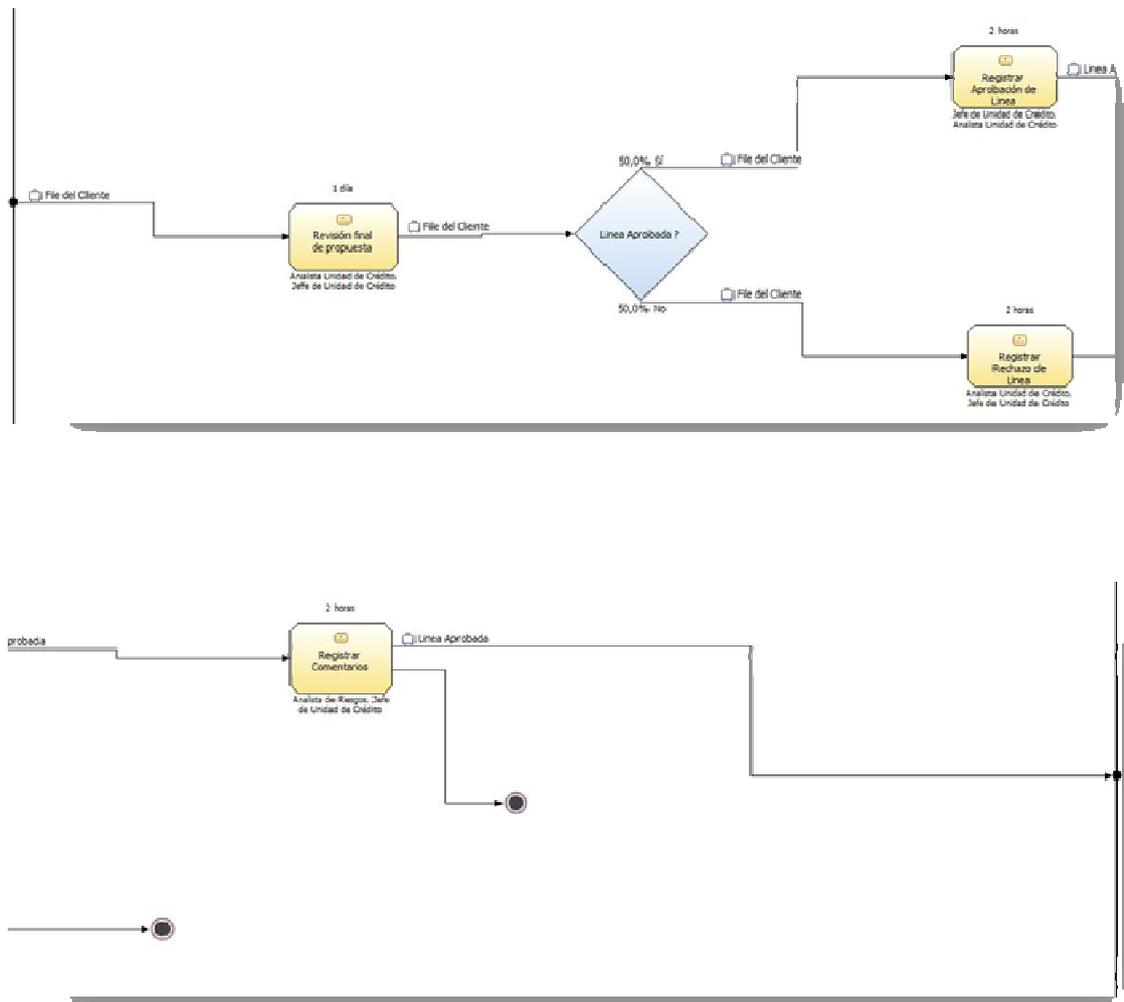


Figura 46. Modelado de Sub-Proceso de Negocio Aprobación de Línea de Factoring

<b>Proceso de Negocio</b>	<b>Tarea / Actividad</b>	<b>Rol</b>
Solicitud de Línea de Factoring	Indicar al cliente Requerimientos	Funcionario Unidad de Crédito
	Recepción de ficha de Solicitud y documentos requeridos para evaluación del cliente	
	Ingresar Cliente al sistema	
	Registro en el Sistema de la Solicitud	
	Creación del File del Cliente	
	Entrega del File al Analista de Unidad de Crédito	
Evaluación de Línea de Factoring	Recepción del File del Funcionario de Unidad de Crédito	Analista de Unidad de Crédito
	Análisis File del Cliente	
	Obtener información Crediticia del cliente	
	Elaboración de la propuesta de línea de Factoring	
	Visto bueno de la propuesta y adjuntar al File del Cliente	
	Informar a la Jefatura de Unidad de Crédito	
	Entrega del File al Funcionario de Unidad de Crédito	Funcionario Unidad de Crédito
	Recepción del File del Analista de Unidad de Crédito	
	Entrega y Solicitud de firma del cliente y/o representantes	
	Solicitar conformidad de Firmas del cliente al área Legal	
	Verificación de Firmas y aspectos legales del contrato	Funcionario Área Legal
	Recepción de conformidad de firmas	Funcionario

	Entrega del File del cliente y Conformidad del Área legal al Analista de Unidad de Crédito	Unidad de Crédito
Garantización de Línea	Verificación de las garantías del cliente	Funcionario de Administración de Créditos
	Registro de Garantías desde en el sistema	
Aprobación de Línea de Factoring	Asignación de Garantías en el modulo de Garantías	Analista de Unidad de Crédito
	Recepción de file del cliente del funcionario de unidad de crédito	Analista de Unidad de Crédito / Jefe Unidad de Crédito
	Revisión final de la propuesta	
	Registrar Aprobación (Rechazo) de Línea de Financiamiento	
	Envío del file de cliente y contrato al Analista de Riesgos	
	Recepción de file del cliente del Analista de unidad de crédito / Jefe de Unidad de Créditos según sea el caso	
	Registro de Comentarios	
Verificación de documentación		
Envío del file de cliente y contrato al funcionario de Administración de Crédito		
Activación de Línea de Factoring	Recepción de file de cliente y contrato del funcionario de unidad de crédito	Funcionario de Administración de Créditos
	Registrar Activación de la línea de financiamiento	
	Verificación o Creación de cuenta para abono del desembolso	

Archiva el file del cliente	
Envío al Servicio de Valores del contrato	
Envío de conformidad de Activación a Unidad de créditos	
Recepción del contrato de Factoring del funcionario de Administración de créditos	Funcionario de Servicio de Valores
Archivar contrato de factoring	
Recepción de conformidad de Activación de línea del Funcionario de Administración de créditos	Funcionario de Unidad de Créditos

*Cuadro 17. Matriz de Procesos y Roles de Negocio en el Otorgamiento de Líneas de Factoring en la Caja Metropolitana*

## B. Proceso de Desembolso de Financiamiento

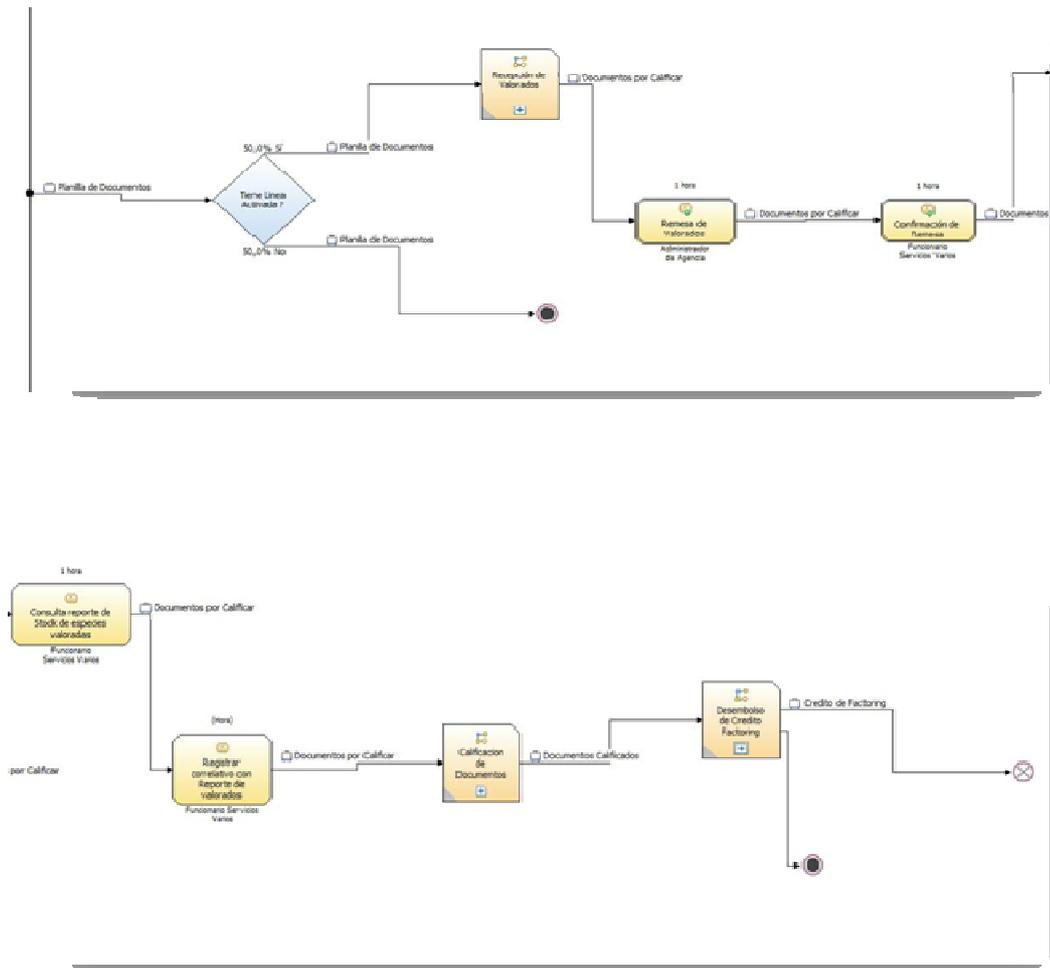


Figura 47. Modelado de Proceso de Negocio Desembolso de Financiamiento

## Sub-proceso: Recepción de documentos valorados

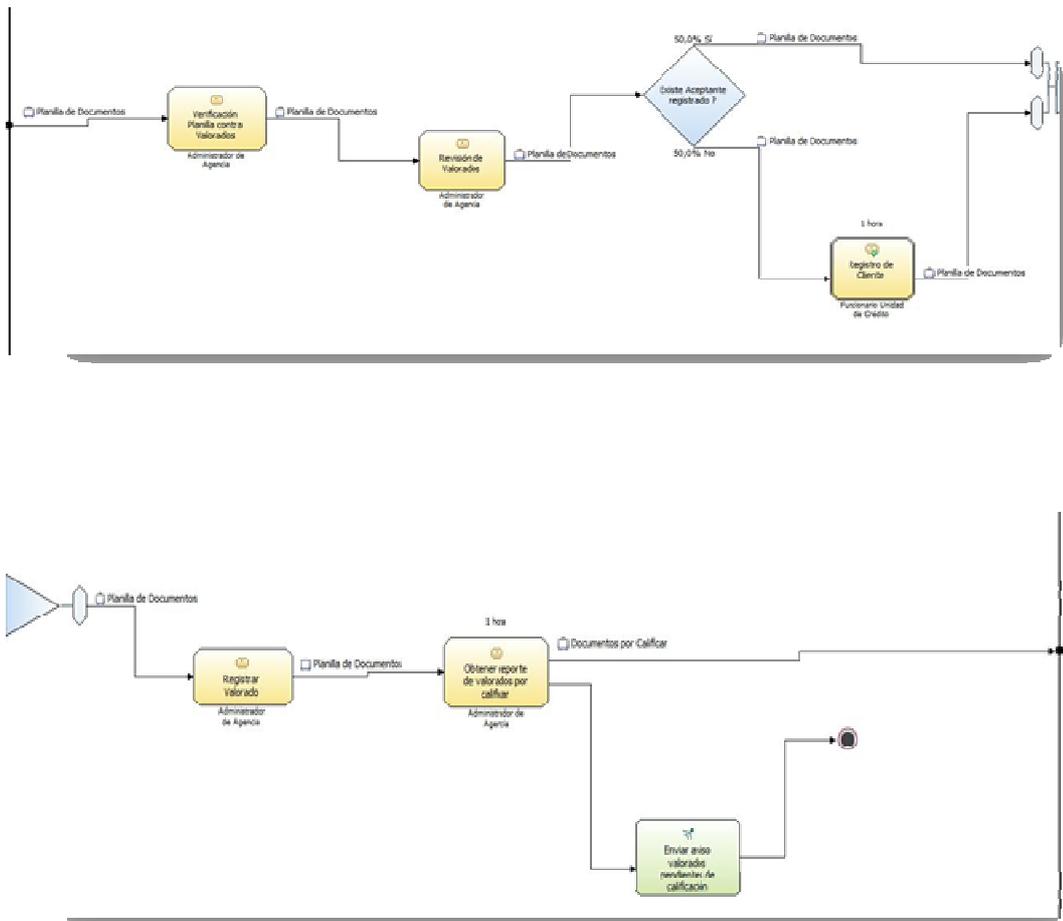


Figura 48. Modelado de Sub-Proceso de Negocio Recepción de documentos valorados

Sub-proceso: Calificación de documentos valorados

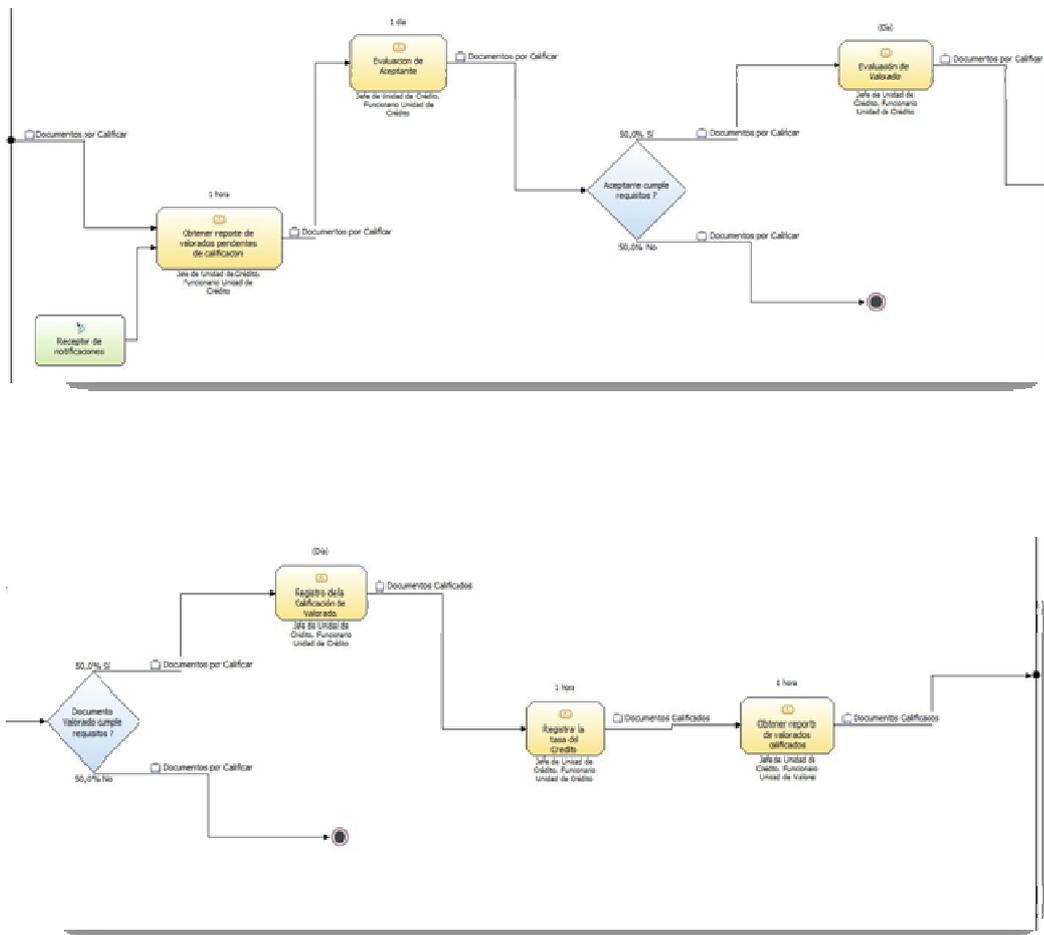


Figura 49. Modelado de Sub-Proceso de Negocio Calificación de documentos valorados

## Sub-proceso: Desembolso de crédito de Factoring

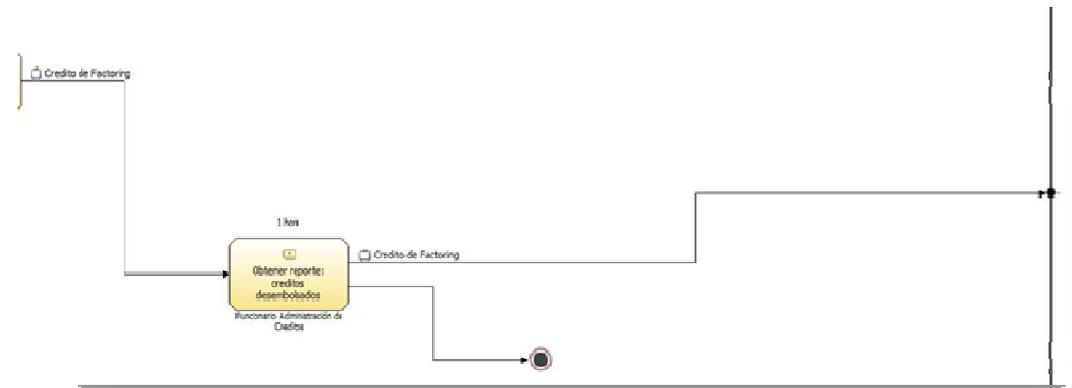
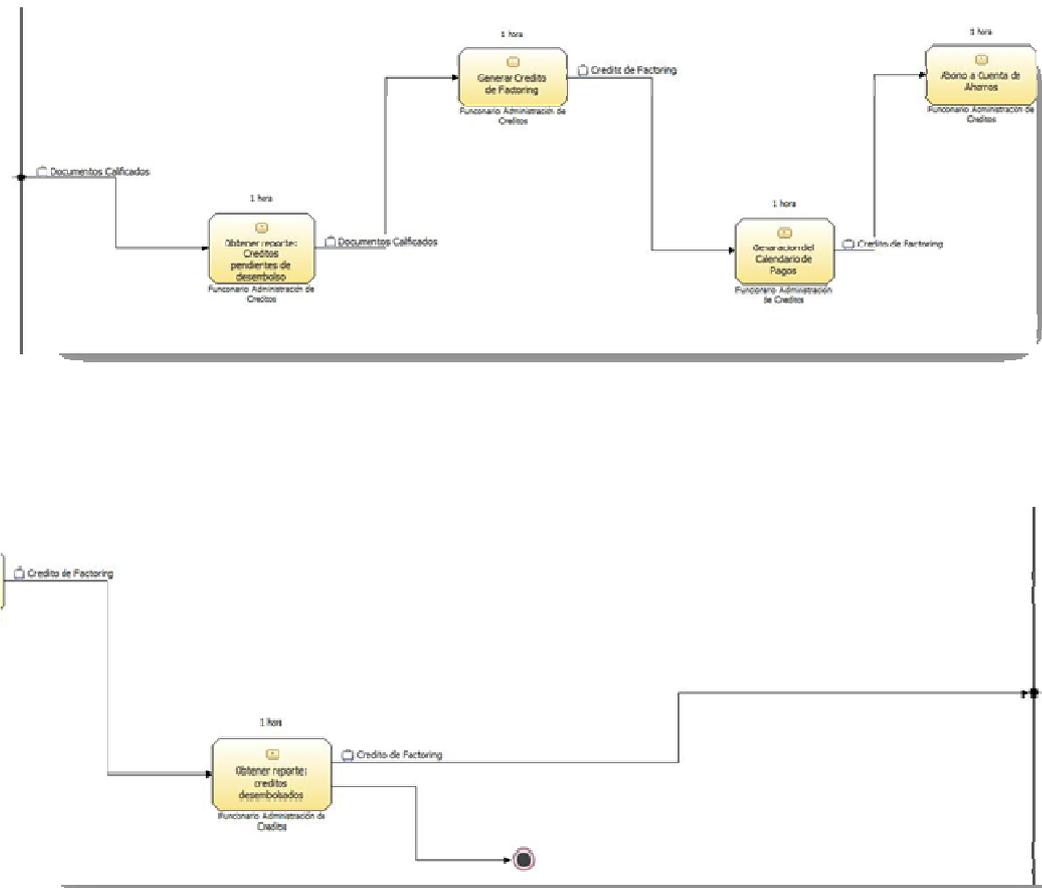


Figura 50. Modelado de Sub-Proceso de Negocio Desembolso de crédito de Factoring

<b>Proceso de Negocio</b>	<b>Tarea / Actividad</b>	<b>Rol</b>
Coordinacion Inicial	Verificación de Activación de línea de Factoring	Funcionario de Unidad de Crédito
	Entrega de formato de Planilla de documentos al cliente para que sea llenada	
	Envío de correo electrónico al Administrador de Agencia	
Recepción de documentos valorados	Recepción de la planilla de documentos y documentos físicos por parte del cliente	Administrador de Agencia
	Verificación de datos de la Planilla contra datos de documentos físicos	
	Verificar endoso de los documentos a favor de la Caja Metropolitana	
	Verificar autenticidad de los documentos y no vencimiento de los mismos	
	Registrar en el sistema a los Aceptantes de los documentos	
	Registrar la planilla de documentos en el sistema	
	Verificar obteniendo el reporte de valorados recibidos del sistema	
	Envío de correo electrónico al Jefe de Unidad de Crédito de documentos recibidos pendientes de calificación	
Remesa de Documentos	Realizar la Remesa de documentos de Agencias a	Administrador

	Servicios Varios en el sistema	de Agencia
	Realizar la remesa física de los documentos a través de empresa de transporte de valores	
	Recepción de la planilla de documentos y documentos físicos por parte del Administrador de la Agencia vía empresa de transporte de valores	
	Realiza en el sistema la confirmación de Remesa de Agencia a Servicios Varios	
	Consulta reporte de Stock de especies valoradas para confirmar el ingreso de documentos	Funcionario de Servicios Varios
	Obtiene reporte de Documentos recibidos y coloca numero autogenerado correlativo a los documentos utilizando numerador	
	Envía los valorados al funcionario de unidad de créditos para la calificación	
Calificación de Documentos	Recibe notificación por correo electrónico de planilla de documentos pendientes de calificación y los documentos	
	Obtiene reporte de Documentos pendientes de calificación	Jefe de Unidad de Crédito / Funcionario Unidad Crédito
	Evalúa cada uno de los aceptantes y documentos siguiendo los criterios establecidos por la norma	
	Procede al registro en el sistema de la calificación de documentos	
	Registrar la Tasa a aplicar a la operación de Factoring	

	Obtiene el reporte de documentos calificados y no calificados	
	Comunica a la Unidad de Administración de Créditos para que realice el desembolso	
	Instruye a servicios varios la devolución de los documentos rechazados en la calificación	
Desembolso del Crédito	Obtiene del sistema el reporte de créditos pendientes de desembolso	Funcionario de Administración de Créditos
	Registro del desembolso en el sistema	
	Obtiene del sistema el reporte de créditos desembolsados	
	Adjunta voucher de desembolso y reporte de crédito desembolsado al File del cliente	

*Cuadro 18. Matriz de Procesos y Roles de Negocio en el desembolso de créditos Factoring en la Caja Metropolitana*

## Proceso de Pago de Documentos Valorados

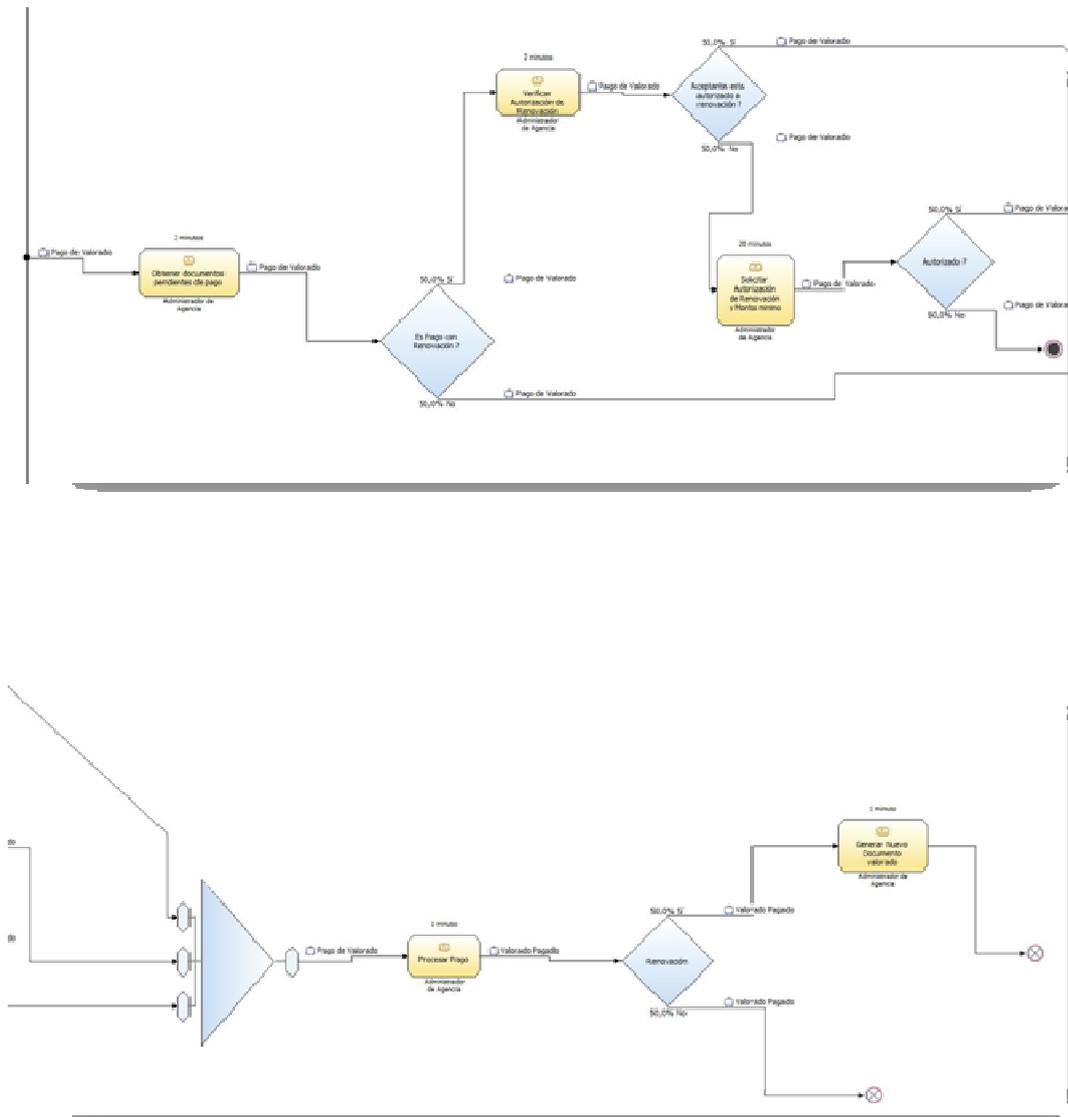


Figura 51. Modelado de Sub-Proceso de Negocio Pago de documentos valorados

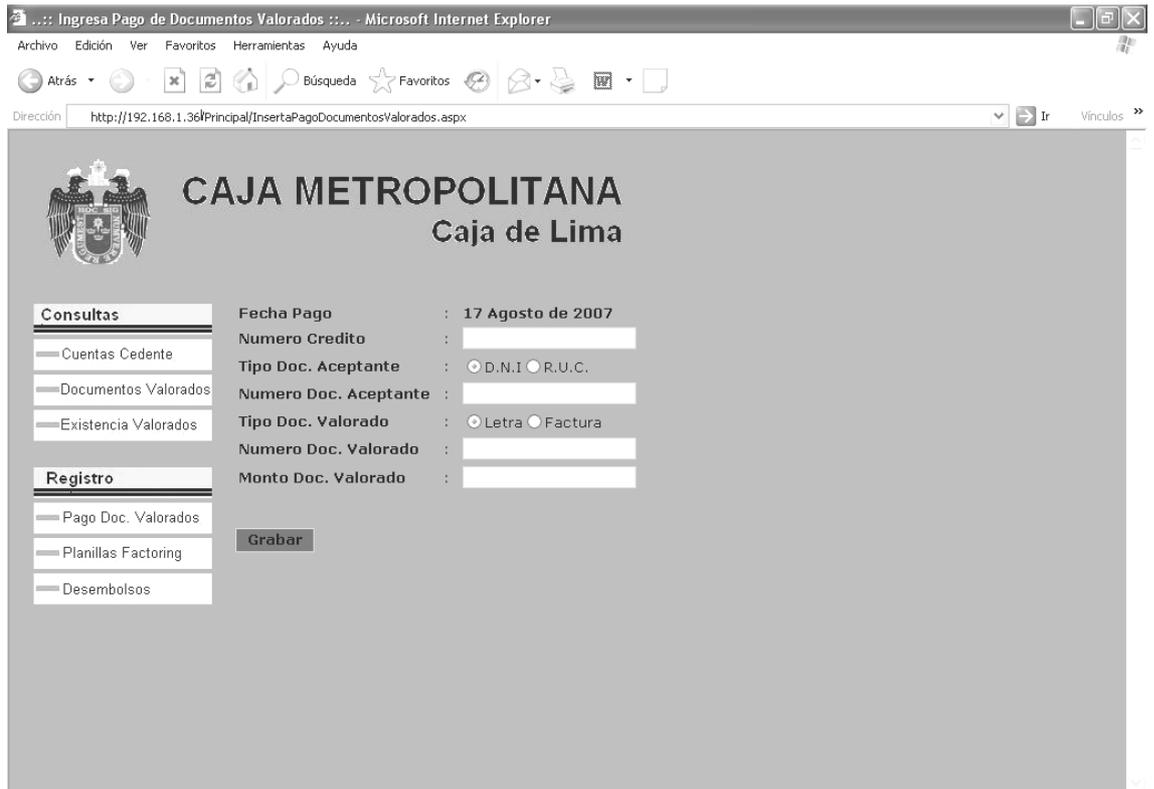
Proceso de Negocio	Tarea / Actividad	Rol
Pago de documentos	Obtener documentos pendientes de pago del cedente	Administrador de Agencia
	Si es pago con renovación comunica por correo al Jefe de Unidad de Crédito para Autorización	
	Autoriza o Rechaza la renovación en el sistema y le asigna el monto mínimo de renovación	Jefe Unidad de Credito
	Realiza en el sistema la cobranza del documento que el aceptante se acerco a cancelar pudiendo ser efectivo / cargo a cuenta / cheque certificado	Administrador de Agencia
	Informa al aceptante que el documento podrá ser recogido dentro de 3 días por norma de la institución	

*Cuadro 19. Matriz de Procesos y Roles de Negocio en el Pago de créditos de Factoring en la Caja Metropolitana*

#### 4.5.2. Procesos de negocio candidatos a convertirse en Servicios Web

Una vez modelados e identificaos los procesos de negocio necesarios para llevar a cabo la operación de Factoring, (esto puede realizarse en la mayoría de lo casos gracias a la existencia de un Reporte de Especificaciones Funcionales del producto), se identifican que proceso o procesos son los que se expondrán mediante Servicios Web que sean consumibles desde fuera de la organización y que se alinean a la estrategia tecnológica de la empresa. En nuestro caso estos procesos son los siguientes:

- **Pago de los documentos valorados por parte de los aceptantes de los financiamientos de Factoring a través de otras instituciones financieras** (Esto puede comprender otras Cajas Municipales, Cajas Rurales, Bancos etc.), esto a cambio de una comisión por cada operación para la entidad financiera, esto brindaría no solo una facilidad a los aceptantes al poder tener más puntos de pago – lo cual minimiza el riesgo de tener cuotas vencidas - sino además ampliaría la presencia de la Caja Metropolitana en lugares donde no tiene una sucursal o agencia, es decir fuera de Lima Metropolitana (Figuras 52, 53). Esto puede extenderse a cualquier tipo de crédito pues la orientación a servicios proporciona una herramienta transparente y fácil de invocar por cualquier institución.



*Figura 52. Interfaz que invoca al servicio web de Ingreso de Pago de Documentos Valorados*

```

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim nRetorno As Integer
    Try
        Dim objWs As New dataTx.insertData
        nRetorno = objWs.GrabarPagoValorados("800", "001", txtNumeroCredito.Text, "EJCG", Today, _
            rbDocAceptante.Selected.Value, txtNumDocAceptante.Text, rbDocValorado.Selected.Value, _
            txtNumDocValorado.Text, Double.Parse(txtMontoValorado.Text))
        If nRetorno = 0 Then
            lblMensaje.ForeColor = Drawing.Color.AliceBlue
            lblMensaje.Text = "Pago realizado exitosamente"
        Else
            lblMensaje.ForeColor = Drawing.Color.Red
            lblMensaje.Text = "Pago no se realizado"
        End If
    Catch ex As Exception
        lblMensaje.ForeColor = Drawing.Color.Red
        lblMensaje.Text = "Problemas en conexion a Red"
    End Try
End Sub

```

*Figura 53. Invocación al servicio web de Pago de Documentos Valorados*

- **Ingreso de planillas y documentos electrónicos digitalizados para pre-calificación de documentos valorados.** Esto permite una mayor

flexibilidad del proceso de calificación de documentos y evita que el cliente tenga que trasladarse físicamente con todos los documentos valorados a las oficinas de la Caja Metropolitana, pudiendo hacerlo desde cualquier lugar. Esto agiliza el proceso ya que los analistas de créditos y el área legal pueden en una primera instancia ir filtrando los documentos valorados y a los aceptantes que los emiten y que serán materia del Factoring. En resumen reduce los tiempos de evaluación en el proceso ( Figuras 54, 55).

...: Ingreso Planillas Factoring :... - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vínculos

Dirección http://192.168.1.36/Principal/InsertaPlanillasFactoring.aspx

**CAJA METROPOLITANA**  
Caja de Lima

**Consultas**

- Cuentas Cedente
- Documentos Valorados
- Existencia Valorados

**Registro**

- Pago Doc. Valorados
- Planillas Factoring
- Desembolsos

Fecha Planilla : 17 Agosto de 2007

Tipo Doc. Cedente :  D.N.I  R.U.C.

Numero Doc. Cedente :

Tipo Doc. Aceptante :  D.N.I  R.U.C.

Numero Doc. Aceptante :

Tipo Doc. Valorado :  Letra  Factura

Numero Doc. Valorado :

Fecha Emision Doc. Valorado : < septiembre de 2007 >

lun	mar	mié	jue	vie	sáb	dom
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Fecha Vencimiento Doc. Valorado : < septiembre de 2007 >

lun	mar	mié	jue	vie	sáb	dom
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Moneda Doc. Valorado :  Soles  Dolares

Monto Doc. Valorado :

Grabar

*Figura 54. Interfaz que invoca al servicio web de Ingreso de Planillas de Factoring*

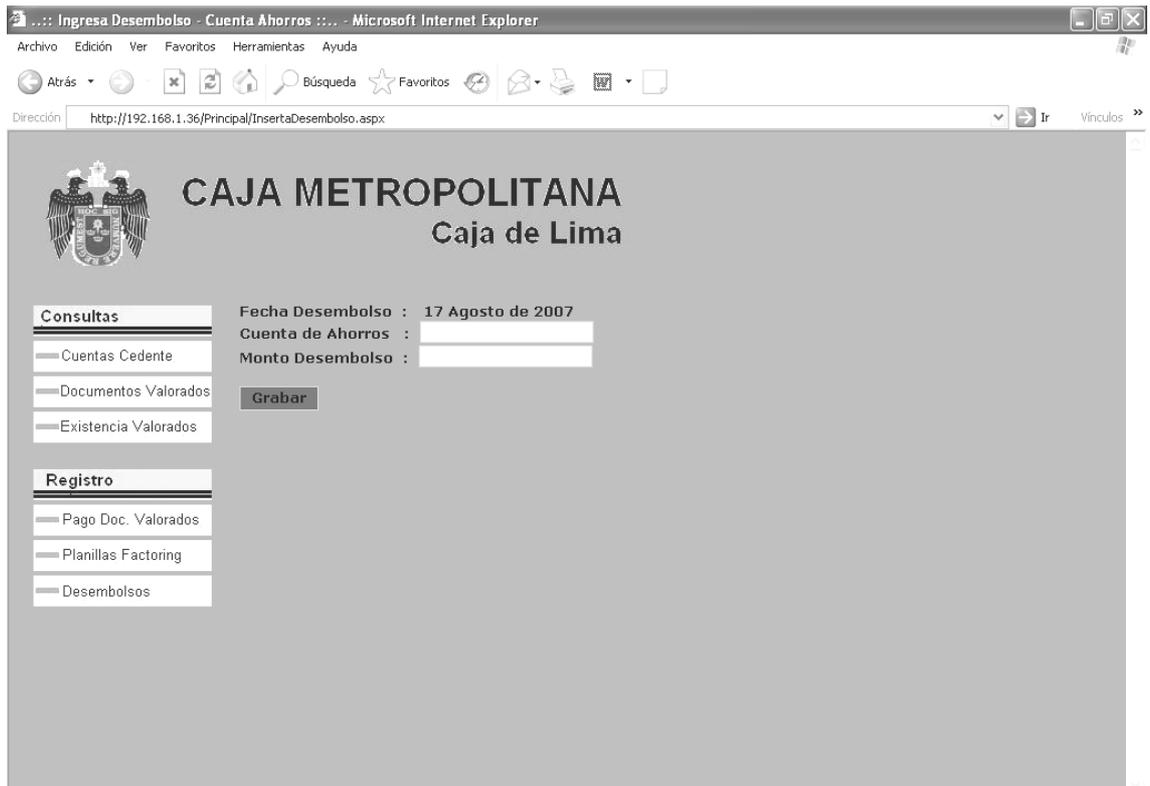
```

Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim sRetorno As Integer
    Try
        Dim objWs As New dataTx.insertData
        sRetorno = objWs.GrabarPlanilla("800", "001", "EJCG", rbDocCedente.SelectedVAlue, txtNumDocCedente.Text, _
            rbDocAceptante.SelectedVAlue, txtNumDocAceptante.Text, rbDocValorado.SelectedVAlue, _
            txtNumDocValorado.Text, clFechaEmision.SelectedDate, clFechaVencimiento.SelectedDate, _
            rbMoneda.SelectedVAlue, Double.Parse(txtMontoValorado.Text))
        If sRetorno <> "" Then
            lblMensaje.ForeColor = Drawing.Color.AliceBlue
            lblMensaje.Text = "Ingreso de Planilla se realizado exitosamente"
        Else
            lblMensaje.ForeColor = Drawing.Color.Red
            lblMensaje.Text = "Ingreso de Planilla no se realizado"
        End If
    Catch ex As Exception
        lblMensaje.ForeColor = Drawing.Color.Red
        lblMensaje.Text = "Problemas en conexion a Red"
    End Try
End Sub

```

*Figura 55. Invocación al servicio web de Ingreso de Planillas de Factoring*

- **Retiro desde la cuenta de ahorros del cliente, del monto desembolsado del financiamiento solicitado por parte del cliente desde cualquier entidad financiera.** Se ha observado que un gran número de clientes de la Caja Metropolitana opera no solo en la capital sino en diferentes localidades.. Esto permitiría a un cliente de un contrato de Factoring ubicado por ejemplo en provincias, no tener que desplazarse hasta las oficinas de la Caja Metropolitana en Lima para realizar su retiro. Este servicio brinda rapidez, y prontitud de la disponibilidad del dinero al cliente así como un ahorro de costos al mismo ( Figuras 56, 57).



*Figura 56. Interfaz que invoca al servicio web de Desembolso de Cuenta de Ahorros*

```

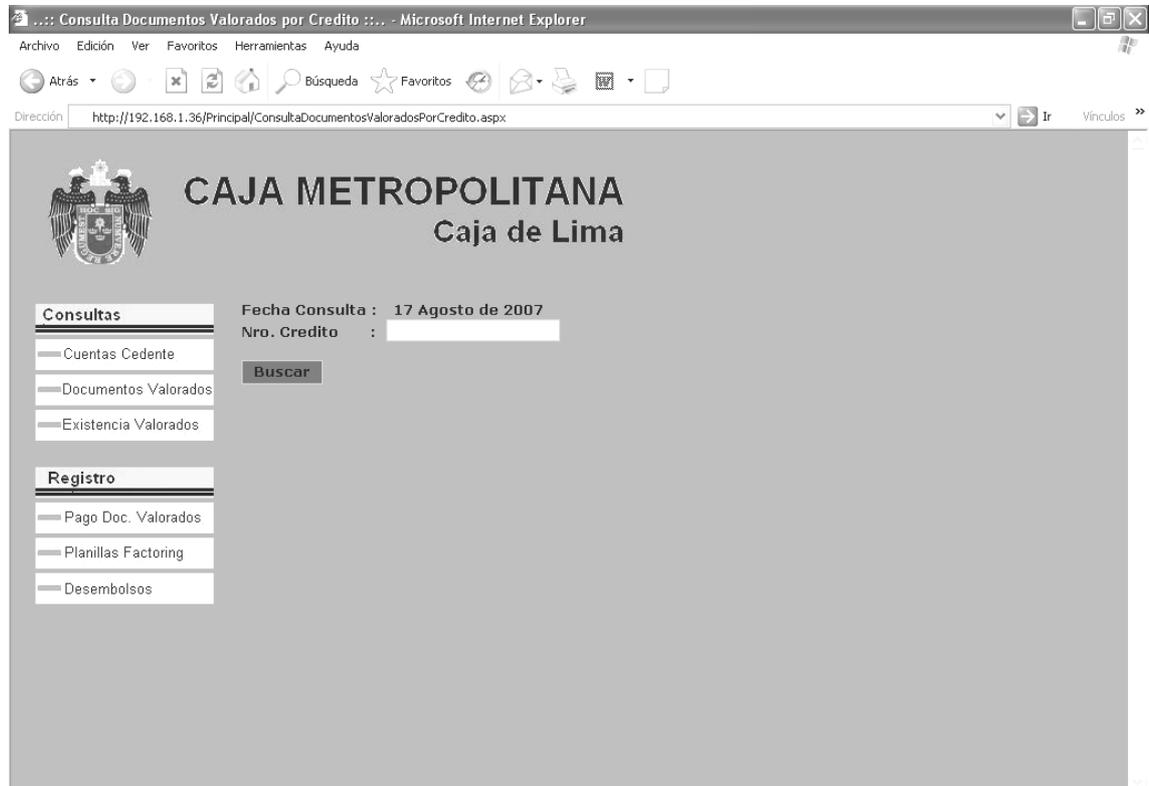
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim nRetorno As Integer
    Try
        Dim objWs As New dataTx.insertData
        nRetorno = objWs.GrabarDesembolsoCuentaAhorros("800", "001", "EJCG", txtCuenta.Text, _
            Double.Parse(txtMonto.Text))
        If nRetorno = 0 Then
            lblMensaje.ForeColor = Drawing.Color.AliceBlue
            lblMensaje.Text = "Desembolso realizado exitosamente"
        Else
            lblMensaje.ForeColor = Drawing.Color.Red
            lblMensaje.Text = "Desembolso no se realizado"
        End If
    Catch ex As Exception
        lblMensaje.ForeColor = Drawing.Color.Red
        lblMensaje.Text = "Problemas en conexion a Red"
    End Try
End Sub

```

*Figura 57. Invocación al servicio web de Desembolso de Cuenta de Ahorros*

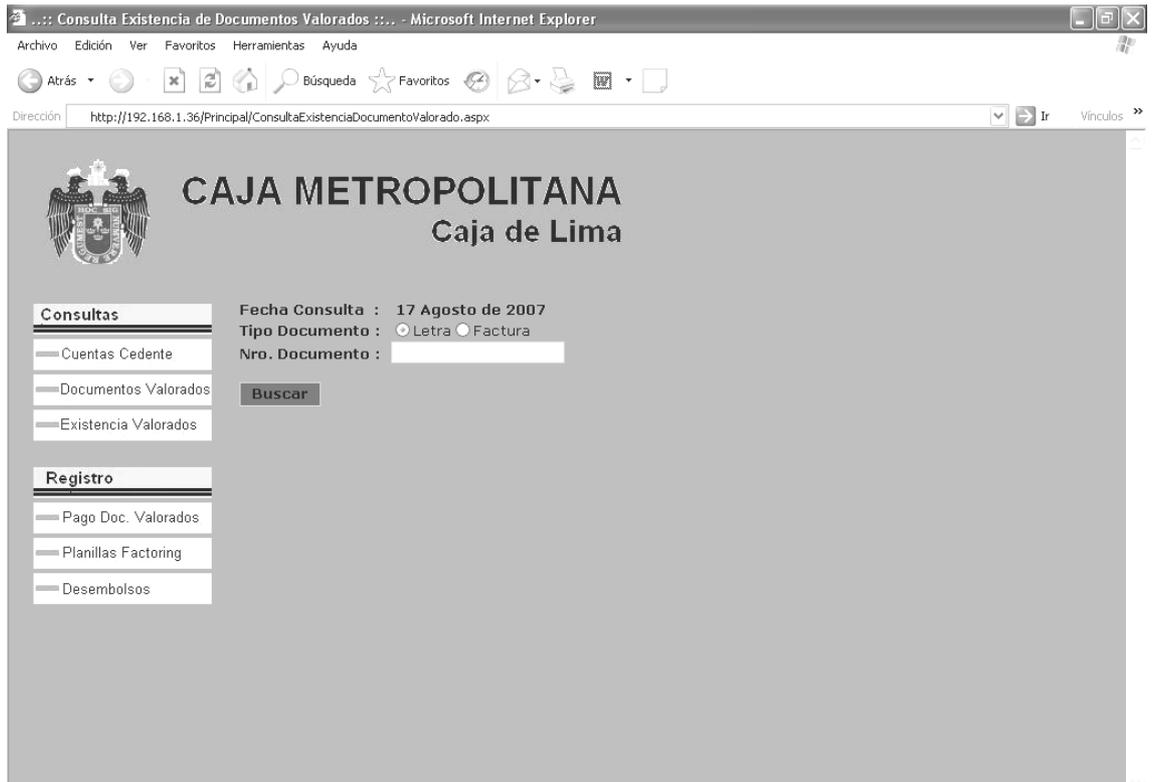
- **Consulta del Calendario del crédito (financiamiento), detalles de pago de los Aceptantes (Deudores del financiamiento) de los documentos desde sus propios sistemas de información.** Este servicio permite a los Cedentes (Clientes del financiamiento) llevar el control en línea de los pagos hechos por los deudores de los

documentos, Esto brinda una herramienta de consulta en línea a nuestros clientes lo cual mejora la calidad de nuestro servicio (Figura 58).



*Figura 58. Interfaz que invoca al servicio web de Consulta de Calendario de Créditos.*

- **Servicio de verificación de valorados a otras instituciones financieras**, este servicio permite conocer si un documento está siendo comprometido en algún tipo de crédito de Factoring en la Caja Metropolitana o si ha sido rechazado, los motivos que causaron el rechazo, esto para brindar a otras instituciones una mayor seguridad al otorgar sus propios créditos ( Figura 59)



*Figura 59. Interfaz que invoca al servicio web de Consulta de existencia de Documentos Valorados*

## **5. CONCLUSIONES**

1. En el presente trabajo hemos analizado una posible solución de la problemática actual en la institución -Caja Metropolitana- mediante la elaboración de una propuesta de arquitectura orientada a servicios basada en una serie de estándares aceptados en la industria de las Tecnologías de Información, desde una perspectiva tecnológica bajo el marco de referencia Microsoft para arquitecturas empresariales.
2. Se concluye que esta arquitectura propuesta es una ventajosa alternativa comparada con las actuales implementaciones en la empresa ya que ofrece independencia tecnológica, permite una amplia disponibilidad de funcionalidad existente en los aplicativos internos, permitirá la reutilización de código una vez que se encuentren desarrollados los servicios principales en el portafolio de servicios y permite la interoperabilidad con las demás Cajas Municipales para implementar procesos de negocio. Además la adopción de este modelo redundara en un menor tiempo de respuesta a los cambios del entorno.
3. En el aspecto conceptual de la arquitectura, esta se encuentra enmarcada en el escenario de arquitectura SOA tipo 1: Escenario de Creación de Servicios, y que permitirá construir e integrar aplicaciones internas para brindar una fachada transparente a los consumidores de estos servicios y

que además permite alinear estratégicamente los sistemas de información con los objetivos de negocio de la empresa

4. En el aspecto de vista lógica de arquitectura proponemos un modelo de 4 capas, las cuales se dividen en presentación (interfaz de usuario), servicios, negocio (lógica de aplicación y entidades) y datos (persistencia). Junto con las capas globales de seguridad y administración,
5. Asimismo concluimos por medio de la vista física, que la arquitectura debe ser de tipo web mediante la implementación de servidores de servicios web y de aplicaciones, mientras que en la tecnología de comunicación y mensajería se concluye que debe estar basada en estándares abiertos de comunicación y mensajería como son XML y SOAP los cuales son independientes de la plataforma y brindan un medio simple y estructurado de intercambio de información sobre el protocolo HTTP, es decir sobre Internet.
6. En el ámbito de plataforma de desarrollo, se opta por soluciones bajo la plataforma .NET de Microsoft. Esto debido a que la empresa en cuestión ya viene implementando gran parte de su infraestructura en base a este aliado tecnológico, otro motivo es que ya existen trabajos desarrollados siguiendo este modelo el cual es también es conocido por el personal técnico que trabaja en la organización y que garantiza una rápida curva de aprendizaje.
7. Asimismo se concluye que debe contarse con una metodología ordenada y precisa que defina cuales son las pautas a seguir en la creación de nuevos servicios ya que estos aun no existen en la organización, estas pautas, y lineamientos generales en la creación de servicios deben ser plasmadas por una metodología. Se propone una metodología para la Generación de

Servicios la cual ayudara a llevar una adecuada identificación, planeamiento, desarrollo, publicación, puesta en marcha, control de cambios y versionamiento de los mismos y así evitar un crecimiento desordenado de Servicios Web.

8. Por tanto se concluye en este trabajo que la arquitectura orientada a servicios propuesta ayudara a la organización a resolver la problemática de interoperabilidad de aplicaciones tanto internas como externas requeridas – en este caso el universo de Cajas Municipales- ya que posibilita una adecuada base para la construcción e integración de procesos de negocio en forma de servicios web, publicables, descubribles y consumibles por parte de aplicativos y usuarios finales. Además de brindar una independencia tecnológica en el ámbito de la construcción de nuevas soluciones intra-empresas. Ayuda a sentar las bases para que los futuros desarrollos aprovechen los beneficios de la reutilización de código y procesos, logrando un beneficio en el menor tiempo de respuesta que la organización requiere para sacar al mercado nuevos productos y servicios apoyándose en una ordenada arquitectura tecnológica.

## **6. RECOMENDACIONES**

1. El desarrollo de todo sistema o sistemas dentro de toda institución debe estar siempre acompañado de una arquitectura ya que mediante ella podemos fomentar la reutilización de módulos creados.
2. Tener en cuenta en todo momento la seguridad de la información, ya que es de vital importancia para la organización mantener la confidencialidad de su información fuera del alcance de los competidores.
3. Se debe tratar de usar Interfaces ya que en ellas tenemos solo la definición de los métodos de nuestras entidades de Negocio.
4. Es indispensable ubicar los componentes de Lógica de Negocio dentro de un Servidor de Componentes, la Capa de Presentación y las Entidades de Negocio dentro de un Servidor Web y nuestra Información dentro de un servidor de Base de Datos, ello debido a que podemos distribuir la carga de los procesos.
5. Si el acceso a nuestros procesos crece exponencialmente entonces debemos usar lo que la opción del Clustering nos ofrece.
6. Asimismo en conjunción con el punto anterior, si más adelante se requiere la conectividad de servicios entre plataformas heterogéneas –esto como consecuencia de ampliar la red de servicios a un rango mayor de consumidores - deberá considerarse la implementación de un Bus de Servicios Empresariales para su correcta comunicación.

## **7. REFERENCIAS BIBLIOGRAFICAS**

INFORMATIVO CABALLERO BUSTAMANTE, Revista de asesoría especializada Vol. 33 Nro. 567 2da Quincena Mayo 2005, Vol. 34 Nro.587 2da Quincena Marzo 2006, Vol. 34 Nro.602 1era Quincena Noviembre 2006.

LAMA More, Antonio, “Un nuevo enfoque para las pequeñas y medianas empresas”. Revista de la Facultad de Ciencias Económicas. 2005

PUEMAPE Postigo, Daniel, “El Factoring en el Perú, Aspectos jurídicos y económicos”. Artículo para maestría en Finanzas y derecho corporativo, ESAN. 2003.

TELLO Flores, Llim, “Análisis del Financiamiento de la Pequeña y Mediana Empresa”. 1998.

MAURICIO PACHAS, Pablo, “Prestamos caros a las Pymes le restan competitividad” Facultad de Ciencias Administrativas UNMSM

[http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07\\_n13/a03.htm](http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07_n13/a03.htm)

FLORES KONJA, Adrian Alejandro, "Metodología de gestión para las micro, pequeñas y mediana empresas en Lima Metropolitana" TESIS DIGITALES UNMSM

[http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Empre/flores\\_ka/cap02.PDF](http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Empre/flores_ka/cap02.PDF)

Superintendencia de Banca y Seguros, Estadísticas:

<http://www.sbs.gob.pe/>

"El Financiamiento para las Pymes" Pablo Mauricio Pachas

[http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07\\_n13/a03.htm](http://sisbib.unmsm.edu.pe/BibVirtual/Publicaciones/administracion/v07_n13/a03.htm)

Las fuentes de financiamiento a corto plazo

<http://www.monografias.com/trabajos33/fuentes-financiamiento/fuentes-financiamiento.shtml>

Introducción a la arquitectura de software, Microsoft

[http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/intro.msp](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.msp)

Blog de Arquitectura Orientada a Servicios (SOA)

<http://arquitecturaorientadaaservicios.blogspot.com/2006/06/principios-de-la-orientacin-servicios.html>

Service-Oriented Architecture Implementation Framework

<http://www.soablueprint.com/whitepapers/SOAPGPart2.pdf>

What Is SOA? An Introduction to Service-Oriented Computing

<http://www.whatissoa.com/>

Principios de la Orientacion a Servicios

<http://tecnoblog.entel.es/?p=23>

Caixa Galicia Tasaciones hipotecarias a golpe de ratón

[http://www.microsoft.com/spanish/msdn/spain/casestudies/cs\\_caixagalicia02.mspx](http://www.microsoft.com/spanish/msdn/spain/casestudies/cs_caixagalicia02.mspx)

IBM – Web Services

<http://www.ibm.com/developerworks/webservices/newto/>

Plan para la implementación exitosa de SOA

[http://huibert-aalbers.com/IT\\_Insight/Spanish/WhitePapers/ITI007Sp-SuccessfulSOAImplementation.pdf](http://huibert-aalbers.com/IT_Insight/Spanish/WhitePapers/ITI007Sp-SuccessfulSOAImplementation.pdf)

World Wide Web Consortium (W3C)

<http://www.w3.org/>

Web Services Interoperability Organization (WS-I)

<http://www.ws-i.org/>

Simple Object Access Protocol (SOAP)

<http://www.w3.org/2000/xml/Group/>

Web Services Description Language (WSDL)

<http://www.w3.org/2002/ws/desc/>

What is a SOA Service?

<http://www.digerateur.com/articles/whatIsAService.jsp>

SOA Blueprint

<http://www.soablueprint.com/index.html>

Microsoft Web Services

<http://www.desarrollaconmsdn.com/msdn/ServiciosWeb/default.aspx>

Oracle - Arquitectura Orientada a Servicios (SOA)

<http://www.oracle.com/global/es/products/technologies/soa/index.html>

IBM Arquitectura

<http://www.ibm.com/developerworks/architecture/>

SOA Agenda

<http://soaagenda.com/journal/>

SOA para negocios

<http://lucasian.com/soa/>

Microsoft Biztalk Server: Service Oriented Architecture Overview

<http://www.microsoft.com/biztalk/solutions/soa/overview.mspx>

Blog de tecnología

[www.mastermagazine.info/articulo/3391.php](http://www.mastermagazine.info/articulo/3391.php)

SOAP and performance

[www-106.ibm.com/developerworks/library/ws-quality.html](http://www-106.ibm.com/developerworks/library/ws-quality.html)

IBM – Lo que hace especial a SOA

[http://www-306.ibm.com/e-business/la/ar/soa\\_site/soa\\_2.shtml](http://www-306.ibm.com/e-business/la/ar/soa_site/soa_2.shtml)

La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real

[http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real\\_World\\_SOA.pdf](http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf)

IT Insight

<http://www.microsoft.com/biztalk/solutions/soa/overview.mspx>

## **LISTADO DE FIGURAS**

Figura 1. Organigrama actual de la CMCPL

Figura 2. Evolución de las colocaciones en la CMCPL en el periodo de Junio 2006 – Junio 2007

Figura 3. Distribución de Colocaciones en la Caja Metropolitana a la fecha

Figura 4. Arquitectura 2-capas dividida en dos entidades con el interfaz por un lado y las reglas de negocio junto con el Acceso a Bases de Datos por otro

Figura 5. Arquitectura 2 -capas con el acceso a la Base de Datos y las reglas de negocio junto a la interfaz de usuario

Figura 6. Proceso de acceso a datos en un modelo cliente/servidor

Figura 7. Interfaz Web de la solución de Caja Galicia – Consulta de Créditos

Figura 8. Interfaz Web de la solución de Caja Galicia – Amortizaciones

Figura 9. Interfaz Web de la solución de Caja Galicia – Amortizaciones II

Figura 10. Interfaz Web de la solución de Caja Galicia – Consulta de Créditos II

Figura 11. Fuente de financiamiento a Corto plazo sin garantías

Figura 12. Fuente de financiamiento a Corto plazo con garantías

Figura 13. Flujo operativo simplificado del Factoring.

Figura 14. Flujo operativo del Factoring detallado.

Figura 15. Distribución porcentual de Pequeñas y Medianas empresas.

Figura 16. Evolución del Factoring en Chile.

Figura 17. Actividades económicas que usan el Factoring.

Figura 18. Tipos de Valores involucrados en el Factoring.

Figura 19. Evolución desde una arquitectura clásica hacia una arquitectura SOA

Figura 20. Ejemplo de aplicación SOA

Figura 21. Elementos que interactúan con el Enterprise Service Bus

Figura 22. Escenarios de SOA y sus relaciones

Figura 23. Elementos básicos relacionados con las actividades

Figura 24. Dependencia entre múltiples actividades y un recurso

Figura 25. Relaciones entre una actividad y múltiples recursos

Figura 26: Fachada para ofrecer servicios de Grano Grueso

Figura 27: Modelo de sistema de interconexión entre entidades usando middleware

Figura 28: Modelo de sistema de interconexión entre la Caja Metropolitana y el CCE

Figura 29: Diagrama de Secuencia de utilización de un Servicio Web

Figura 30: Variantes de distribución en la Arquitectura de 2 capas

Figura 31: Arquitectura tradicional de 3 capas

Figura 32: Arquitectura de 5 capas

Figura 33: Vista Lógica de la Arquitectura Propuesta

Figura 34: Componentes de un Servicio Web

Figura 35: Componentes lógicos de acceso de datos y Origen de Datos

Figura 36: Vista Física de una Arquitectura tipo Web

Figura 37: Vista Física de una Arquitectura típica Cliente / Servidor

Figura 38: Fase Identificación de Servicios

Figura 39: Fase Planeamiento de Servicios

Figura 40: Fase de Realización de Servicios

Figura 41: Puesta en producción de Servicios

Figura 42. Modelado de Proceso de Negocio Otorgamiento de Líneas de Factoring

Figura 43. Modelado de Sub-Proceso de Negocio Solicitud de Línea de Factoring

Figura 44. Modelado de Sub-Proceso de Negocio Evaluación de Línea de Factoring

Figura 45. Modelado de Sub-Proceso de Negocio Garantización de Línea de Factoring

Figura 46. Modelado de Sub-Proceso de Negocio Aprobación de Línea de Factoring

Figura 47. Modelado de Proceso de Negocio Desembolso de Financiamiento

Figura 48. Modelado de Sub-Proceso de Negocio Recepción de documentos valorados

Figura 49. Modelado de Sub-Proceso de Negocio Calificación de documentos valorados

Figura 50. Modelado de Sub-Proceso de Negocio Desembolso de crédito de Factoring

Figura 51. Modelado de Sub-Proceso de Negocio Pago de documentos valorados

Figura 52. Interfaz que invoca al servicio web de Ingreso de Pago de Documentos Valorados

Figura 53. Invocación al servicio web de Pago de Documentos Valorados

Figura 54. Interfaz que invoca al servicio web de Ingreso de Planillas de Factoring

Figura 55. Invocación al servicio web de Ingreso de Planillas de Factoring

Figura 56. Interfaz que invoca al servicio web de Desembolso de Cuenta de Ahorros

Figura 57. Invocación al servicio web de Desembolso de Cuenta de Ahorros

Figura 58. Interfaz que invoca al servicio web de Consulta de Calendario de Créditos.

Figura 59. Interfaz que invoca al servicio web de Consulta de existencia de Documentos Valorados

## **LISTADO DE CUADROS**

Cuadro 1. Ranking de Colocaciones, Captaciones y Patrimonio por Caja Metropolitana a la fecha

Cuadro 2. Distribución de las colocaciones en la Caja Metropolitana a la fecha

Cuadro 3. Ranking de Colocaciones Comerciales por Caja Metropolitana a la fecha

Cuadro 4. Ranking de Colocaciones a Microempresas por Caja Metropolitana a la fecha

Cuadro 5. Ranking de Colocaciones Consumo por Caja Metropolitana a la fecha

Cuadro 6. Ranking de Colocaciones Hipotecario por Caja Metropolitana a la fecha

Cuadro 7. Proceso de acceso a datos en un modelo cliente/servidor

Cuadro 8. Principales problemas revelados por Microempresarios - Pymes

Cuadro 9. Ranking en dificultad de solucionar problemas revelados por Microempresarios - Pymes

Cuadro 10. Análisis FODA del Factoring

Cuadro 11. Elementos representables básicos de un modelado de procesos de negocio

Cuadro 12. Elementos representables avanzados de un modelado de procesos de negocio

Cuadro 13. Metodología Microsoft para el desarrollo de Arquitecturas de sistemas

Cuadro 14. Comparación entre Arquitecturas de Sistemas

Cuadro 15: Casos de Uso genéricos versus escenarios de arquitectura SOA

Cuadro 16. Lineamientos generales del proceso de Generación de servicios

Cuadro 17. Matriz de Procesos y Roles de Negocio en el Otorgamiento de Líneas de Factoring en la Caja Metropolitana

Cuadro 18. Matriz de Procesos y Roles de Negocio en el desembolso de créditos Factoring en la Caja Metropolitana

Cuadro 19. Matriz de Procesos y Roles de Negocio en el Pago de créditos de Factoring en la Caja Metropolitana

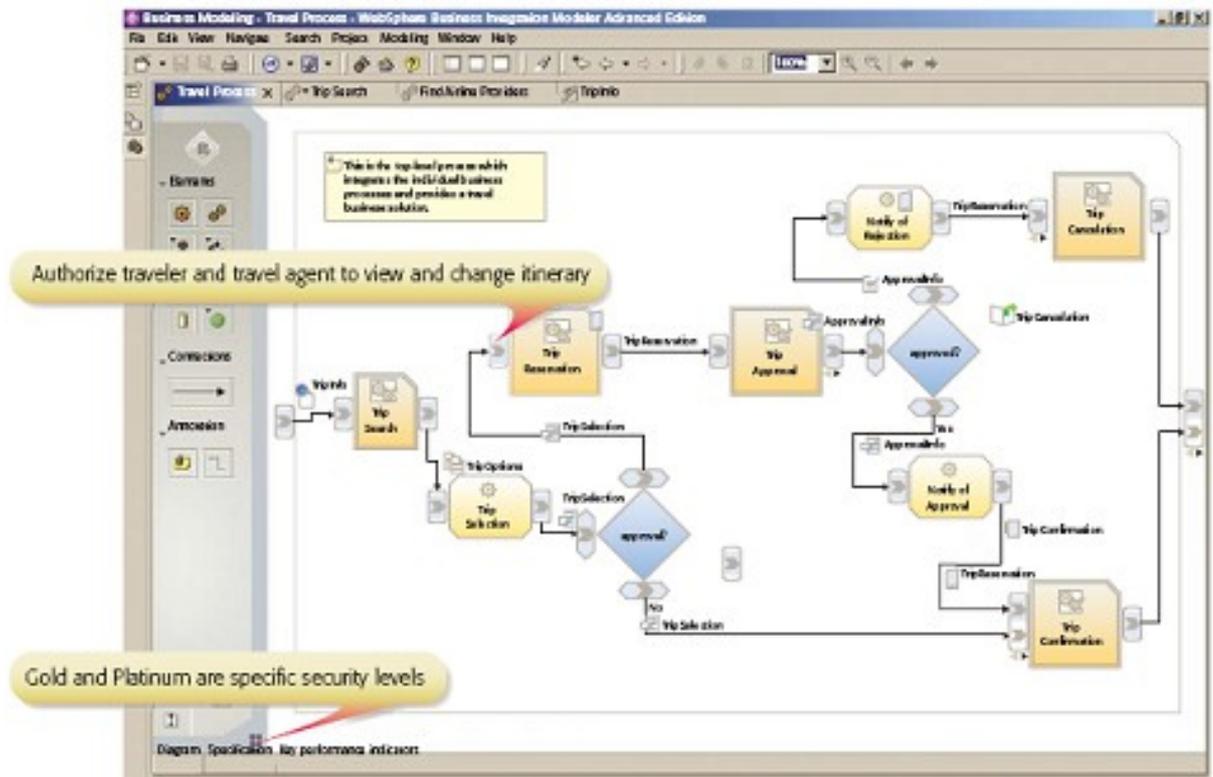
## 8. ANEXOS

### **ANEXO 1. Herramienta de Modelamiento de Procesos de Negocio:**

#### **WebSphere Business Modeler**

WebSphere Business Modeler es una herramienta de modelado construida según estándares abiertos y basada en Eclipse, una plataforma universal de herramientas de desarrollo de software de código abierto, lo cual le permite integrarse fácilmente con la arquitectura existente de una organización. El software se basa en BPEL (Business Process Execution Language), un estándar abierto basado en XML utilizado en la automatización de procesos de negocio basado en Web Services, y brinda soporte para exportar los modelos a formatos UML y las herramientas de desarrollo Rational Rose XDE.

Es especialmente útil para brindar valor a las organizaciones en el clima actual de fusiones y adquisiciones, tercerización de procesos de negocios, reingeniería de procesos y automatización de los procesos que antes eran manuales o semiautomáticos, debido a que las compañías desean evitar la dificultad y el gasto de re instrumentar un proceso que no cumple con el desempeño esperado.



Los productos de WebSphere Business Modeler son componentes básicos para la optimización y la innovación empresarial, que permiten apreciar el estado de los procesos empresariales en las compañías y la tecnología IT, contextualizar el estado en relación con los objetivos y las tendencias y, a continuación, actuar con rapidez para mejorar la ejecución. Una de las principales funciones básicas es la capacidad de analizar y modelar los procesos empresariales; ofrece herramientas de software para diseñar, simular y analizar procesos empresariales complejos con rapidez y eficacia. Los gestores de negocio y los analistas empresariales pueden diseñar y desarrollar más fácilmente flujos de procesos para mejorar la actividad de la empresa. Ofrece una opción de bajo coste para los usuarios empresariales que buscan una herramienta sencilla y fácil de utilizar para diseñar, documentar e imprimir los procesos empresariales.

Los productos de WebSphere Business Modeler le permitirán maximizar la eficacia empresarial mediante la optimización de procesos que le llevan el liderazgo en el entorno empresarial on demand actual; ayudan a las organizaciones a visualizar, comprender y documentar sus procesos de negocio. Los directores y los analistas de negocio pueden diseñar y desarrollar flujos de procesos para maximizar su efectividad. Incluye un motor de simulación que permite simular el comportamiento dinámico de un proceso, permitiendo el análisis de cargas y cuellos de botella.

Entre las características más importantes de esta herramienta tenemos:

- Permite visualizar análisis de los procesos, recursos, actividades y colas durante la simulación o después de la simulación.
- Los flujos se pueden animar en una simulación paso a paso.
- Permite exportar los procesos definidos a varios formatos, entre otros BPEL.
- Ofrece funciones complejas de análisis y simulación de modelos. Además, permite a los usuarios específicos de la tecnología IT exportar modelos a varios entornos de compilación, para impulsar el desarrollo de aplicaciones.
- Proporciona funciones de modelado y colaboración exhaustivas y fáciles de usar para el personal informático y de producción.
- Facilita la integración sin problemas de funciones de modelado, creación de flujos de trabajo y simulación.

- Proporciona simulación basada en un repositorio, informes de modificaciones de diseño y medición del rendimiento en tiempo real.
- Permite la supervisión del flujo de trabajo basada en Internet a través de un interfaz de navegador estándar.
- De fácil uso, permite tanto a las líneas de negocio como al departamento de TI modelar y diseñar el flujo de procesos antes de la implementación y complementa capacidades de modelado de SOA de Rational Software Architect.
- Proporciona funciones globales y muy intuitivas de colaboración y modelado de procesos de negocio para diseñar gráficamente procesos entre personas, partners y aplicaciones.
- Los negocios pueden volver a diseñar rápidamente procesos a medida que cambian sus necesidades.
- Soporte a múltiples metodologías de modelado y estándares del sector.
- Simula y valida procesos modelados.
- Proporciona un entorno de equipo para compartir y mantener versiones de modelos.
- Capacidad para transformar modelos de nivel de negocio en modelos y objetos de TI para minimizar la codificación.
- permite a los analistas empresariales documentar, diseñar, analizar y optimizar los procesos de negocio, así como sentar las bases para las soluciones de integración de los procesos.
- Comprender y transformar el negocio mediante funciones superiores de modelado de negocio, de simulación, de análisis y de colaboración.

- Definir situaciones y resultados de situaciones para obtener indicadores clave de rendimiento (KPI) y acciones puntuales y adecuadas de la unidad de medida.
- Analizar los resultados de la simulación para corregir correctamente los problemas de los modelos, como los atascos o los desequilibrios en la carga de trabajo.
- Ver los flujos de simulación graduales animados para datos en tiempo real. Las instantáneas de simulación están disponibles para su consulta.
- Mejorar el diseño de procesos mediante la simulación basada en hechos.
- Anticipar los cambios del mercado y realizar mejoras en el proceso para que el negocio sea flexible y adaptable.
- Generar informes a partir de modelos o resultados de la simulación mediante una gran variedad de plantillas de informes predefinidas o crear sus propios informes personalizados.
- Optimizar todos los procesos empresariales antes de desplegarlos y aumentar el ROI y la satisfacción del cliente.
- Documentar los procesos empresariales para que cumplan las normativas.
- Asegurar un despliegue rápido y preciso de las soluciones con una transferencia limpia a IT.

## **ANEXO 2. Formato de Publicación de Servicios**

<b>Nombre / Clase</b>	<i>Asignación de Garantías</i>
<b>Descripción Funcional</b> <b>(Objetivos)</b>	<i>Servicio Web que realiza la asignación de Garantías existentes del cliente a una solicitud de Línea de Financiamiento pasando el estado de la solicitud a GARANTIZADA</i>
<b>Responsables</b>	<i>Versión Actual : 2007.1.0</i> <i>Jefe de Equipo : Juan Rodriguez</i> <i>Arquitecto : Carlos Lozada</i> <i>Analista Funcional : Javier Palomino</i> <i>Analista Técnico : Augusto Redondo</i>
<b>Estado de desarrollo del Servicio</b>	<i>Versión:2007.2.0</i> <i>Planificación: Diciembre-2006</i> <i>Análisis Funcional / Técnico : Diciembre-2006</i> <i>Construcción / Certificación : Enero-2007</i> <i>Producción: Enero 2007</i> <i>Requerimiento: CMCP 823 OYM</i>
<b>Consumidores del servicio</b>	<i>Nombre: SICMECT – Negocio</i> <i>Tipo : Aplicación</i>
<b>Interfaces expuestas</b>	<i>Listar las interfaces que el servicio posee</i>
<b>Nivel de Servicio</b>	<i>Disponibilidad : 24x7</i> <i>Tiempo de respuesta :</i> <i>Volumen soportado :</i> <i>Concurrencia</i>
<b>Clasificación del Servicio</b>	<i>Tipo de Servicio: Venta</i> <i>Prioridad de Negocio: Alta</i>

Fuente: Elaboración propia