

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**Facultad de Ingeniería en Electricidad y Computación**

**“IMPLEMENTACIÓN DE UN SISTEMA DE INDEXACIÓN DISTRIBUIDA DE  
DOCUMENTOS PARA LA CONTRATACIÓN DE PRODUCTOS DE  
CRÉDITO EN UNA ENTIDAD FINANCIERA”**

**TESIS DE GRADO**

Previo a la obtención del título de:

**MAGÍSTER EN SISTEMAS DE INFORMACIÓN GERENCIAL**

Autor:

**ALMEIDA CAMPOS FAUSTO STANLEY**

Guayaquil – Ecuador

2017

## **AGRADECIMIENTO**

Agradezco principalmente a Dios por esta nueva meta alcanzada, por haber permanecido fiel a pesar del desánimo y las adversidades. Por haber puesto personas de valor a mi alrededor durante este proceso.

A la Escuela Superior Politécnica del Litoral por haberme abierto las puertas para cursar esta aventura y a los docentes que en ella participaron durante el proceso de formación ética y académica.

A la doctora Phd. Mónica Villavicencio C., por haberme dirigido durante la elaboración de este documento, por su profesionalismo, paciencia y dedicación.

A mi madre, a mi hermana y a mi novia, ya que han sido de consejeros cada vez que mis prioridades se desorganizaban, y debía retomar este proyecto para alcanzar lo que tanto anhelaba.

## **DEDICATORIA**

A mi padre, que en paz descansa, a mi madre y a mi hermana cuyos legados de aprendizaje, superación y entrega continúan hasta el día de hoy.

A mi novia, futura esposa, como fruto de su dedicación y apoyo durante este proceso.

## TRIBUNAL DE SUSTENTACIÓN

---

Mgs. Lenin Freire C.

Director MSIG

---

Phd. Mónica Villavicencio C.

Tutor de Tesis

---

Mgs. Carlos Martín B.

Miembro del Tribunal

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de ESPOL)

.....  
**Fausto Stanley Almeida Campos**

## RESUMEN

El trabajo de titulación presentado a continuación abordará un caso de estudio de una entidad financiera ecuatoriana cuyo proceso de contratación de productos de crédito fue rediseñado con la finalidad de agilizar los tiempos de respuesta al cliente y asegurar una correcta administración de los respaldos digitales de los documentos obtenidos o generados a través de dicho proceso de contratación.

Este documento presenta de forma detallada la situación de la empresa en cuanto al proceso de negocio de contratación de productos de crédito de consumo (llamado Multi-Crédito), cuya problemática estaba directamente asociada al tiempo que tardaba el traslado de los documentos físicos necesarios para habilitar dicha contratación. Luego de esto, propone una solución amparada en tecnología existente y las herramientas que serán necesarias para construir a dicha solución.

El presente trabajo también describe conceptos asociados a la digitalización de documentos, revisa brevemente los sistemas de gestión de contenido empresarial más relevantes y las tecnologías más apropiadas para solucionar el caso planteado anteriormente.

Avanza de forma ordenada por un proceso de análisis y diseño en donde las necesidades específicas de los futuros usuarios de la solución son tomadas en cuenta para implementar una herramienta personalizada que permitiera cargar documentos digitales dentro del sistema de gestión de contenidos de la empresa, y la posterior consulta de estos documentos.

Finalmente, se describen los interesantes resultados obtenidos luego de la implementación de la herramienta, tales como: ahorro económico de más del 90% en cuanto a la inversión necesaria frente a otras alternativas y la disminución de más del 90% del tiempo de traslado y validación de documentos de los clientes durante los procesos de contratación de productos.

## ÍNDICE GENERAL

AGRADECIMIENTO.....	II
DEDICATORIA.....	III
TRIBUNAL DE SUSTENTACIÓN.....	IV
DECLARACIÓN EXPRESA.....	V
RESUMEN.....	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABLAS.....	XV
INTRODUCCIÓN.....	XVII
<b>CAPÍTULO 1 GENERALIDADES .....</b>	<b>1</b>
1.1 ANTECEDENTES.....	1
1.2 DESCRIPCIÓN DEL PROBLEMA .....	2
1.3 SOLUCIÓN PROPUESTA.....	4
1.4 OBJETIVO GENERAL.....	9
1.5 OBJETIVOS ESPECÍFICOS.....	9
1.6 METODOLOGÍA .....	10
1.6.1 Revisión de literatura.....	10
1.6.2 Búsqueda de herramientas o soluciones existentes .....	11
1.6.3 Comparación de herramientas o soluciones existentes .....	12
1.6.4 Personalización de herramientas .....	13
1.6.5 Pruebas realizadas.....	13
<b>CAPÍTULO 2 MARCO TEÓRICO .....</b>	<b>15</b>
2.1 CONCEPTOS GENERALES DE LA DIGITALIZACIÓN DE IMÁGENES.....	15
2.1.1 Contenido y Documentos .....	17
2.1.2 Escaneo .....	17
2.1.3 Categorización .....	18

2.1.4	Indexación.....	18
2.1.5	Repositorios .....	18
2.1.6	Almacenamiento.....	19
2.1.7	Búsqueda y obtención.....	19
2.2	SISTEMAS DE GESTIÓN DE CONTENIDO EMPRESARIAL.....	20
2.2.1	Hyland OnBase .....	21
2.2.2	Alfresco .....	29
2.2.3	Breve comparativa técnica: OnBase vs. Alfresco.....	32
2.3	TECNOLOGÍAS UTILIZADAS EN LA IMPLEMENTACIÓN DE LA SOLUCIÓN.....	34
2.3.1	Java Enterprise Edition .....	35
2.3.2	Java Advanced Imaging (JAI) .....	38
2.3.3	Java Web Start (JWS).....	40
2.3.4	Java API for XML Web Services (JAX-WS) .....	47
2.3.5	Message Transmission Optimization Mechanism (MTOM) .....	51
2.3.6	JBoss Application Server .....	54
CAPÍTULO 3	ANÁLISIS DE REQUERIMIENTOS .....	57
3.1	DEFINICIÓN DE LA SITUACIÓN ACTUAL .....	57
3.1.1	Aspecto técnico.....	57
3.1.2	Aspecto procedural .....	60
3.2	DEFINICIÓN DE LA SITUACIÓN ESPERADA.....	62
3.2.1	Proceso de digitalización en la punta.....	62
3.2.2	Proceso del aplicativo Gestor Documental.....	64
3.2.3	Proceso de indexación en segundo plano.....	66
3.3	REQUERIMIENTOS FUNCIONALES.....	67
3.4	ATRIBUTOS DE CALIDAD.....	70
CAPÍTULO 4	ANÁLISIS Y DISEÑO DE LA SOLUCIÓN .....	72
4.1	ANÁLISIS DE LA INTEGRACIÓN .....	72
4.1.1	Sistema de Solicitudes de Contratación de Servicios .....	74
4.1.2	Sistema de Gestión de Contenido Empresarial.....	75
4.2	DISEÑO DE LA SOLUCIÓN PROPUESTA.....	79
4.2.1	Vista de Escenarios.....	79
4.2.2	Vista Lógica.....	80

4.2.3	Vista de Desarrollo .....	86
4.2.4	Vista de Procesos .....	87
4.2.5	Vista Física.....	89
CAPÍTULO 5 DESARROLLO Y PRUEBAS DE LA IMPLEMENTACIÓN .		91
5.1	PLAN DE DESARROLLO .....	91
5.1.1	Lenguaje de programación.....	91
5.1.2	Herramientas de desarrollo .....	92
5.1.3	Herramienta de control de versiones.....	92
5.2	PLAN DE PRUEBAS .....	92
5.3	RESULTADOS DEL PLAN DE PRUEBAS .....	95
CAPÍTULO 6 ANÁLISIS DE RESULTADOS .....		99
6.1	DISMINUCIÓN DEL TIEMPO PROMEDIO DE LIQUIDACIÓN DE SOLICITUDES DE CRÉDITO.....	99
6.2	INCREMENTO DE LA CANTIDAD DE OPERACIONES DE CRÉDITO DIARIAS .....	102
6.3	COSTO DE IMPLEMENTACIÓN INFERIOR AL DE LA SOLUCIÓN PROPIETARIA .....	104

## ÍNDICE DE FIGURAS

Figura 1.1. Diagrama de la solución propuesta .....	8
Figura 2.1. Arquitectura Referencial de OnBase.....	27
Figura 2.2. Arquitectura referencial de Alfresco .....	31
Figura 2.3. Contenedores Java JEE .....	36
Figura 2.4. Tecnologías Java EE utilizadas en la solución .....	37
Figura 2.5. Tecnologías Java EE en el Contenedor Web .....	37
Figura 2.6. Perspectiva técnica de Java Web Start .....	42
Figura 2.7. Ejemplo de un archivo JNLP.....	44
Figura 2.8. Ejemplo de una clase expuesta como servicio web.....	49
Figura 2.9. Contrato WSDL generado automáticamente .....	50
Figura 2.10. Comparativa entre el envío XML Base64 y el uso de MTOM ...	53
Figura 4.1. Integración entre Gestor Documental, SCS y OnBase .....	73
Figura 4.2. Comparación T.E.P. servicio en línea y mini batch.....	78
Figura 4.3. VE – Casos de Uso de Gestor Documental.....	80
Figura 4.4. VL - Clases principales del cliente de Gestor Documental .....	81
Figura 4.5. VL - Clases de dominio del servicio web de Gestor Documental	82
Figura 4.6. VL - Clases principales del servicio web de Gestor Documental	83
Figura 4.7. VL - Clases principales del sitio web de Gestor Documental.....	85
Figura 4.8. VD - Componentes del sistema Gestor Documental.....	86
Figura 4.9. VP - Búsqueda de documentos requeridos e indexados .....	87
Figura 4.10. VP - Clasificación de documentos .....	87
Figura 4.11. VP - Carga de documentos clasificados .....	88

Figura 4.12. VP - Indexación de Documentos a OnBase.....	88
Figura 4.13. VF - Despliegue del sistema Gestor Documental .....	89
Figura 6.1. Tiempo promedio de digitalización de documentos .....	101
Figura 6.2. Promedio diario de expedientes ingresados por mes .....	102
Figura 6.3. Porcentaje de expedientes liquidados por mes .....	103
Figura 6.4. Ecosistema de clientes de Gestor Documental .....	106

## ÍNDICE DE TABLAS

Tabla 1. Características de la solución propuesta .....	7
Tabla 2. Principales componentes de OnBase instalados .....	28
Tabla 3. Principales componentes de Alfresco .....	32
Tabla 4. Comparativa técnica entre OnBase y Alfresco [10], [11].....	33
Tabla 5. Principales clases e interfaces del API de JAI .....	39
Tabla 6. Principales elementos y atributos de un contrato JNLP .....	45
Tabla 7. Anotaciones JAX-WS más utilizadas .....	51
Tabla 8. Integraciones de Gestor Documental.....	72
Tabla 9. Tiempo de respuesta promedio del servicio de carga en línea .....	77
Tabla 10. T.E.P. Servicio de carga mini batch .....	78
Tabla 11. Plan de Pruebas del aplicativo Gestor Documental .....	93
Tabla 12. Resultados del Plan de Pruebas de Gestor Documental .....	96
Tabla 13. Comparación tiempos liquidación de solicitudes de crédito .....	100
Tabla 14. Tiempo promedio de digitalización de documentos. ....	101
Tabla 15. Estimación de costo de licencias OnBase .....	105
Tabla 16. Estimación de costo de desarrollo de Gestor Documental.....	105
Tabla 17. Comparación costos OnBase y Gestor Documental.....	108

## INTRODUCCIÓN

Las regulaciones para las empresas financieras a nivel mundial son cada vez más y en el caso de Ecuador, esto no es la excepción. Iniciativas para la detección de fraudes y la prevención de lavado de activos obligan a que los procesos de negocio deban ser estudiados y certificados constantemente por auditorías internas y externas a la entidad. Estas regulaciones exhortan a que cada producto o servicio contratado por el cliente cuente con el debido respaldo documental de cada interacción entre la empresa y el cliente.

Este tipo de información no estructurada ingresa a la entidad en grandes volúmenes, debe ser revisada, aprobada, almacenada y custodiada de forma física en la mayoría de los casos. Esto no solo acarrea problemas de seguridad sino también un alto costo en los tiempos de respuesta hacia los clientes y por otra parte, una huella ecológica que se podría evitar.

Por tales motivos, la utilización de herramientas tecnológicas que permitan su digitalización y manipulación de una forma ágil y segura, se vuelve un punto de suma importancia para entidades financieras reguladas constantemente por organismos públicos y privados.

A continuación, abordaremos el caso específico de una entidad financiera cuyos procesos de contratación de servicios requirieron de un rediseño para poder agilizar tiempos de respuesta con los clientes y así mismo registrar y manipular la información documental de cada uno de ellos apoyándose en tecnología gratuita.

## **CAPÍTULO 1**

### **GENERALIDADES**

#### **1.1 ANTECEDENTES**

La entidad financiera objetivo de este trabajo de tesis lleva 90 años ofreciendo servicios financieros de calidad a la población ecuatoriana. Esta entidad en la actualidad posee una cartera de 1'950.000 clientes y una red de agencias e islas de atención a nivel nacional.

Esta empresa posee una infraestructura tecnológica madura. Su ecosistema de aplicaciones comprende: un core de negocio principal implementado en mainframe, y por otra parte, cores de negocio secundarios y aplicaciones satélites implementados sobre tecnología Microsoft en su mayoría.

Luego de un proyecto de gran envergadura, por medio del cual se logró la actualización de la infraestructura del mainframe y del ambiente de virtualización, la institución ahora se está enfocando en proyectos para actualizar el software que soporta sus principales procesos de negocio.

## **1.2 DESCRIPCIÓN DEL PROBLEMA**

Para esta institución, las operaciones de crédito a clientes comprenden el principal mecanismo de colocación de fondos. En la actualidad se liquidan un aproximado de 1500 operaciones cada mes. Estas operaciones se llevan a cabo por los gestores comerciales asignados a cada agencia de la institución, y a pesar de ser la principal fuente de colocación de dinero para la entidad, se apoyan en un proceso que puede ser mejorado desde el punto de vista operativo y tecnológico.

La principal falencia del proceso actual es el manejo de los documentos físicos que cada cliente entrega a su asesor comercial al momento de la contratación de un crédito. Estos documentos son digitalizados en la agencia o sucursal para ser compartidos con los oficiales de la Fábrica de Crédito de la institución, para que realicen una primera inspección de los mismos. Luego de esto, los documentos físicos son envalijados y enviados a la Fábrica de Operaciones para que ellos a su vez realicen una validación de segundo nivel, vuelvan a digitalizar los documentos y los guarden en el sistema de gestión de contenido empresarial (ECMS

por sus siglas en inglés) en un formato específico. Una vez llegado a este punto la solicitud hecha por el cliente días atrás en la agencia puede iniciar su etapa de liquidación.

La Fábrica de Operaciones posee tres oficinas, en cada una se centraliza la recepción de los documentos físicos de todas las operaciones de crédito realizadas a nivel regional y se realiza una segunda digitalización e indexación de documentos, ya que la primera digitalización no fue realizada bajo los parámetros necesarios y aceptados por el ECMS. Esto provoca que la carga operativa debido a la digitalización sea muy alta y además redundante.

También se puede percibir que el tiempo transcurrido durante el viaje de la valija de documentos desde las agencias hasta la Fábrica de Operaciones suele llegar a ser de hasta 2 días. Durante todo este tiempo la operación no puede avanzar a la siguiente etapa.

El escenario para el cliente puede complicarse aún más si la Fábrica de Operaciones, durante la revisión de segundo nivel que realizan a los documentos llegara a declarar una no conformidad, sea por temas de legibilidad, caducidad, etc. debe volver a enviarlos a la agencia correspondiente para que el asesor comercial se comuniquen con el cliente y le solicite realizar la sustitución necesaria. Esto significa que el

tiempo que le tomaría liquidarse a dicha solicitud puede llegar a duplicarse y triplicarse generando una molestia para los clientes.

Como se puede apreciar, el pequeño pero crítico aspecto del proceso que tiene que ver con el manejo de los documentos físicos y digitales involucrados en la contratación de productos de crédito, tiene un impacto directo en la calidad del servicio que genera varias inconformidades y reclamos por parte de los clientes, quienes en la mayoría de ocasiones necesitan el dinero solicitado de manera urgente.

La institución ha planteado en más de una ocasión el poder implementar una solución propietaria de terceros para poder resolver estos inconvenientes y mejorar el proceso, sin embargo, se convierten en alternativas no viables, ya sea porque los costos involucrados en licenciamiento exceden el presupuesto o porque la implementación de la herramienta no cubre todos los requerimientos de la institución.

Mientras pasa el tiempo, el riesgo de quedar permanentemente con una imagen negativa no solo frente a los clientes actuales sino también frente a los nuevos, se hace se hace cada vez más latente.

### **1.3 SOLUCIÓN PROPUESTA**

Dada la naturaleza del problema, el presente documento de tesis plantea una solución basada en un desarrollo hecho a la medida, basado en

tecnologías de código abierto y gratuitas, que sirva como integrador entre un proceso estandarizado de digitalización de documentos realizado en la punta (agencias y sucursales), el sistema web de solicitudes de contratación de servicios de crédito (SCS) y el sistema de gestión de contenido empresarial.

Inicialmente, se plantea mejorar y estandarizar la digitalización de documentos en la punta. Esto se logrará instalando en cada agencia un escáner de alta velocidad, cuyas configuraciones de formato por tipo de documento se encuentren estandarizadas. Este escáner podrá ser compartido en red y permitirá a cada asesor comercial digitalizar los distintos tipos documentos del cliente, cada uno con su perfil de configuración correspondiente: documentos de identificación a colores, documentos a blanco y negro, resolución, formato TIFF, etc. El escáner guardará directamente cada documento en una ruta personal de la PC del asesor.

El aplicativo cliente del sistema propuesto se conectará al SCS y de acuerdo a la operación de contratación que el asesor esté llevando en ese momento, obtendrá la lista de los documentos que deberá solicitar al cliente. Luego de que el asesor reciba por parte del cliente éstos documentos, el aplicativo cliente se encargará de leer los documentos digitalizados previamente en su ruta personal y le brindará al asesor

comercial la interfaz gráfica necesaria para que el mismo pueda clasificar los documentos: relacionar cada imagen digitalizada con un documento solicitado por el sistema SCS. Luego de esta clasificación, el asesor procederá a terminar este proceso iniciando la indexación de cada documento clasificado hacia el sistema ECMS.

En cuestión de minutos, todos los documentos solicitados al cliente habrán sido indexados directamente en el ECMS y el sistema SCS podrá ofrecer una pantalla de consulta de documentos a las áreas de Fábrica de Crédito y Fábrica de Operaciones, permitiendo que la solicitud sea aprobada en horas y ya no en días o semanas.

La solución comprende el desarrollo de:

- Software cliente de Gestor de Documentos (Consulta, clasificación e indexación de documentos).
- Servicio web de Gestor de Documentos (Consulta e indexación de documentos).
- Página web de Gestor de Documentos (Consulta de documentos).

La solución asume la existencia previa de:

- Servicio web de SCS (Integración: Consulta de información de documentos requeridos por operación de crédito).

- Servicio web de ECMS (Integración: Consulta de documentos digitales indexados en el repositorio).
- Proceso de indexación en ECMS (Integración: Indexación de documentos nuevos).

Tabla 1. Características de la solución propuesta

<b>Arquitectura</b>	Cliente pesado / Servidor (2 Capas)
<b>Lenguaje de Desarrollo</b>	Java 1.7
<b>Ambientes de Desarrollo</b>	Netbeans 8.2, Eclipse Mars.1
<b>Plataforma de Despliegue</b>	JBoss Application Server 7.1
<b>APIs y Librerías</b>	<ul style="list-style-type: none"> <li>- Java Swing</li> <li>- Java Advanced Imaging (JAI)</li> <li>- Java Architecture for XML Binding (JAXB)</li> <li>- Java API for XML Web Services (JAX-WS)</li> <li>- Message Transmission Optimization Mechanism (MTOM)</li> </ul>

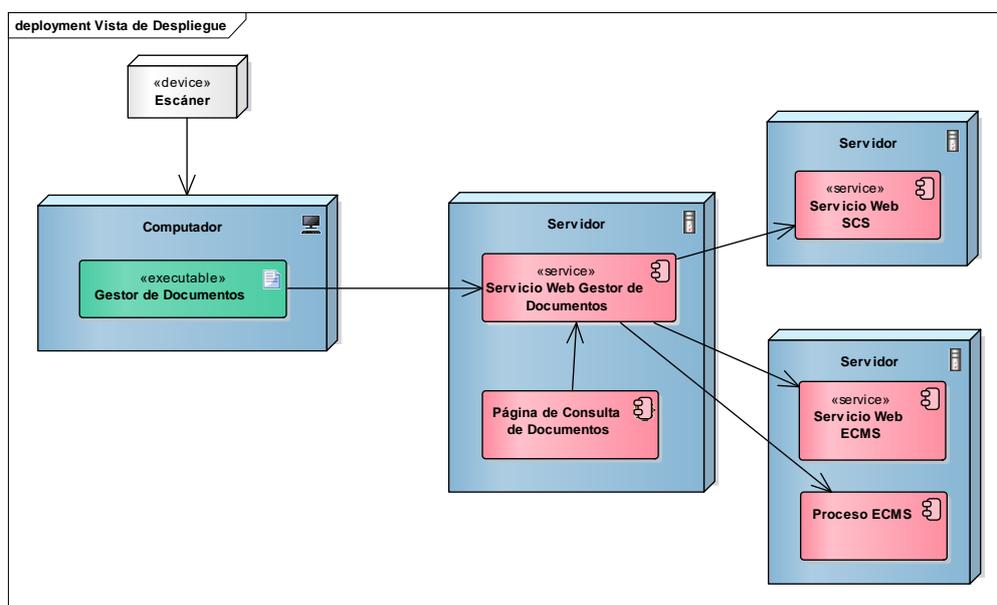


Figura 1.1. Diagrama de la solución propuesta

La solución traerá consigo los siguientes beneficios:

- Disminución de los tiempos de liquidación de solicitudes de crédito.
- Disminución de la utilización de papeles impresos durante los procesos de contratación.
- Incremento de la productividad de los oficiales de Fábrica de Operaciones y Fábrica de Crédito.
- Ahorro en licencias de software de terceros.

## **1.4 OBJETIVO GENERAL**

Disminuir la carga operativa y los tiempos involucrados en la contratación de productos de crédito en una entidad financiera implementando un sistema de indexación distribuida de documentos que permita descentralizar el almacenamiento de documentos digitales solicitados durante estos procesos.

## **1.5 OBJETIVOS ESPECÍFICOS**

- Presentar los conceptos fundamentales que rodean la digitalización, transferencia y almacenamiento de documentos, así como las tecnologías y productos que ofrece actualmente el mercado para este fin.
- Conocer las necesidades y requerimientos del área comercial de una entidad financiera en cuanto al proceso de contratación de productos de crédito.
- Diseñar una solución tecnológica escalable que permita obtener, clasificar, transferir y almacenar documentos digitales a través de protocolos web.
- Construir una solución tecnológica acorde al diseño planteado e integrada al actual entorno informático comercial y al repositorio de documentos digitales, utilizando lenguajes de programación y plataformas de despliegue de código abierto.

- Agilizar los procesos de contratación de productos de crédito disminuyendo sus tiempos de captura, transferencia y revisión documental e incrementando la productividad del área comercial de la entidad financiera.

## **1.6 METODOLOGÍA**

A continuación, se abordarán brevemente cada una de las etapas que conforman la metodología que se siguió para el desarrollo del presente trabajo de titulación. Cada una de estas etapas está muy relacionada con la anterior y todas en conjunto permitieron fundamentar y construir la solución al problema planteado.

### **1.6.1 Revisión de literatura**

Se procedió a indagar en distintas fuentes de información acerca de conceptos técnicos específicos y necesarios para fundamentar el desarrollo de la solución.

El tipo de literatura que se utilizó, corresponde principalmente a artículos publicados por la IEEE, artículos publicados por los fabricantes de las tecnologías investigadas y artículos elaborados por los *heavy users* de las mismas.

Las fuentes de información utilizadas comprenden: la librería digital IEEE Xplore, el buscador de artículos Google Scholar, el

sitio web de la Asociación de la Administración de Información e Imágenes (AIIM por sus siglas en inglés) y los sitios web oficiales de las tecnologías evaluadas tales como: OnBase, Alfresco y Java Enterprise Edition.

Los criterios de búsqueda de toda esta información se basaron en palabras clave en idioma inglés tales como: *ecm, content management system, comparison, architecture, onbase, alfresco, jee, java web start, jax-ws, java advance imaging, etc.* Los resultados obtenidos de estas búsquedas se fueron filtrando para así poder obtener información más actualizada y mejor focalizada.

Esta revisión permitió poner los cimientos de un marco teórico sobre el cual se apoyaron las principales decisiones funcionales y técnicas de la solución propuesta.

### **1.6.2 Búsqueda de herramientas o soluciones existentes**

Una vez que se comenzó a revisar la literatura correspondiente, se procedió a investigar sobre herramientas afines a la solución propuesta. Entre estas herramientas tenemos:

- Herramientas de gestión de contenido empresarial
- Herramientas de desarrollo de código abierto

- Tecnologías de código abierto para el desarrollo de aplicaciones ricas en interfaz gráfica
- Tecnologías de código abierto para el desarrollo de servicios web
- Arquitecturas referenciales, mejores prácticas y estándares de desarrollo para estas tecnologías.

Los criterios que se utilizaron para la búsqueda de éstas herramientas fueron los siguientes:

- Para las herramientas de gestión de contenido empresarial: el entorno tecnológico de la institución financiera, la madurez del producto, los costos de implementación.
- Para las herramientas de desarrollo de la solución: la madurez del producto, los costos de implementación, casos de éxito y la calidad de la documentación existente.

### **1.6.3 Comparación de herramientas o soluciones existentes**

Las principales características funcionales y técnicas de cada una de las herramientas revisadas fueron descritas y tabuladas.

Los criterios técnicos que se abordaron para la selección son los siguientes: sistemas operativos soportados, servidores de base

de datos soportados, servidores de aplicaciones soportados, tecnologías utilizadas, etc. Cada uno de estos criterios fue evaluado con una ponderación equitativa. Este proceso permitió comparar sus ventajas y desventajas, para posteriormente emitir conclusiones acerca de los mejores casos prácticos para cada una de estas herramientas.

#### **1.6.4 Personalización de herramientas**

De acuerdo a las características que poseía la herramienta seleccionada, se diseñó la solución propuesta. Este diseño contempló los aspectos técnicos que se debieron abordar para lograr un desarrollo de alta calidad y una integración fluida con la herramienta. La premisa que se buscó cumplir es la de utilizar tecnologías cuyo licenciamiento no implique un costo para su despliegue en un ambiente corporativo.

#### **1.6.5 Pruebas realizadas**

Las pruebas de la solución se realizaron en las siguientes instancias del proyecto:

- Pruebas alfa: Realizadas durante la construcción de cada componente de software y al finalizar la integración de los mismos.

- Pruebas beta: Realizadas al finalizar la integración de la solución propuesta con el sistema SCS y el sistema ECMS en ambiente de Desarrollo.
- Pruebas gama: Realizadas por el usuario luego de la implementación de la solución propuesta en ambiente de Certificación.

## **CAPÍTULO 2**

### **MARCO TEÓRICO**

#### **2.1 CONCEPTOS GENERALES DE LA DIGITALIZACIÓN DE IMÁGENES**

Desde sus inicios, los sistemas de información se han encargado de manipular y procesar datos estructurados, es decir, datos que pueden ser tabulados, ordenados e interpretados fácilmente por un computador. La mayoría de éstos sistemas de información se apoyan en bases de datos relacionales y brindan el soporte respectivo para los procesos de negocio de una empresa.

Sin embargo, existe otro tipo de información que se genera a partir de los mismos procesos de negocio, esta información no está estructurada y por lo general nace de una forma análoga. Esta información no es

fácilmente procesable por un computador y se requiere de un ser humano para su interpretación, por ejemplo: documentos PDF, Word, PowerPoint, etc.

*“Tras la promulgación de la Sarbanes-Oxley Act en 2002, las acciones y tareas necesarias para controlar y auditar la información corporativa de las organizaciones resultaron ineludibles. A ello había que añadir una tendencia creciente a desarrollar programas y proyectos de gestión del conocimiento dentro de las organizaciones, por diferentes causas, en los cuales la gestión de información con herramientas informáticas resultaba imprescindible.”* (Jesús Tramullas) [1]

La manipulación y organización de este tipo de información requiere un gran esfuerzo tecnológico y además necesita de herramientas que brinden las facilidades necesarias para las tareas más comunes con este tipo de información: guardado, consultas, edición, etc. Para solventar y cubrir esta necesidad, en la actualidad existen los denominados Sistemas de Gestión de Contenido Empresarial (ECMS por sus siglas inglés: Enterprise Content Management System). [2]

Un ECMS está compuesto por estrategias, métodos y herramientas que ayudan en la captura, gestión, almacenamiento, preservación y entrega de contenido a través de todo el ciclo de vida de los procesos de negocio de una organización. Este contenido comprende archivos binarios que

tienen una representación visual y son de fácil comprensión para un ser humano: imágenes, documentos, etc.

Una implementación exitosa de un sistema ECMS trae consigo beneficios en cuanto al manejo de documentos y meta-información de los mismos, tales como: respaldo del contenido crítico, reducción de tiempos de búsqueda, reducción en los tiempos de trabajo y traslado, etc. [3]

A continuación, se revisan los conceptos que bordean a un sistema ECMS y a sus operaciones cotidianas.

### **2.1.1 Contenido y Documentos**

Se refiere a todo tipo de contenido no estructurado, el mismo que puede originarse dentro o fuera de la organización y posee un ciclo de vida definido. Por ejemplo: correo electrónico, documentos de texto, hojas de cálculo, formularios electrónicos, documentos físicos, etc.

### **2.1.2 Escaneo**

Es la forma mediante la cual los documentos físicos ingresan a la empresa. En una operación de escaneo centralizada, se trabaja con lotes de una gran cantidad de documentos; estos lotes son procesados por dispositivos de mucha capacidad y tamaño. Cuando las operaciones de escaneo son distribuidas, se

trabaja con menor cantidad de documentos y dispositivos de menor capacidad alojados en distintos puntos.

### **2.1.3 Categorización**

El manejo de una categorización o taxonomía permite la organización de la información documental en una jerarquía definida, propia para la empresa. Esta práctica es crucial para el correcto almacenamiento de los documentos y puede realizarse de forma automática, con apoyo de una herramienta tecnológica, o de forma operativa.

### **2.1.4 Indexación**

Este proceso se suele realizar durante la captura del documento y se refiere a la obtención de los meta-datos del mismo, por ejemplo: el número de operación de una solicitud de crédito, el tipo de crédito que se está solicitando, etc. Estos meta-datos podrán ser almacenados como *keywords* (palabras clave) junto con el documento y van a permitir su futura búsqueda y ubicación.

### **2.1.5 Repositorios**

Es la pieza más importante de los sistemas ECMS. Es el lugar en donde se guardarán los documentos una vez que hayan sido

cargados, y de dónde serán leídos cuando los mismos sean consultados. Este aspecto es al cual se dedica la mayor inversión de recursos. Dependiendo del producto ECMS, este repositorio puede ser muy sofisticado a nivel de infraestructura o un sencillo sistema de archivos.

#### **2.1.6 Almacenamiento**

Se refiere al medio físico en donde residirán los repositorios de un sistema ECMS. Pueden ser: discos de estado sólido, discos ópticos, cinta magnética, etc.

#### **2.1.7 Búsqueda y obtención**

Una de las características más importantes de los sistemas ECMS es la de poder buscar contenido en base a su categorización y palabras clave, permitiendo el acceso rápido y confiable a los documentos que han sido guardados dentro del repositorio.

“La eficacia, la eficiencia, el cumplimiento y la continuidad se combinan, en distintas proporciones, para conducir el caso de negocio para la gestión de contenidos en la mayoría de las organizaciones.” (AIIM) [4]

Una organización que se ha propuesto manejar procesos ágiles y eficientes debe contar con una gestión de documentos rápida y

consistente, esto permitirá que el contenido sea utilizado para ayudar a tomar acciones y decisiones.

## **2.2 SISTEMAS DE GESTIÓN DE CONTENIDO EMPRESARIAL**

En el mercado existen muchas soluciones para ECM, cada una de estas soluciones responde a una tipología de necesidad específica en lo que se refiere a gestión de contenidos. Por ejemplo, herramientas tecnológicas muy conocidas como Joomla y Drupal se enfocan en la gestión de contenido web y su ámbito es el de la publicación de información como artículos y noticias. [5]

Sin embargo, el enfoque que se adoptará desde ahora en adelante será el de las herramientas de gestión de contenido documental. Es decir, herramientas tecnológicas que permiten la captura, modificación, clasificación, validación, almacenamiento y depuración de documentos digitales. En este enfoque, el abanico de posibilidades se reduce y encontramos opciones como: Alfresco, Microsoft SharePoint y Hyland OnBase.

Al momento del inicio del proyecto, la entidad financiera ya cuenta con una solución ECMS implementada desde hace algunos años. Dado que la naturaleza del proyecto no es la de implementar una nueva herramienta ECMS sino la de integrar la existente con otros sistemas y procesos, se procederá a revisar de forma breve la herramienta llamada

OnBase de la empresa Hyland. Posteriormente se presentarán algunas características de su contraparte Open Source, dejando así planteada la posibilidad de una profundización en algún trabajo posterior ajeno a este trabajo de titulación.

### **2.2.1 Hyland OnBase**

De acuerdo a su página web oficial, la empresa Hyland:

*“...tiene como objetivo ofrecer a las empresas las herramientas y conocimientos necesarios para que desarrollen sus actividades sin documentos de papel físico, para mayor eficacia y productividad.”* (Hyland) [6]

El producto estrella de esta compañía es la solución tecnológica llamada OnBase. Ésta solución es una aplicación para la gestión de contenidos empresariales potente y flexible en cuanto a configuración. OnBase posee casi un centenar de módulos que pueden ser acoplados para incrementar su productividad y prestaciones. Actualmente está implementado en alrededor de 12000 instituciones alrededor del mundo.

## **Características**

A continuación, se presentan las principales características de este ECMS organizadas en los aspectos más relevantes en los que se desenvuelve la herramienta:

### Captura de documentos e información

Este aspecto comprende algunas de las operaciones más importantes de OnBase. La captura de información es la puerta de entrada del contenido hacia el repositorio documental que utiliza y ofrece alternativas que van desde la digitalización de documentos físicos hasta la importación de archivos digitales de distintos tipos: CAD, DOCX, PPTX, XLSX, PDF, etc.

Durante la captura de los documentos, la herramienta tiene la capacidad de extraer información relevante de cada uno de ellos por medio de tecnologías OCR, agilizando así los procesos de ingreso de información manual.

### Administración de documentos y procesos

Esta herramienta permite manejar una categorización jerárquica en forma de carpetas digitales para los tipos de documentos que se definan de acuerdo al modelo de negocio de la empresa, independientemente del formato del documento en sí. OnBase

también posee módulos para la administración de flujos de procesos, permitiendo la automatización de excepciones y aprobaciones, facilitando la toma de decisiones en los mismos.

#### Acceso a la información

OnBase ofrece y controla el acceso a su información, documentos y procesos utilizando varios mecanismos que se complementan entre sí: su aplicación cliente de escritorio, su módulo web, su app móvil, y además plugins que permiten la integración con herramientas de terceros, por ejemplo, Microsoft Office. Todas estas interfaces permiten que los usuarios siempre tengan acceso en línea a los repositorios del ECMS, logrando así agilidad y optimización de tiempos en los procesos de negocio de la compañía.

#### Integración con otras aplicaciones

Cuando nos referimos a la integración, nos referimos a la capacidad que posee la herramienta para compartir su información y funcionalidad con aplicaciones de terceros. Estas aplicaciones pueden ser adquiridas por la empresa o desarrolladas en casa.

Básicamente, OnBase ofrece integración en 3 niveles o capas:

- A nivel de plugins, lo que permite establecer una conexión directa entre OnBase y aplicaciones como Microsoft Office, PeopleSoft, SAP, etc.
- A nivel de pantalla, por medio de librerías que pueden ser importadas en las aplicaciones que se desarrollan dentro de casa y permiten levantar controles de usuario e interfaces gráficas que se conectan a OnBase para realizar las tareas necesarias.
- A nivel de datos, por medio de su servidor de integración empresarial (EIS por sus siglas en inglés: Enterprise Integration Server) que coordina el intercambio de información instantáneo entre los back-ends de los sistemas de información que se requerían.

### Análisis

OnBase también cuenta con funcionalidades específicas que le brindan al usuario la facilidad de poder conocer en tiempo real y de manera histórica el estado de los procesos, la integridad de la información y el correcto desempeño de la misma. Por medio de una interfaz web, OnBase brinda información detallada de los procesos empresariales que administra. También posee una

aplicación web dedicada para la auditoría del sistema y de los cambios generados sobre el mismo. Todo esto se complementa con una batería de reportes diseñados para el usuario final, dedicados a conocer los resultados de ejecución de los procesos y del sistema en general.

### Almacenamiento

Es sin duda alguna, una característica clave de esta plataforma. La administración del almacenamiento de todo el contenido que maneja cubre los aspectos de: seguridad, protección y retención. La seguridad para almacenar toda su información bajo fuertes algoritmos de cifrado, permite que la autenticación y autorización de acceso hacia el contenido pueda ser delegada a un Directorio Activo LDAP. La protección de los datos permite implementar esquemas de alta disponibilidad asegurando así, la continuidad del negocio. Y finalmente, la retención del contenido permite que, por cualquier regla o política definida, éste deba ser eliminado de forma segura.

*“Scanning paper documents and importing existing electronic information (email, PDF, etc.) is important, but it is really only a first step towards getting the most business value from capture.”*

(Nick Geddes) [7]

En este punto, cabe indicar que todas las características aquí mencionadas acerca de la plataforma de OnBase corresponden a su última versión. En la entidad financiera, objeto de este estudio, la versión implementada no es la última, y ciertas características como, por ejemplo: el servidor de integración empresarial, no se encuentran implementadas en el banco.

Una vez que estuvo definido el problema descrito en este documento (ver sección 1.2), se optó por evaluar la actualización e implementación de la nueva versión de OnBase. Ésta ofrecía una integración mucho más sutil con aplicaciones de terceros y contaba con clientes de escritorio mucho más funcionales, que en conjunto ofrecían ser la mejor solución al problema. Finalmente, esto no se pudo materializar ya que la inversión en esfuerzo y recursos que se necesitaba para realizar dicha actualización resultaban en una inviabilidad para el proyecto.

Por tal motivo, y de acuerdo a lo que se revisará en los capítulos 3 y 4, se optó por diseñar y desarrollar la aplicación dentro de casa e integrarla a OnBase utilizando las interfaces que ofrece la versión que actualmente está implementada en el banco. A pesar de no tener la misma flexibilidad que el servidor de integración

empresarial, la aplicación cumplió a cabalidad con la funcionalidad requerida por la solución propuesta.

### Arquitectura referencial de OnBase

La Figura 2.1 muestra la arquitectura de despliegue que posee actualmente esta herramienta en el banco.

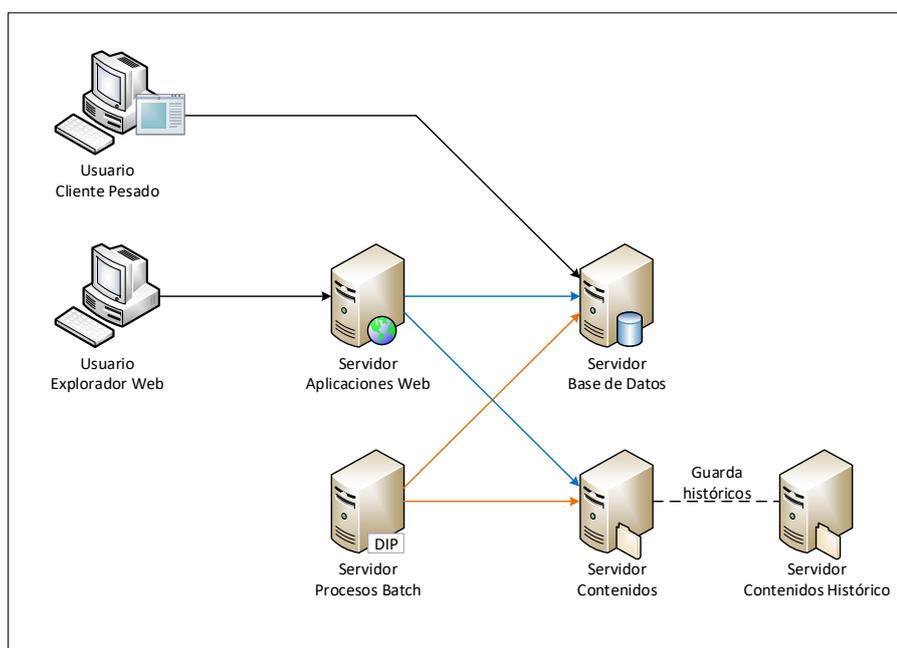


Figura 2.1. Arquitectura Referencial de OnBase

A continuación, una breve descripción de cada uno de los componentes mencionados en el diagrama presentado anteriormente.

Tabla 2. Principales componentes de OnBase instalados

<b>Componente</b>	<b>Descripción</b>
Servidor de Base de Datos	En este servidor reside la base de datos de la plataforma de OnBase. Aquí se almacena toda la información de la plataforma, configuración de tareas calendarizadas, procesos de negocio, etc. Uno de los datos más importantes que se almacena aquí, es la ruta física que posee cada documento en el sistema de archivos del Repositorio de Documentos.
Servidor de Contenidos	Este servidor funciona como un sistema de archivos dividido lógicamente en unidades llamadas DiskGroups. Aquí se almacena físicamente cada documento cargado en el ECMS y la ruta que adquiere se guarda en el registro correspondiente de la Base de Datos.
Servidor de Contenidos Histórico	Cuando los DiskGroups cumplen su tiempo de retención, se realiza un proceso de respaldo y depuración. Este servidor aloja los DiskGroups respaldados con la finalidad de que el desempeño de las búsquedas de documentos relevantes no se vea afectado por la cantidad de información.
Servidores de Aplicaciones Web	Este servidor aloja dos componentes web. Una aplicación web llamada "OnBase Web" que ofrece una interfaz liviana para administrar y consultar el contenido del repositorio, realizar búsquedas, etc. Y por otra parte, un servicio web llamado "Hyland Services", que se encarga de ofrecer un API y funciona como interfaz para la integración de otras aplicaciones con el ECMS. Mediante este API se puede realizar cualquier tipo de operación sobre el contenido del ECMS: carga, consulta, modificaciones, etc.

Componente	Descripción
Servidor de Procesos Batch	Este servidor sirve como ambiente de ejecución de los procesos batch del ECMS. La gran mayoría de estos son procesos DIP (Document Import Procesor, o Procesador de Importación de Documentos por sus siglas en inglés.) Estos procesos son los encargados de realizar cargas de contenido masivas hacia el ECMS.

Como se puede observar, OnBase de Hyland es una de las herramientas más completas del mercado de software propietario. Sin embargo, el mercado de software de código abierto, ofrecen también algunas alternativas interesantes. Tal vez éstas no posean toda la cantidad de herramientas extras y plugins que posee su competencia, pero eso no quiere decir que no sean útiles, robustas y funcionales.

### 2.2.2 Alfresco

De acuerdo a su página web oficial, el producto de Alfresco:

*“...es una potente plataforma abierta de ECM que es fácil de ampliar, personalizar e integrar en sus aplicaciones y procesos existentes. La plataforma ofrece un acceso móvil sencillo al contenido, permite una colaboración entre usuarios simple pero*

*Íntegra, y ayuda a los clientes a maximizar el valor de su contenido.” (Alfresco) [8]*

A pesar de que el proyecto nace como un proyecto de código abierto y en la actualidad se puede contar con una versión comunitaria gratuita y completamente funcional, existe también una versión de código cerrado, con su contrato de soporte y valor de licenciamiento correspondiente. Como es de esperarse, esta última es la recomendada para ambientes productivos en las empresas.

### **Características**

En breves rasgos podemos decir que Alfresco ofrece las mismas funcionalidades que su competencia más próxima, OnBase. Sin embargo, a continuación, se nombran algunas diferencias funcionales mínimas.

Los métodos de captura de contenido de Alfresco soportan una gran cantidad de formatos de archivos. Aunque estos archivos pueden ser capturados a través de su interfaz web, esta plataforma no ofrece ningún cliente pesado que pueda comunicarse directamente con dispositivos externos como un escáner y permitir la captura directamente desde el dispositivo.

Las características referentes a la administración de documentos y procesos son muy parecidas a las que ofrece OnBase, y de la misma forma ocurre con las facilidades que brinda en cuanto al acceso de la información a través de plugins que permiten interactuar con el repositorio desde aplicaciones como MS Office, Outlook, SAP, etc. [9]

### Arquitectura referencial de Alfresco

La Figura 2.2 muestra la arquitectura referencial que Alfresco sugiere para la implementación de su solución. Posteriormente, en la Tabla 3 se describen cada uno de estos componentes.

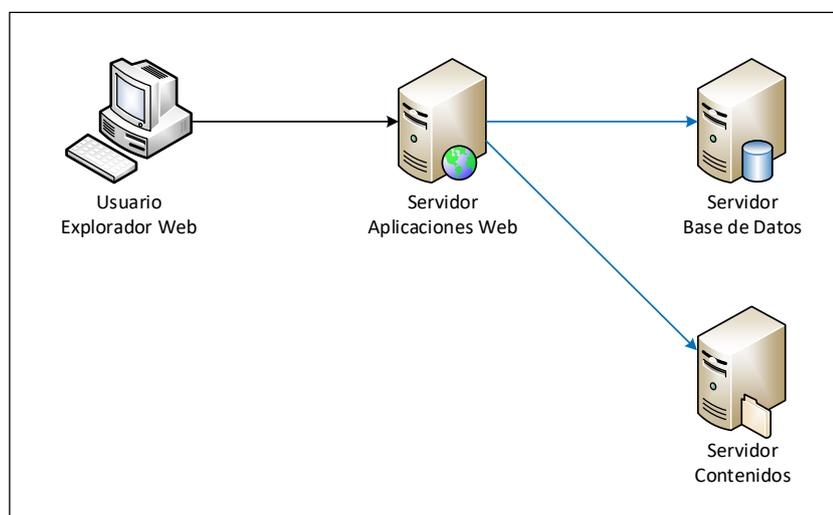


Figura 2.2. Arquitectura referencial de Alfresco

Tabla 3. Principales componentes de Alfresco

<b>Componente</b>	<b>Descripción</b>
Servidor de Base de Datos	En este servidor reside la base de datos de la plataforma de Alfresco. Aquí se almacena toda la información de la plataforma, configuración de tareas calendarizadas, procesos de negocio, etc. Los únicos datos que no se almacenan aquí son el contenido y los índices del contenido.
Servidor de Contenidos	Este servidor funciona como un sistema de archivos. Aquí se almacena físicamente cada documento cargado en el ECMS así como los índices de este contenido. Éstos índices serán utilizados para agilizar las búsquedas de los documentos posteriormente.
Servidor de Aplicaciones Web	Este servidor aloja dos componentes web. Una aplicación web llamada "Alfresco" que ofrece una interfaz muy completa para administrar y consultar el contenido del repositorio, además de ofrecer el API de integración con el repositorio. Y por otra parte, otro sitio web llamado "Share", que se encarga de ofrecer un entorno colaborativo para la gestión de contenido.

### 2.2.3 Breve comparativa técnica: OnBase vs. Alfresco

Si bien a nivel funcional ambas plataformas son muy similares, sus mayores diferencias se encuentran en el aspecto técnico. A continuación, listamos una comparación de las principales características técnicas que en determinado momento podrían

ayudar a tomar una decisión y elegir entre una u otra herramienta.

Tabla 4. Comparativa técnica entre OnBase y Alfresco [10], [11]

<b>Característica</b>	<b>OnBase</b>	<b>Alfresco</b>
Sistemas Operativos soportados	Microsoft Windows	Linux, MacOS, Solaris, Microsoft Windows, etc.
Servidores de Base de Datos soportados	MS SQL Server	MySQL, MS SQL Server, Oracle, PostGre SQL, etc.
Servidores de Aplicaciones soportados	Internet Information Services	Tomcat, JBoss AS, Oracle Weblogic, IBM WebSphere, etc.
Principales tecnologías utilizadas	.Net Framework	JEE Platform, Spring Framework
Plataformas móviles	Android, Blackberry, iOS, Mobile Web, Windows Phone.	Android, iOS, Mobile Web.

De acuerdo a lo que se puede observar, Alfresco se puede situar como una excelente alternativa a OnBase, especialmente cuando:

- Se trata de implementaciones nuevas: Una vez que un sistema ECMS ha sido implementado en una empresa y lleva mucho tiempo de madurez, es una tarea muy compleja el poder migrar hacia otro producto.

- Se trata de ahorrar presupuesto en licencias: Todo el software que funciona como plataforma para Alfresco está compuesto por herramientas comunitarias.

### **2.3 TECNOLOGÍAS UTILIZADAS EN LA IMPLEMENTACIÓN DE LA SOLUCIÓN**

Tal como se ha mencionado anteriormente, la solución propuesta para resolver el problema no se basa en la migración de producto ECMS ni en la actualización del mismo. Sino que, se optó por diseñar construir una herramienta específica, que funcione como un front-end para el usuario y que el back-end de este aplicativo se integre tanto al producto ECMS como al sistema de solicitudes de contratación de productos financieros.

La tecnología que se escogió para desarrollar esta herramienta es Java, cuyas principales características revisaremos a continuación.

Java posee una versión estándar que se denomina JSE (Java Standard Edition) y ésta se distribuye en dos tipos de instaladores: un ambiente de ejecución (JRE) y un ambiente de desarrollo (JDK).

El JRE (Java Runtime Environment) contiene todos los programas necesarios para ejecutar programas Java. Y el JDK (Java Development Kit) contiene todos los programas necesarios para compilar y ejecutar

programas Java; además de ciertas herramientas de diagnóstico muy útiles para el rastreo de errores y demás.

El JSE posee todo el conjunto de clases necesarias para crear y ejecutar programas cubriendo la mayoría de escenarios funcionales para un aplicativo de escritorio. Para que una aplicación JSE pueda ser ejecutada, únicamente necesita de un ambiente de ejecución JRE.

### **2.3.1 Java Enterprise Edition**

Así como existe una versión estándar de Java, también existe otra versión llamada JEE (Java Enterprise Edition). La misma, extiende el universo de clases que se pueden utilizar y cubren necesidades que generalmente se dan en aplicaciones de escala empresarial: exposición de servicios web, construcción de aplicaciones web, mapeos ORM, protocolos de mensajería, etc.

Para que una aplicación JEE pueda ser ejecutada, necesita de los siguientes requisitos:

- Una JVM (Java Virtual Machine, o Máquina Virtual de Java por sus siglas en inglés), incluida en el JDK.
- Un programa compilador Java, incluido en el JDK. Esto debido a que las aplicaciones de Java realizan compilaciones en tiempo de ejecución.

- El contenedor de aplicaciones JEE correspondiente Por lo general, este es el rol que asumen plataformas como Apache Tomcat, Red Hat JBoss Application Server, Oracle WebLogic Server, etc.

A continuación, la Figura 2.3 describe una arquitectura básica de una aplicación JEE. Así mismo, la Figura 2.5 describe el respectivo listado de las tecnologías que se comprenden en el Contenedor Web de aplicaciones JEE, ya que este último es el que principalmente utilizaremos en el desarrollo de nuestra plataforma.

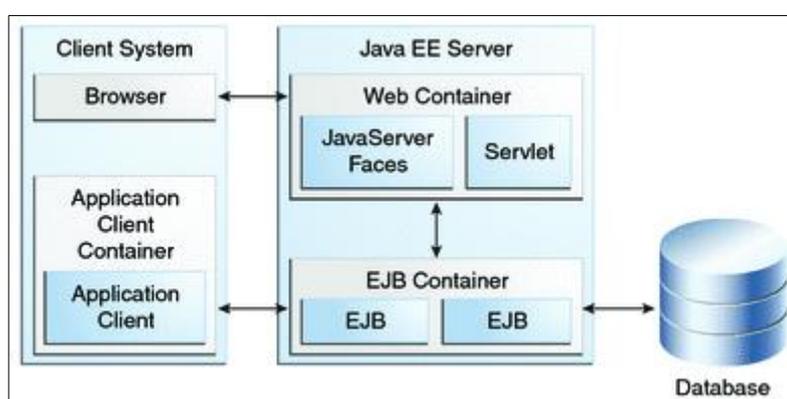


Figura 2.3. Contenedores Java JEE  
Referencia: Oracle, The Java EE 6 Tutorial [12]

De todas éstas tecnologías, procuraremos centrarnos en las siguientes: Servlet, JSP, JAX-WS, y además algunas tecnologías que serán necesarias en el lado del cliente como son: JAI y JWS. La Figura 2.4 muestra un diagrama que representa, de forma

conceptual, la interacción entre estas distintas tecnologías que será necesario implementar para el desarrollo de la solución propuesta.

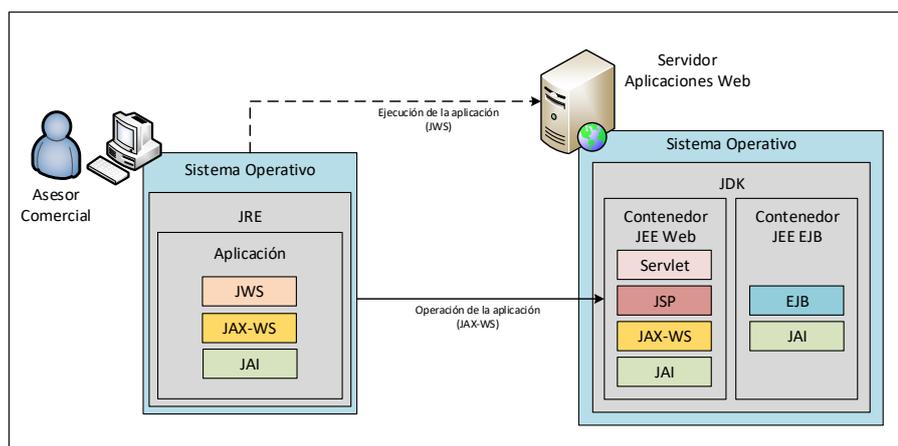


Figura 2.4. Tecnologías Java EE utilizadas en la solución

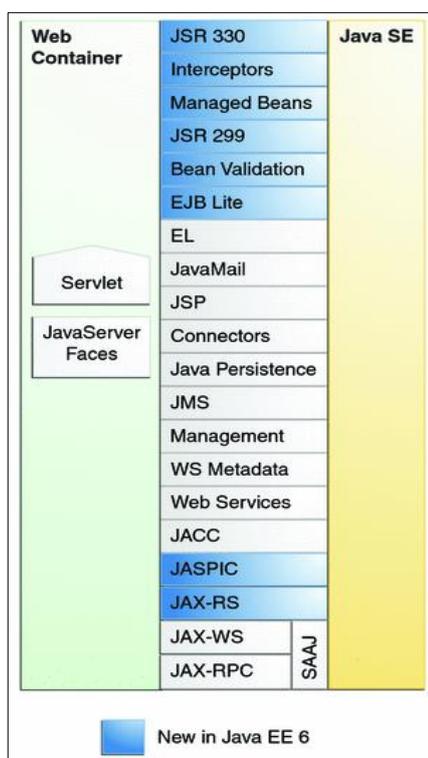


Figura 2.5. Tecnologías Java EE en el Contenedor Web  
Referencia: Oracle, The Java EE 6 Tutorial [12]

A continuación, hablaremos brevemente de las principales tecnologías y APIs mencionadas en la Figura 2.4.

### **2.3.2 Java Advanced Imaging (JAI)**

El API del JDK trae consigo las clases necesarias para realizar operaciones sobre imágenes. Estas clases ofrecen una interfaz para poder abrir archivos de imágenes, realizar cambios de resolución, codificación a otros formatos, decodificación, interpretación, etc. Sin embargo, la cantidad de formatos con los que se puede trabajar está limitado a imágenes BMP, PNG y JPEG.

Uno de los formatos con los que no puede trabajar de forma nativa el API del JDK, es el formato TIFF. Este formato es un tanto más complejo debido a que puede llegar a funcionar como un contenedor de varias imágenes, las mismas que se presentan como páginas de un solo documento. Otra característica de los archivos TIFF es que permiten que las páginas que contiene puedan estar codificadas en un formato diferente, aprovechando así -por ejemplo- las capacidades de compresión de los métodos JPEG y G4-CCITT para imágenes a color y blanco y negro respectivamente.

Esto quiere decir que pueden llegar a existir un archivo en formato TIFF que contenga varias páginas a BN y que cada página esté compresada en G4-CCITT y otro archivo TIFF que contenga varias páginas a color compresadas como JPEG. En ambos casos, el sistema operativo los verá como un solo archivo con extensión TIFF. De esta manera, podremos contar con archivos TIFF multi-página relativamente livianos cuando se traten de documentos con gran cantidad de información. En este caso es donde se torna relevante apoyarse en una librería que extienda el alcance de formatos que trae el API tradicional.

JAI (Procesamiento Avanzado de Imágenes para Java, por sus siglas en inglés) es un API que permite la manipulación avanzada de imágenes para las aplicaciones Java. JAI incorpora el soporte para varios formatos de imagen complejos, entre ellos el formato TIFF, extendiendo así las capacidades del programa.

Las clases e interfaces que serán de mayor utilidad están descritas en la tabla a continuación:

Tabla 5. Principales clases e interfaces del API de JAI

<b>Clase / Interfaz</b>	<b>Descripción</b>
ImageIO	Clase a partir de la cual se pueden localizar los correspondientes objetos que permiten hacer lecturas o escrituras de imágenes de forma

Clase / Interfaz	Descripción
	sencilla. Además permite realizar operaciones simples de codificación y decodificación de imágenes.
ImageInputStream	Es una interfaz que permite encapsular distintos tipos de flujos de lectura de imágenes. Estos flujos de lectura pueden ser desde archivos o desde memoria. A través de ella, se lee la información de una imagen.
ImageReader	Es una clase abstracta que permite realizar el análisis sintáctico y la decodificación de una imagen.
ImageOutputStream	Es una interfaz que permite encapsular distintos tipos de flujos de escritura de imágenes. Estos flujos de escritura pueden ser hacia archivos o hacia memoria. A través de ella, se escribe la información de una imagen.
ImageWriter	Es una clase abstracta que permite realizar la escritura y la codificación de una imagen.
JPEGImageWriteParam	Esta clase permite extender los parámetros de escritura de una imagen JPEG. Estos parámetros pueden ser: el modo de compresión, el tipo de compresión, y la calidad de compresión.

### 2.3.3 Java Web Start (JWS)

La distribución o aprovisionamiento de software siempre ha sido un importante aspecto a considerar, sobre todo cuando se

trabaja con aplicaciones de escritorio. Éste tipo de aplicaciones, normalmente conocidas como “clientes pesados”, deben cumplir con requisitos funcionales específicos tales como: fluidez en la interfaz de usuario, acceso al sistema de archivos local, comunicación con otros dispositivos mediante puertos USB o COM, etc. Sin embargo, el manejar la distribución inicial y las actualizaciones de versiones de este tipo de aplicativos en ambientes organizacionales donde el número de los equipos de cómputo supera el centenar de unidades, también representa un gran reto.

La tecnología de Java Web Start (Inicio Web de Java, por sus siglas en inglés) tiene como objetivo principal el cubrir esta necesidad, brindando un mecanismo para publicar, descargar y ejecutar aplicaciones Java a través de la red [13]. La Figura 6 muestra la forma de ejecución de este tipo de aplicaciones desde una perspectiva técnica:

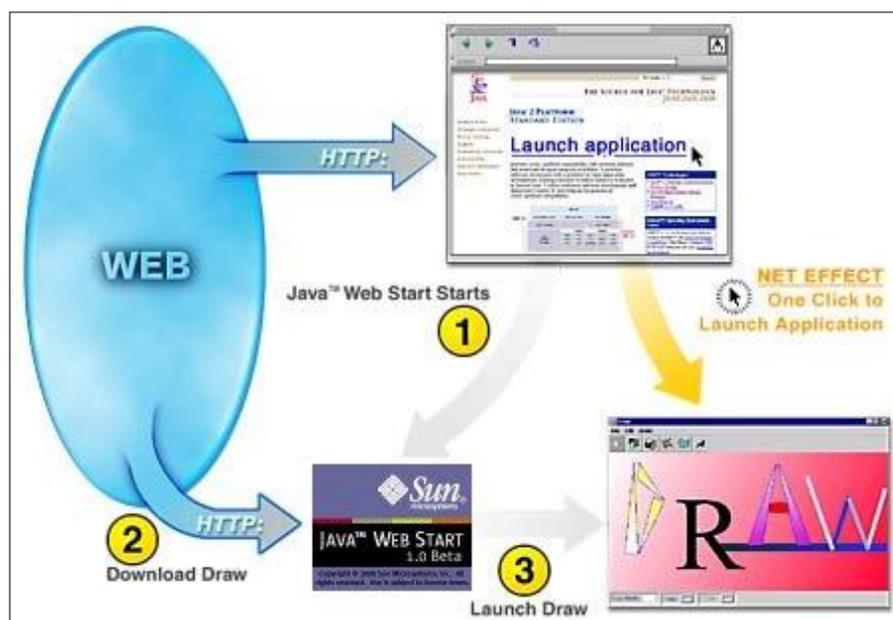


Figura 2.6. Perspectiva técnica de Java Web Start  
Referencia: Oracle, Java Web Start Architecture [14]

A breves rasgos, los pasos que describe la Figura 6 son los siguientes:

- El programa ejecutable de una aplicación de escritorio es alojado en un servidor web y es publicado a través de una URL.
- Cuando el explorador accede a esta URL, el recurso descargado es un archivo XML que funciona como un contrato de un protocolo llamado JNLP (Java Network Launching Protocol).

- El explorador delega la interpretación de este contrato a la máquina virtual de Java del cliente, quien a su vez detecta e invoca todos los recursos necesarios para la ejecución del programa tales como: ubicación del archivo JAR que contiene la clase principal del programa, ubicación de los archivos JAR que funcionan como dependencias para la ejecución del programa, parámetros de ejecución, etc.
- La JVM realiza una comprobación automática de la versión de cada uno de estos componentes (programa y dependencias) y si constata que es la primera vez que ejecuta el programa o la versión que tiene descargada no es la última, procede a descargar cada archivo.
- Cuando la JVM comprueba de que tiene el programa descargado y actualizado, procede a ejecutarlo localmente en un contexto protegido sobre el sistema operativo cliente.

Este mecanismo permite que la distribución de programas de escritorio y las futuras actualizaciones se realicen de una forma automática a través de una URL, trayendo consigo las siguientes ventajas:

- Ejecución de las aplicaciones con un solo clic.
- Garantía de la ejecución de la última versión de la aplicación.
- Eliminación de complejos procedimientos de instalación y actualización.

## Ejemplo de un archivo JNLP

A continuación, la Figura 2.7 muestra un ejemplo básico de cómo se vería un archivo JNLP.

```

1 <jnlp spec="1.0+" codebase="https://appserver/BG.JGestorDocumental/apps/qd/">
2   <information>
3     <title>Gestor Documental</title>
4     <vendor>Empresa SA</vendor>
5     <homepage href="null"/>
6     <description>Modulo de Digitalizacion e Indexacion de documentos</description>
7     <description kind="short">GestorDocumental</description>
8   </information>
9   <security>
10    <all-permissions/>
11  </security>
12  <update check="always" policy="always"/>
13  <resources>
14    <java java-vm-args="-Djavax.xml.bind.JAXBContext=com.sun.xml.internal.bind.v2.ContextFactory"
15      version="1.7+/">
16    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/BG.JGestorDocumental.Client.jar"
17      download="eager" main="true"/>
18    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/lib/commons-logging-1.1.3.jar"
19      download="eager" main="false"/>
20    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/lib/log4j-1.2.17.jar"
21      download="eager" main="false"/>
22    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/lib/jai_imageio.jar"
23      download="eager" main="false"/>
24    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/lib/pdfbox-2.0.1.jar"
25      download="eager" main="false"/>
26    <jar href="https://appserver/BG.JGestorDocumental/apps/qd/lib/jGrid-0.2.jar"
27      download="eager" main="false"/>
28    <property name="jnlp.LOGS_DIR" value="GestorDocumental\logs"/>
29    <property name="jnlp.LOGS_LEVEL" value="DEBUG"/>
30    <property name="jnlp.IMAGE_IN_DIR" value="GestorDocumental\imagenes\scan"/>
31    <property name="jnlp.IMAGE_OUT_DIR" value="GestorDocumental\imagenes\upload"/>
32  </resources>
33  <application-desc main-class="com.empresa.jgestordocumental.JDocumentMatcher"/>
34 </jnlp>

```

Figura 2.7. Ejemplo de un archivo JNLP

Cuyos elementos y atributos principales se describen a continuación:

Tabla 6. Principales elementos y atributos de un contrato JNLP

<b>Elemento</b>	<b>Atributo</b>	<b>Descripción</b>
Security		Es opcional y se lo utiliza para solicitar permisos de mayor privilegio sobre la PC del cliente.
All-permissions		Solicita el mayor privilegio de permisos para la aplicación sobre el sistema operativo. Por ejemplo el acceso al sistema de archivos.
Resources		Describe los recursos que serán utilizados por la aplicación.
Java		Describe información relevante para la máquina virtual que interpretará este contrato.
	Java-vm-args	Parámetros de la máquina virtual de Java que se añadirán a la ejecución del programa en el cliente.
	Versión	Las versiones de Java que serán necesarias para ejecutar la aplicación.
Jar		Archivo JAR necesario para la ejecución de la aplicación.

Elemento	Atributo	Descripción
	Href	URL de la ubicación del archivo JAR.
	Download	Tipo de descarga que se va a realizar del archivo JAR.
	Main	Denota si el JAR en mención posee la clase principal para la ejecución de la aplicación.
Property		Parámetros de aplicación que pueden ser enviados en el momento de la ejecución.
Application-desc		Denota que este se refiere a un archivo JNLP para una aplicación.
	Main-class	El nombre de la clase que posee el método Main para ejecutar la aplicación.

### Seguridad en un archivo JNLP

La seguridad en este mecanismo de ejecución de programas es un punto muy importante a considerar, y por tal motivo, el JRE brinda un marco de trabajo seguro que debe ser implementado cuando se utiliza esta tecnología en ambientes productivos:

- La URL del recurso JNLP compartido debe trabajar sobre el protocolo HTTPS.

- Todos los archivos JAR publicados en el archivo JNLP deben estar firmados con un certificado digital.

Esto previene que la información del programa que viaja a través de la red no pueda ser descifrada, pero mejor aún, que no se pueda implementar un esquema de *man in the middle* y ejecutar una versión modificada del programa.

#### **2.3.4 Java API for XML Web Services (JAX-WS)**

JAX-WS es una de las tecnologías más relevantes en todo el conjunto de APIs del JEE. Tal como su nombre lo indica, JAX-WS es un API para generar y consumir servicios web basados en XML. En JAX-WS, la ejecución de operaciones se realiza por medio del intercambio de mensajes por medio del protocolo SOAP, pero esto no significa que el desarrollador de la aplicación deba interactuar directamente con este protocolo, sino que cuenta con una interfaz de programación sencilla para que una clase común que esté escrita en Java pueda exponer sus métodos directamente como operaciones de un servicio web.

Esto implica que el desarrollador se puede enfocar en la lógica de negocio del sistema, y la tecnología se encargará de generar todo lo necesario en tiempo de ejecución para publicar el servicio web correspondiente [15]. Los componentes principales que

genera esta tecnología son: los contratos WSDL, los esquemas de entidades XML, los mensajes SOAP de envío y recepción, etc.

De esta forma, se puede implementar una solución web, cuya funcionalidad pueda ser aprovechada por distintos sistemas cliente y con todas las ventajas implícitas del lenguaje de programación Java; como, por ejemplo: la capacidad de poder desplegarse en casi cualquier plataforma de sistema operativo.

JAX-WS es parte de las tecnologías estándar de JEE, esto quiere decir que, para poder lograr el despliegue de un servicio web desarrollado con esta tecnología, se debe contar con un contenedor JEE. Quien asuma este rol, puede ser un producto como JBoss Application Server o WebLogic Application Server. Más adelante se revisarán las características del JBoss AS que fue el producto seleccionado para implementar esta solución.

### **Ejemplo de un servicio web basado en JAX-WS**

La Figura 2.8 muestra lo sencillo que puede resultar exponer una clase como un servicio web utilizando JAX-WS.

A simple vista, la clase TestJAXWS es una clase común y corriente, que posee atributos y métodos. Sin embargo, posee una característica especial y es que se encuentra anotada a nivel

de clase con una directiva llamada “@WebService” y a nivel de método con una directiva llamada “@WebMethod”. Éstas directivas se llaman anotaciones y funcionan para agregar metadata al código Java.



```
1 package com.tesistests;
2
3 import javax.jws.WebMethod;
4
5
6 @WebService
7 public class TestJAXWS {
8
9     @WebMethod
10    public String holaMundo() {
11        return "Hola mundo";
12    }
13
14    @WebMethod
15    public String holaUsuario(String usuario) {
16        return "Hola " + usuario;
17    }
18
19 }
```

Figura 2.8. Ejemplo de una clase expuesta como servicio web

Cada una de estas anotaciones tiene un propósito y es el de indicarle a la máquina virtual de Java que, durante el tiempo de compilación, esta clase necesita un pre procesamiento y un tratado especial. La Figura 2.9 muestra que luego de la ejecución de esta clase sobre un servidor de aplicaciones, desde el explorador ya se puede acceder al contrato WSDL que

representa el servicio web y las operaciones web que nosotros definimos en nuestra clase por medio de las anotaciones mencionadas.

```

- <wsdl:definitions name="TestJAXWSService" targetNamespace="http://tesistests.com/">
- <wsdl:types>
- <xs:schema elementFormDefault="unqualified"
  targetNamespace="http://tesistests.com/" version="1.0">
  <xs:element name="holaMundo" type="tns:holaMundo"/>
  <xs:element name="holaMundoResponse" type="tns:holaMundoResponse"/>
  <xs:element name="holaUsuario" type="tns:holaUsuario"/>
  <xs:element name="holaUsuarioResponse" type="tns:holaUsuarioResponse"/>
- <xs:complexType name="holaMundo">
  <xs:sequence/>
</xs:complexType>
- <xs:complexType name="holaMundoResponse">
- <xs:sequence>
  <xs:element minOccurs="0" name="return" type="xs:string"/>
</xs:sequence>
</xs:complexType>
- <xs:complexType name="holaUsuario">
- <xs:sequence>
  <xs:element minOccurs="0" name="arg0" type="xs:string"/>
</xs:sequence>
</xs:complexType>
- <xs:complexType name="holaUsuarioResponse">
- <xs:sequence>
  <xs:element minOccurs="0" name="return" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:schema>
</wsdl:types>
</wsdl:definitions>

```

Figura 2.9. Contrato WSDL generado automáticamente

A continuación, describiremos brevemente las principales anotaciones de JAX-WS que nos permitirán desplegar servicios web más complejos.

Tabla 7. Anotaciones JAX-WS más utilizadas

<b>Anotación</b>	<b>Ámbito</b>	<b>Descripción</b>
WebService	Clase	Indica que la clase sobre la cual está puesta esta anotación debe ser desplegada como un servicio web SOAP.
WebMethod	Método	Indica que el método sobre la cual está puesta esta anotación debe ser implementado como una operación SOAP del servicio web que la contiene.
WebParam	Parámetro	Permite redefinir el nombre que tendrá dicho parámetro en el contrato WSDL.
MTOM	Clase	Indica que los mensajes SOAP que se generen para dicho servicio web deben implementar la tecnología MTOM, la misma que será descrita en el apartado siguiente.

### 2.3.5 Message Transmission Optimization Mechanism (MTOM)

Los servicios web ayudan a que distintas aplicaciones puedan conversar un mismo idioma y puedan intercambiar información mutuamente. Los servicios web basados en SOAP y XML, por lo general, están implementados sobre el protocolo HTTP y esto hace que una de sus principales características sea la de únicamente poder transportar datos en tipo texto.

Esto no quiere decir que sea imposible enviar información binaria a través de un servicio web basado en SOAP y XML, sino que

hay que tener ciertas consideraciones al aplicar alguna de las distintas técnicas con las que se cuenta para implementar esta funcionalidad.

Uno de los primeros acercamientos que se tiene es el de utilizar una técnica llamada DIME (Direct Internet Message Encapsulation). Esta técnica dicta que la información binaria viaje de forma encapsulada en el mensaje SOAP, de forma codificada en texto. Uno de los algoritmos más utilizados para esta tarea es el Base64. De esta forma, la información binaria pasa por un proceso de codificación previo que genera una cadena de caracteres que representa la información original, y ésta cadena de caracteres viaja como parte del mensaje SOAP.

Lamentablemente, ésta no siempre es la mejor de las opciones. Una cadena codificada en Base64 siempre va a pesar 33% más que la información original [16]. Además, que hay una recarga de procesamiento previo al envío y posterior a la llegada del mensaje que debe ser asumido. Debido a esta característica, este acercamiento resulta impráctico cuando se desean transferir información binaria de gran tamaño.

Para solventar éste inconveniente se diseñó un segundo acercamiento llamado MTOM (Message Transmission

Optimization Mechanism). Éste tiene como principio que la información binaria no sea transformada y enviada dentro del mensaje SOAP, sino que viaje como un dato adjunto al mensaje. Este mecanismo permitiría que la información binaria sea dividida en porciones más pequeñas y sean transmitidas de una forma que asegure su integridad hasta su llegada al servidor. De esta forma ya no es necesario el procesamiento extra para codificación y decodificación de mensajes de gran tamaño, y así mismo el tiempo que se tomaba la transmisión de un archivo con un 33% de peso extra.

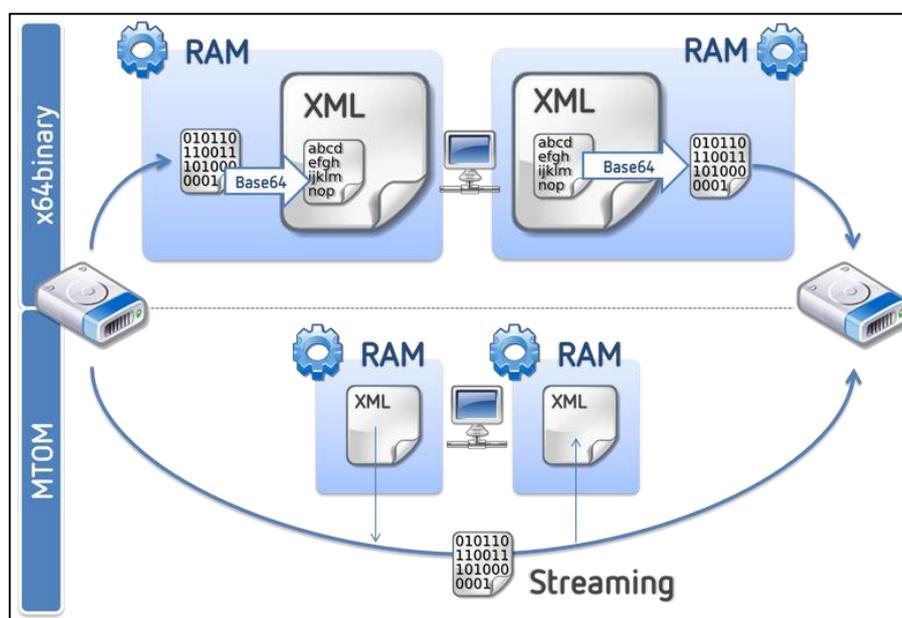


Figura 2.10. Comparativa entre el envío XML Base64 y el uso de MTOM

Referencia: Transmisión de ficheros en el esquema del Documento Electrónico [17]

Lo interesante de estar trabajando sobre la tecnología Java Enterprise Edition es que para que un servicio web pueda comportarse utilizando el mecanismo MTOM, únicamente hay que anotar la clase con la directiva “@MTOM”. Esto será suficiente para que, al momento de despliegue de la aplicación, el contrato del servicio web adopte este comportamiento y cualquier información binaria que sea enviada o consumida del servicio web viaje a través de la red utilizando las bondades de MTOM.

### **2.3.6 JBoss Application Server**

Tal como se mencionó anteriormente, para poder implementar una solución basada en Java Enterprise Edition, es necesario contar con una plataforma de despliegue de esta tecnología. En el mercado existen muchos servidores de aplicaciones que poseen entre sus principales características el ser “JEE Certified” o “JEE Compliant”. Esto quiere decir que dicho servidor de aplicaciones puede desplegar una aplicación basada en JEE.

Entre las varias opciones que existen, uno de los servidores de aplicaciones más utilizados es el JBoss Application Server. Éste nació como un proyecto de código abierto, y conforme fue madurando y creciendo en comunidad se fue consolidando como

una de las soluciones más robustas para desplegar aplicaciones JEE.

En la actualidad, a pesar de que fue comprado por RedHat y posee una versión comercial, continúa dando vida a una versión comunitaria gratuita que puede ser descargada y utilizada por cualquier implementador de soluciones Java.

Cabe mencionar que, como servidor de aplicaciones está en la capacidad de desplegar aplicaciones de varios tipos como:

- WAR: Web application Archive. Tipo de archivo que generalmente encapsula aplicaciones web y servicios web.
- EAR: Enterprise application Archive. Tipo de archivo que puede llegar a contener además de aplicaciones y servicios web, aplicaciones basadas en la tecnología EJB (Enterprise Java Beans, componentes de lógica de negocio especializados), aplicaciones basadas en mensajes y colas, aplicaciones de acceso a datos a través de mapeos ORM (Object Relational Mapping), etc.

Este servidor de aplicaciones también soporta el funcionar en un modo de clúster, compartiendo entre dos procesos diferentes

(que pueden estar en la misma PC o en PCs diferentes conectadas en red) los objetos de negocio, una memoria caché HTTP o de aplicación, etc. Lo que permitirá mantener un crecimiento horizontal para asegurar la escalabilidad y alta disponibilidad del sistema que corra sobre esta plataforma.

## **CAPÍTULO 3**

### **ANÁLISIS DE REQUERIMIENTOS**

#### **3.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL**

A continuación, se describirá la situación actual de la institución enfocados en dos aspectos fundamentales: el técnico y el procedural.

##### **3.1.1 Aspecto técnico**

###### **Sistema de gestión de contenido empresarial**

La institución financiera cuenta en la actualidad con el ECMS llamado OnBase, el mismo que lleva aproximadamente 9 años desde su implementación y 4 años desde su última actualización. Los principales usuarios de este sistema pertenecen al área de

Fábrica de Operaciones (FO). Dentro de ésta área, un grupo del personal está encargado de la recepción, clasificación, digitalización e indexación de los documentos que llegan desde las distintas agencias por motivo de operaciones de venta que se encuentran en proceso.

La plataforma de OnBase cuenta con una aplicación de escritorio llamada OnBase Client. Esta herramienta es utilizada para digitalizar, indexar y consultar documentos sobre la plataforma. La indexación de documentos a través de este cliente se puede realizar tanto en lote como de uno en uno; sin embargo, el primer acercamiento es el más utilizado cuando se trata de los procesos operativos de la FO.

OnBase también cuenta con un diseñador, configurador y motor de flujos de trabajo sencillos que se disparan a partir de la indexación de ciertos tipos de documentos. Esta funcionalidad se utiliza en los casos en que se indexan varios documentos habilitantes para una operación de crédito y los mismos llegan a las bandejas de entrada de usuarios específicos. Estos usuarios pertenecen al área de Riesgos y son quienes revisan y aprueban la validez de cada documento para que la operación pueda avanzar hacia sus etapas siguientes.

Todos los procesos de indexación masiva, son ejecutados por servicios DIP configurados en OnBase. Estos se encargan de leer archivos de texto plano que poseen información de cada documento (tipo o caracterización al que se deberá adjuntar el documento, ubicación del documento a subir y lista de keywords que se deberá agregar a cada documento).

### **Sistema de solicitudes de contratación de servicios de crédito**

El sistema web SCS es el principal entorno operativo que poseen los asesores comerciales de la institución financiera, quienes son los encargados de los procesos de venta de productos del banco, tales como: créditos de consumo, créditos para capital de trabajo, cuentas de ahorros, cuentas corrientes, pólizas, tarjetas de crédito, etc.

Este sistema tiene automatizadas el 95% de las etapas que intervienen en un proceso de venta, desde la solicitud del cliente hasta la finalización de la venta, incluyendo etapas intermedias como: validación de políticas, aprobaciones y liquidaciones de crédito.

Este sistema fue desarrollado dentro de la institución y posee una arquitectura de 3 capas descrita brevemente a continuación:

- Servidor de base de datos: Funciona con SQL Server 2008 y además de los datos, mantiene lógica de acceso a datos en procedimientos almacenados compilados dentro de la misma base.
- Servidor de aplicaciones: Ofrece servicios web escritos en C# .Net. Los mismos manejan la lógica de negocio del aplicativo y la comunicación con el core bancario y otros sistemas de soporte.
- Servidor web: Mantiene desplegado los sitios web que componen la aplicación. Estos sitios están desarrollados en ASP .Net y son la interfaz gráfica para el usuario.

### **3.1.2 Aspecto procedural**

#### **Proceso de digitalización de documentos de operaciones**

Este proceso, visto de una manera generalizada consta de los siguientes pasos:

- Los usuarios en la actualidad reciben físicamente los documentos solicitados de los clientes y proceden a ingresar la operación y la información en el sistema SCS.

- Una vez finalizada la interacción con el sistema SCS, el usuario se encarga de envalijar los documentos correspondientes a la operación para que sean enviados hacia la Fábrica de Operaciones.
- Luego de un lapso promedio de 2 días, la valija de documentos llega a la Fábrica de Operaciones y los empleados de la FO realizan la digitalización y la indexación de los documentos.

La digitalización de documentos la realizan por medio de escáneres de alta capacidad que dejan guardados los documentos digitales de cada operación en una ruta compartida. Luego, por medio del aplicativo OnBase Client, generan un lote de documentos digitales, ingresando para cada grupo de documentos su correspondiente Código de Operación. El lote inicia un flujo de trabajo dentro de OnBase, en donde:

- Se ejecuta el proceso de autollenado: Con el código de operación correspondiente se ejecuta un proceso de búsqueda de información relacionada y dicha información se guarda como keywords para el grupo de documentos correspondiente.

- Los documentos viajan hacia distintas bandejas para su revisión y aprobación.

Una vez que los documentos han sido digitalizados se archivan nuevamente y se envían para el almacenamiento de los mismos.

### **3.2 DEFINICIÓN DE LA SITUACIÓN ESPERADA**

La solución propuesta consta de un aplicativo desarrollado a la medida que permita descentralizar el proceso de digitalización de documentos llevándolo a cada agencia. El aplicativo a desarrollar se llamará Gestor Documental y deberá permitir a los asesores comerciales de cada agencia digitalizar la documentación de cada operación de crédito de una forma sencilla y rápida. Tomando y actualizando información de los aplicativos SCS y OnBase.

La implementación de esta herramienta también traerá consigo la modificación de los procesos que se llevan en la actualidad en las agencias. A continuación, se describen los pasos de cada uno de ellos, definiendo así un escenario futuro deseado.

#### **3.2.1 Proceso de digitalización en la punta**

Este proceso, visto de una manera generalizada deberá constar de los siguientes pasos:

- a) El usuario (asesor comercial) recibirá físicamente los documentos solicitados de los clientes y procederán a ingresar los datos de la operación en el sistema SCS.
- b) El usuario se dirigirá hacia la ubicación de un scanner de alta velocidad en donde procederá a escanear los documentos de la operación actual.
- c) El escáner se encargará de depositar los archivos digitalizados en una carpeta local en la PC del usuario automáticamente.
- d) Luego de que el usuario escanee los documentos, deberá invocar la aplicación Gestor Documental desde la opción Digitalizar Documentos del sistema SCS.
- e) La aplicación Gestor Documental permitirá combinar de una manera visual cada documento digitalizado con el tipo de documento correspondiente al producto y operación actual. El listado de tipos de documentos por productos será proporcionado por el sistema SCS.
- f) Cuando el usuario termine de combinar los documentos, podrá enviarlos directamente hacia OnBase desde el aplicativo Gestor Documental.

- g) Los documentos asociados serán cargados en segundo plano y el usuario podrá continuar con las siguientes etapas de la solicitud desde el aplicativo SCS.
- h) Una vez de terminada la interacción con el Gestor Documental y el sistema SCS, el usuario se encarga de envalijar los documentos correspondientes de la operación para que sean enviados hacia la Fábrica de Operaciones - Custodia. Sin embargo, ya no será necesaria la carga operativa de digitalizar e indexar estos documentos por parte de FO.

### **3.2.2 Proceso del aplicativo Gestor Documental**

Por medio de este aplicativo el usuario podrá obtener información acerca de la operación de crédito, documentos digitalizados localmente y documentos indexados en OnBase para dicha operación. La edición y eliminación de dichos documentos será parametrizada en el sistema SCS. Los pasos que seguirá el Gestor Documental son los siguientes:

- a) El aplicativo se conectará al sistema SCS y a OnBase y se encargará de buscar la siguiente información:
  - a. Información del cliente de la operación.

- b. Listado de tipos de documentos requeridos para la operación.
  - c. Listado de documentos digitalizados en la carpeta local del usuario (aquella en la que el scanner ha copiado automáticamente las imágenes).
  - d. Listado de documentos que ya han sido indexados en OnBase para esa operación (en caso de existir).
- b) El usuario procederá a escoger, por cada tipo de documento requerido, un documento del listado de documentos digitalizados, y los irá uniendo uno a uno (clasificación).
- c) El usuario podrá seleccionar cualquier imagen digitalizada o indexada para poder visualizarla en tamaño real.
- d) Una vez que el usuario haya terminado de enlazar los documentos digitalizados con su respectivo tipo de documento, deberá dar clic en el botón correspondiente para que las imágenes se transfieran a otro directorio local de salida. Éstas imágenes y su respectiva meta-data se transmitirán hacia OnBase en segundo plano.

### 3.2.3 Proceso de indexación en segundo plano

Cada vez que se finalice una operación, se levantará una tarea en segundo plano y realizará las siguientes operaciones:

- a) Identificará las órdenes de indexación (documento digitalizado y metadata) que se deberán cargar hacia OnBase.
- b) Con la información recolectada, procederá a cargar cada uno de los documentos alojados en la ruta local de salida, asignando los keywords necesarios e invocando iterativamente el servicio correspondiente de indexación de documentos.
- c) Cada documento que se ha transmitido correctamente hacia OnBase, se marcará como enviado en la base de datos del sistema SCS y su respectivo archivo local será depurado.
- d) Toda la transmisión irá registrando en un archivo local la traza correspondiente, indicando si las transmisiones ocurrieron satisfactoriamente o no, con la finalidad de poder auditar el servicio fácilmente.

### 3.3 REQUERIMIENTOS FUNCIONALES

Código: RF001	Prioridad: 5
El sistema deberá permitir la visualización de los documentos digitalizados desde el escáner del usuario.	
Código: RF002	Prioridad: 5
El sistema deberá presentar la lista de documentos indexados en OnBase de una operación de crédito existente.	
Código: RF003	Prioridad: 5
El sistema deberá presentar la lista de documentos requeridos para una operación de crédito existente.	
Código: RF004	Prioridad: 5
El sistema deberá permitir la visualización de los documentos indexados en OnBase de una operación de crédito.	
Código: RF005	Prioridad: 5
El sistema deberá permitir la descarga de los documentos indexados en OnBase de una operación de crédito.	
Código: RF006	Prioridad: 5
El sistema deberá permitir que el usuario pueda asociar un documento digitalizado por el escáner del usuario con un documento requerido	
Código: RF007	Prioridad: 5
El sistema deberá permitir que el usuario pueda deshacer las asociaciones entre un documento digitalizado y un documento requerido siempre y cuando aún no se hayan indexado en OnBase.	
Código: RF008	Prioridad: 5

El sistema deberá permitir la eliminación de los documentos digitalizados por el escáner.	
Código: RF009	Prioridad: 5
El sistema deberá permitir la eliminación de los documentos indexados en OnBase de una operación de crédito.	
Código: RF010	Prioridad: 5
El sistema deberá permitir indexar en OnBase los documentos digitalizados con la meta información correspondiente ingresada previamente en el SCS.	
Código: RF011	Prioridad: 5
El sistema se podrá ejecutar directamente desde el sistema SCS.	
Código: RF012	Prioridad: 5
El sistema deberá permitir la visualización web de los documentos indexados en OnBase de una operación de crédito.	
Código: RF013	Prioridad: 5
El sistema podrá ejecutarse únicamente por aquellos usuarios que tengan el permiso necesario configurado en el sistema SCS.	
Código: RF014	Prioridad: 5
El sistema deberá guardar una bitácora de cada operación ejecutada.	
Código: RF015	Prioridad: 5
El sistema deberá presentar el nombre del titular de la operación de crédito sobre la cual se está trabajando.	
Código: RF016	Prioridad: 5
El estado de la indexación (pendiente, indexado) de cada documento se deberá poder consultar desde el aplicativo SCS.	

Código: RF017	Prioridad: 5
La lista de documentos requeridos para una operación de crédito se deberá poder configurar desde el aplicativo SCS.	
Código: RF018	Prioridad: 5
La lista de keywords que se deba adjuntar a cada documento indexado se deberá poder configurar desde el aplicativo SCS.	
Código: RF019	Prioridad: 5
Los keywords con los que se indexe cada documento deberá poder ser consultada desde el aplicativo cliente de OnBase.	
Código: RF020	Prioridad: 5
El sistema deberá poder trabajar con los siguientes formatos de archivos: TIFF y PDF.	
Código: RF021	Prioridad: 5
El sistema deberá tener acceso al sistema de archivos del PC del usuario para poder leer y escribir información en un directorio particular.	
Código: RF022	Prioridad: 5
El sistema deberá realizar la indexación de documentos mediante un proceso que se ejecute en segundo plano.	
Código: RF023	Prioridad: 5
Si existe un error durante el envío de los documentos desde el PC del usuario hacia el servidor, se deberá manera un reproceso automático del envío.	
Código: RF024	Prioridad: 5

Cada documento requerido del listado, deberá contener su propia lista de keywords con los valores correspondientes obtenidos desde el sistema SCS.	
Código: RF025	Prioridad: 5
Si el aplicativo cliente se llegara a cerrar por algún motivo, el proceso de indexación en segundo plano deberá continuar indexando si tuviere algún documento pendiente.	

### 3.4 ATRIBUTOS DE CALIDAD

Código: AC001	Tipo: Seguridad	Prioridad: 5
El aplicativo cliente deberá validar el usuario de sesión de Windows contra los permisos que tenga configurado en el sistema SCS.		
Código: AC002	Tipo: Seguridad	Prioridad: 5
El tipo de acceso (consulta o modificación) a cada documento de cada operación deberá ser configurado en el sistema SCS. Esta configuración debe ser por: documento y etapa.		
Código: AC003	Tipo: Seguridad	Prioridad: 5
La transmisión de información entre el aplicativo cliente y el servidor (red de usuarios) deberá ser a través del protocolo HTTPS.		
Código: AC003	Tipo: Seguridad	Prioridad: 5
La transmisión de información entre el servidor de Gestor Documental y los servidores del aplicativo SCS y OnBase (red de servidores) deberá ser a través del protocolo HTTP.		
Código: AC003	Tipo: Desempeño	Prioridad: 5

El tiempo de ejecución de una consulta de documentos requeridos hacia el sistema SCS no deberá ser mayor a 1 segundo.		
Código: AC004	Tipo: Desempeño	Prioridad: 5
El tiempo de ejecución de una consulta de documentos indexados hacia el sistema OnBase no deberá ser mayor a 2 segundos por cada documento.		
Código: AC005	Tipo: Desempeño	Prioridad: 5
El tiempo de ejecución de la indexación de un documento hacia el servidor de Gestor Documental no deberá ser mayor a 2 segundos por cada documento.		
Código: AC006	Tipo: Desempeño	Prioridad: 5
El sistema deberá ser capaz de soportar la indexación concurrente de documentos, en una tasa de hasta 4 documentos/segundo.		
Código: AC007	Tipo: Disponibilidad	Prioridad: 5
Se deberá asegurar la alta disponibilidad a nivel servidores de Gestor Documental.		

## CAPÍTULO 4

### ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

#### 4.1 ANÁLISIS DE LA INTEGRACIÓN

De acuerdo a los requerimientos descritos en los apartados anteriores (Secciones 3.3, 3.4), se procederá a analizar las integraciones que se deberán implementar entre el Gestor Documental y los demás sistemas con los que deberá interactuar e intercambiar información. Cada una de las interacciones se encuentra descrita en la Tabla 8.

Tabla 8. Integraciones de Gestor Documental

<b>Sistema</b>		<b>Funcionalidad</b>
Sistema de Solicitudes	de	Consulta de información general del titular de la operación.
Contratación de Servicios	de	Consulta de documentos requeridos para la operación.

Sistema	Funcionalidad
	Actualización de estado de indexación de documentos.
Sistema de Gestión de Contenido Empresarial	Consulta de los documentos indexados en OnBase.
	Consulta de keywords de un documento en OnBase.
	Consulta de la imagen de un documento en OnBase.
	Actualización de keywords de un documento en OnBase.
	Proceso de indexación de documentos hacia OnBase.

Así mismo, estas interacciones se encuentran representadas en la Figura 4.1, estableciendo así los límites de cada aplicación.

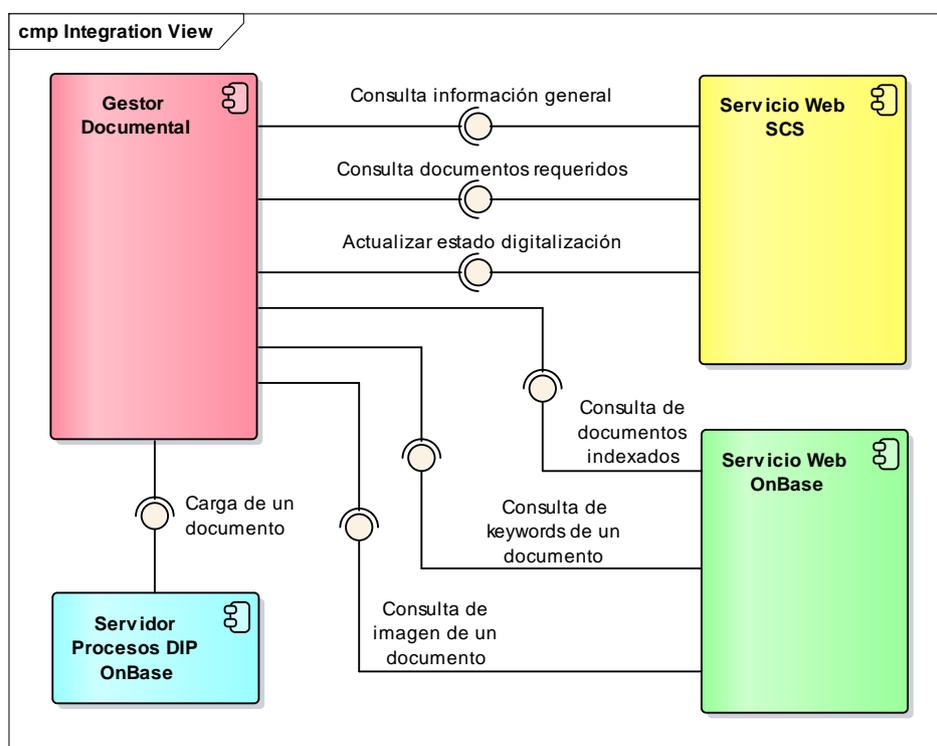


Figura 4.1. Integración entre Gestor Documental, SCS y OnBase

#### **4.1.1 Sistema de Solicitudes de Contratación de Servicios**

El aplicativo de Gestor Documental se convertirá en un cliente para el sistema SCS. Esto quiere decir que el SCS deberá ofrecer una lista de servicios para que Gestor Documental los consuma y pueda implementar la funcionalidad deseada. A continuación, se describirán cada uno de éstos servicios.

- Consulta de información general del titular de la operación. Permitirá obtener la información personal de un cliente o titular de una operación de crédito.
- Consulta de documentos requeridos para la operación. Permitirá obtener el listado de documentos requeridos para una operación de crédito. Cada documento deberá traer consigo su propia lista de keywords, y cada keyword, su valor correspondiente de acuerdo a la información del expediente.
- Actualización de estado de indexación de documentos. Permitirá guardar el estado de la indexación de cada documento que se vaya procesando, desde Gestor Documental hacia el sistema SCS.

#### **4.1.2 Sistema de Gestión de Contenido Empresarial**

Así mismo, el aplicativo de Gestor Documental se convertirá en un cliente para el sistema OnBase. Esto quiere decir que OnBase deberá ofrecer una lista de servicios para que Gestor Documental los consuma y pueda implementar la funcionalidad deseada. A continuación, se describirán cada uno de éstos servicios.

- Consulta de los documentos indexados en OnBase. Permitirá encontrar la meta información de documentos que hayan sido previamente indexados en OnBase. La meta información que se recibe por medio de este método no son los keywords, sino propia de OnBase.
- Consulta de keywords de un documento en OnBase. Permitirá obtener los keywords asociados a un documento que haya sido previamente indexado en OnBase.
- Consulta de la imagen de un documento en OnBase. Permitirá obtener la información binaria de un documento que haya sido previamente indexado en OnBase.
- Actualización de keywords de un documento en OnBase. Permitirá actualizar los keywords asociados a un

documento que haya sido previamente indexado en OnBase.

### **Carga de documentos hacia OnBase**

Luego del análisis realizado, el diseño de la integración hacia los aplicativos SCS y OnBase estaba definido como un esquema totalmente en línea, a través de servicios web. Sin embargo, con la finalidad de cumplir los requerimientos de calidad solicitados, se procedió a realizar las respectivas pruebas de carga preliminares sobre uno de los servicios más críticos: el encargado de la indexación de documentos.

Para estas pruebas, se midió el tiempo de respuesta que podía ofrecer el servicio de carga de documentos hacia OnBase en situaciones de alta concurrencia.

Tal como se puede apreciar en la Tabla 9, el resultado de estas pruebas no fue satisfactorio, incluso en los escenarios en donde se emuló una cantidad de usuarios mayor a doscientos, el servicio colapsaba por una denegación de servicio.

Tabla 9. Tiempo de respuesta promedio del servicio de carga en línea

<b>Usuarios / Periodo (segundos) / Cantidad de requerimientos</b>	<b>Tiempo de respuesta promedio (milisegundos)</b>
50 / 60 / 5	1397
100 / 60 / 5	3204
150 / 60 / 5	8375
200 / 60 / 5	22367
250 / 60 / 5	*
300 / 60 / 5	*

Por este motivo se definió como segunda alternativa para la indexación de documentos un mecanismo mini batch. Este mecanismo se implementó de tal manera que el servicio web de Gestor Documental recibiera los documentos, pero los guardara en un directorio temporal en el servidor. Luego una tarea en segundo plano se estaría ejecutando cada minuto, tomando estos archivos y generando un lote de varios documentos para que un proceso DIP de OnBase los indexe finalmente. Esto implica que, al subir un documento desde el cliente, el mismo estaría disponible en OnBase luego de 2 minutos (en el peor de los escenarios). A pesar de este ligero “retraso” en la operación, este acercamiento permitió que el tiempo de respuesta del servicio de carga se disminuyera en gran manera y que, a su vez, soportara una mayor cantidad de peticiones concurrentes, de

acuerdo a lo que muestra la Tabla 10. Así mismo, la Figura 4.2 muestra la comparación de tiempos de respuesta entre ambas implementaciones del servicio de carga: en línea y mini batch.

Tabla 10. T.E.P. Servicio de carga mini batch

<b>Usuarios / Periodo (segundos) / Cantidad de requerimientos</b>	<b>Tiempo de respuesta promedio (milisegundos)</b>
50 / 60 / 5	183
100 / 60 / 5	526
150 / 60 / 5	807
200 / 60 / 5	1031
250 / 60 / 5	2558
300 / 60 / 5	4254

Tal como se puede observar en la Figura 4.2, este cambio permitió que el proceso de la indexación de los documentos cumpliera con los requerimientos de calidad solicitados.

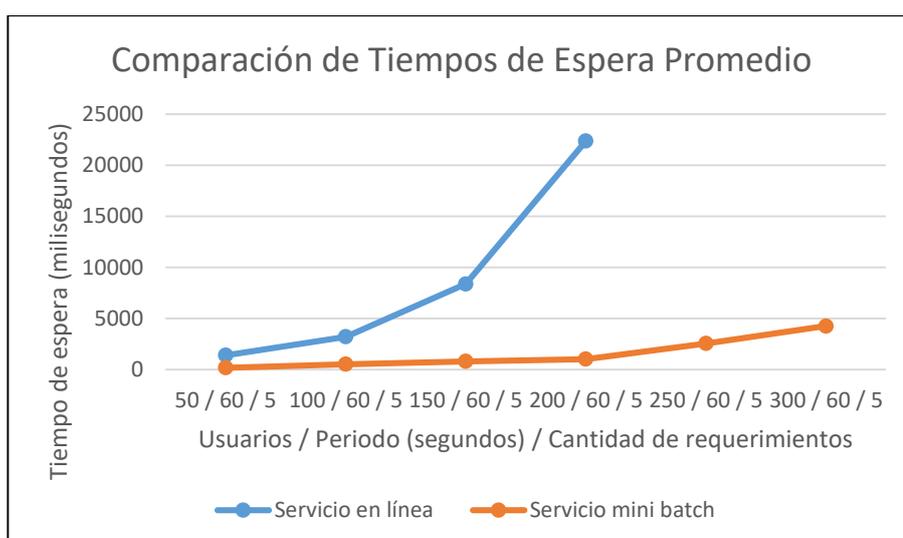


Figura 4.2. Comparación T.E.P. servicio en línea y mini batch

## **4.2 DISEÑO DE LA SOLUCIÓN PROPUESTA**

Para describir el diseño y la arquitectura de la solución propuesta, el aplicativo Gestor Documental, se siguió el Modelo de 4 + 1 Vistas de Kruchten [18]. A continuación, se mostrará la representación de las dos vistas más representativas para nuestro enfoque, las mismas que fueron creadas utilizando el lenguaje de modelado UML 2.

### **4.2.1 Vista de Escenarios**

Esta vista busca representar todas las interacciones que habrá entre el usuario y el sistema a desarrollar. Los principales insumos para elaborar esta vista son: los requerimientos funcionales y no funcionales levantados durante el análisis de la solución. Para este caso se utilizó un Diagrama de Casos de Uso.

La Figura 4.3 muestra a los distintos tipos de usuarios y las interacciones que cada uno podrá tener con el sistema Gestor Documental. Fácilmente se puede observar que los usuarios de la Fábrica de Operaciones y los de la Fábrica de Crédito podrán consultar documentos desde el sistema, mientras que el Asesor Comercial tendrá, además, la opción de indexar nuevos documentos a las operaciones correspondientes.

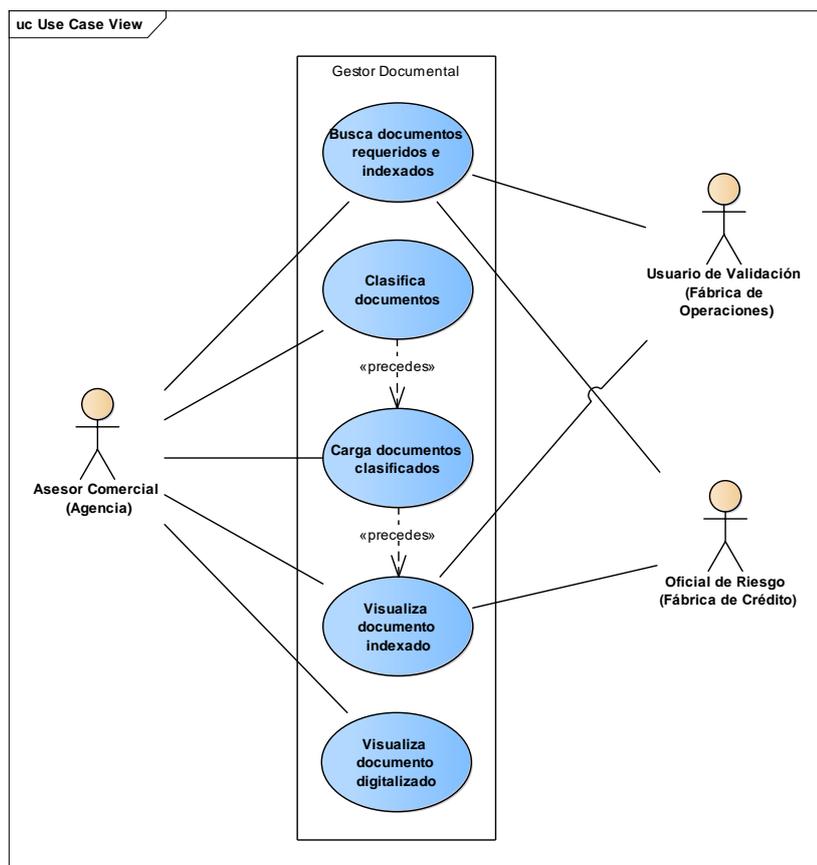


Figura 4.3. VE – Casos de Uso de Gestor Documental

#### 4.2.2 Vista Lógica

Las vistas lógicas tienen como objetivo describir las principales estructuras, atributos y relaciones de las entidades que ayudarán a construir el modelo de la aplicación. Para representar esta vista se utilizaron varios Diagramas de Clase, organizados y distribuidos de acuerdo al componente que conformaron.

## Vista lógica del Cliente Gestor Documental

Las clases más importantes que se construyeron para el aplicativo cliente de Gestor Documental son las representadas a continuación.

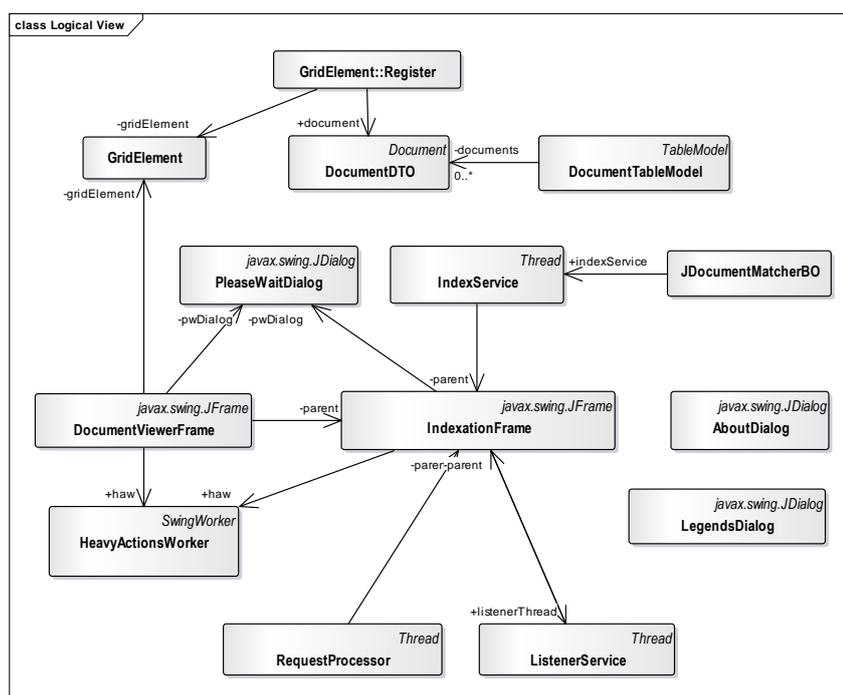


Figura 4.4. VL - Clases principales del cliente de Gestor Documental

La Figura 4.4 muestra las clases principales del programa cliente, entre ellas tenemos:

- Clase `IndexationFrame`, es la ventana principal del aplicativo, orquesta las interacciones del usuario y el comportamiento de los componentes de interfaz gráfica.

- Clase DocumentViewerFrame, es la segunda ventana más utilizada y se permite visualizar en tamaño un documento digitalizado o indexado.
- Clase IndexService, gestiona los hilos de la tarea que corre en segundo plano para realizar la carga de documentos hacia el servidor de Gestor Documental.
- Clase DocumentDTO, es la clase DTO más importante, encapsula toda la información de un documento digitalizado o indexado.

### Vista l3gica del Servicio Web de Gestor Documental

Para el servicio web de Gestor Documental se utilizaron varias clases, a continuaci3n, se representan las m3s importantes.

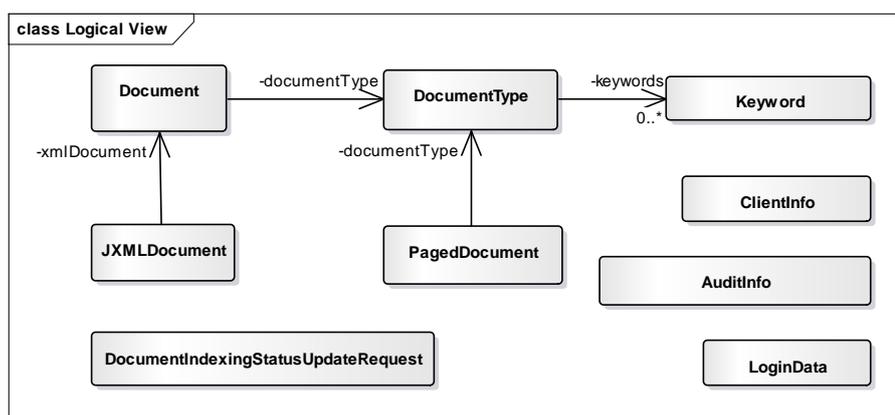


Figura 4.5. VL - Clases de dominio del servicio web de Gestor Documental

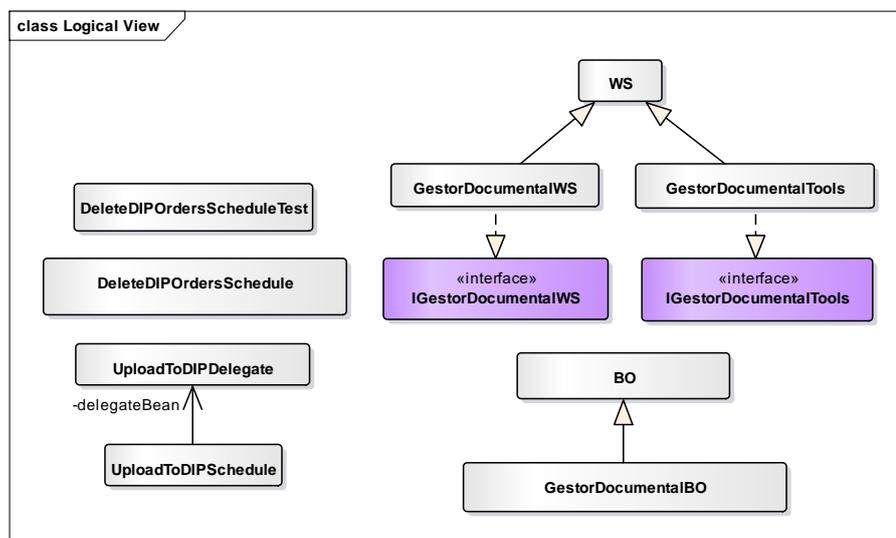


Figura 4.6. VL - Clases principales del servicio web de Gestor Documental

La Figura 4.5 muestra las clases que conforman el dominio del aplicativo Gestor Documental, entre las cuales, las más importantes son:

- Clase Keyword, que permite encapsular la información de un keyword, principalmente su clave y su valor.
- Clase DocumentType, que encapsula la información de un tipo de documento como, por ejemplo: el nombre del tipo (categorización) en la plataforma de OnBase, el código del tipo en la plataforma de OnBase, etc. Además, posee un atributo para almacenar una lista de keywords asociados al tipo de documento.

- Clase Document, que encapsula la información de un documento cargado o indexado en la plataforma: nombre del archivo, código único, información binaria del archivo de imagen como tal, etc. Además, posee un atributo para almacenar el tipo de documento al que pertenece este documento.

La Figura 4.6 muestra las clases principales del servicio web, entre ellas tenemos:

- Clases WS, poseen las anotaciones respectivas a la tecnología JAX-WS, con la finalidad de separar los detalles del contrato de la lógica de negocio.
- Clases BO, contienen toda la lógica de negocio del servicio web.
- Clases Schedule, manejan las tareas en segundo plano dentro del servidor para realizar la indexación hacia OnBase.

### **Vista lógica del Sitio Web de Gestor Documental**

Para el sitio web de Gestor Documental se utilizaron varias clases representadas a continuación.

La Figura 4.7 muestra las clases principales del sitio web, entre ellas tenemos:

- Clase `GestorDocumentalServlet`, encargada de atender las peticiones de ejecución del aplicativo por vía HTTP. Esta clase genera el contrato JNLP en tiempo de ejecución cuando es invocada.
- Clases `GetDocumentServlet` y `GetDocumentPageServlet`, permiten generar documentos y páginas hacia una página JSP para permitir la visualización de documentos a través de la web.

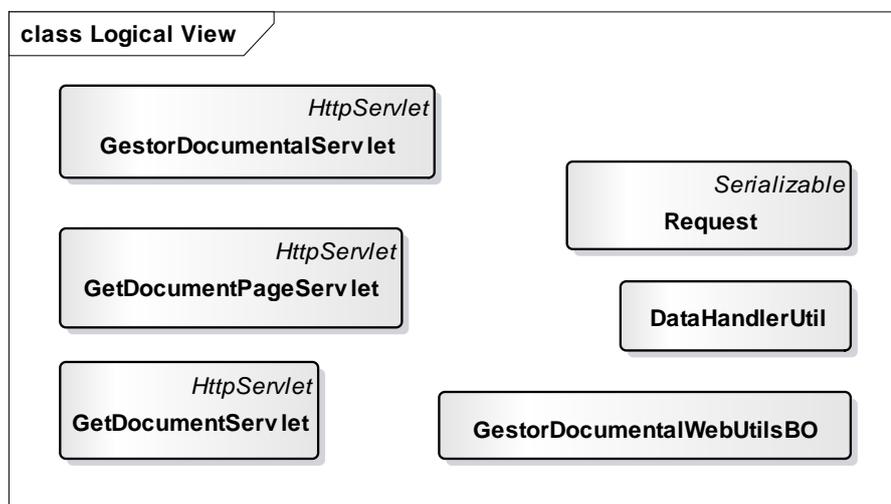


Figura 4.7. VL - Clases principales del sitio web de Gestor Documental

### 4.2.3 Vista de Desarrollo

La Figura 4.8 muestra un Diagrama de Componentes del aplicativo Gestor Documental, representa un nivel más alto que el de la vista lógica del Punto 4.2.2. Tiene como objetivo demostrar la relación que existe entre los principales conjuntos de estructuras del modelo y las interfaces que exponen entre sí.

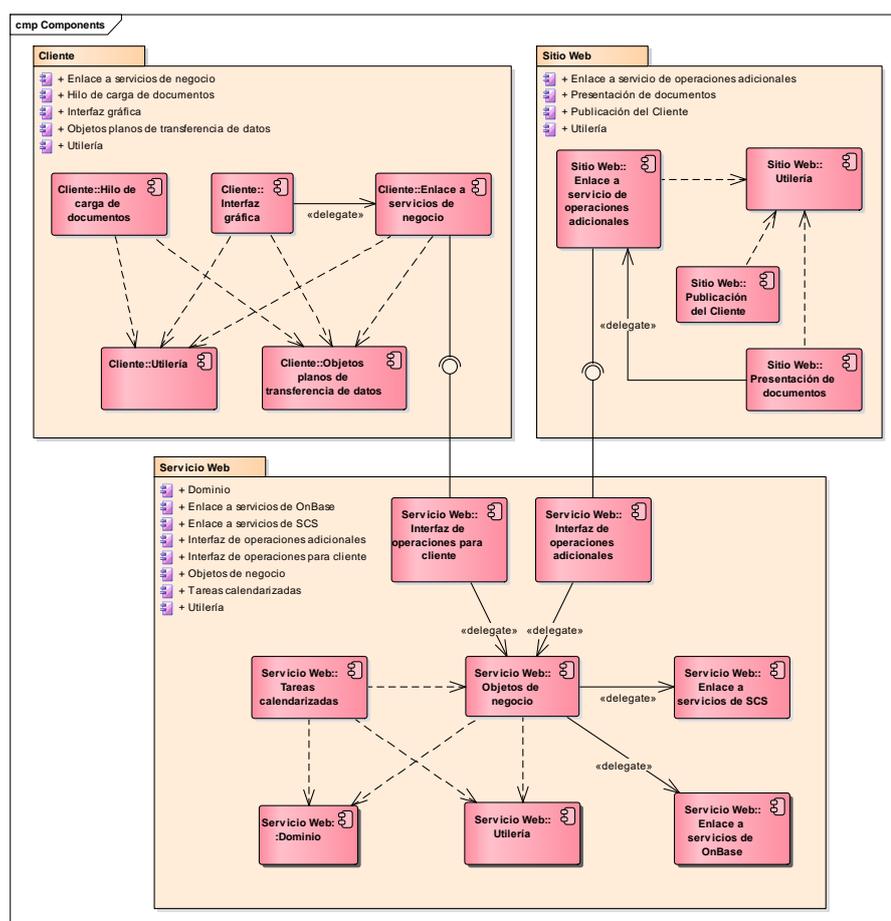


Figura 4.8. VD - Componentes del sistema Gestor Documental

#### 4.2.4 Vista de Procesos

Esta vista permitirá describir el flujo básico del programa, representar los principales puntos de toma de decisiones y las acciones que se disparen durante las mismas. Para elaborar esta vista se utilizaron varios Diagramas de Actividad, organizados y distribuidos de acuerdo al caso de uso correspondiente.

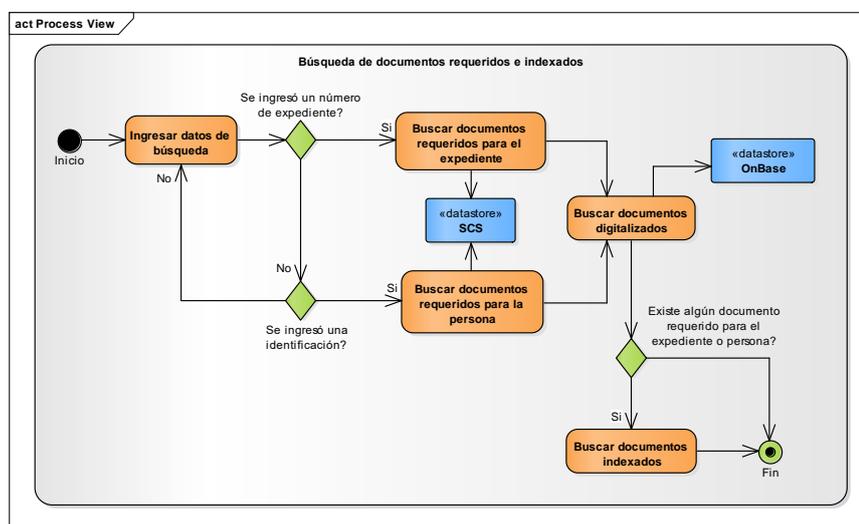


Figura 4.9. VP - Búsqueda de documentos requeridos e indexados

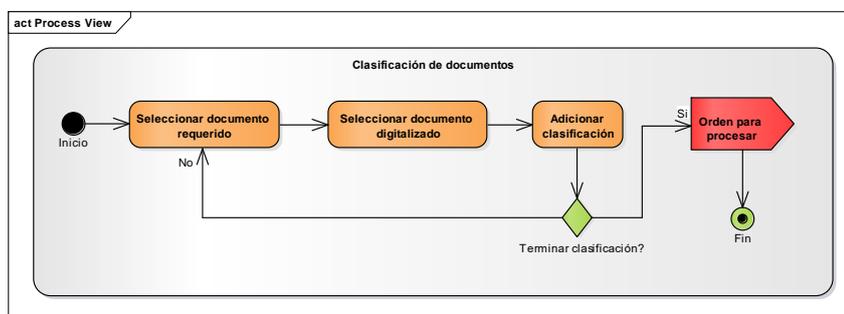


Figura 4.10. VP - Clasificación de documentos

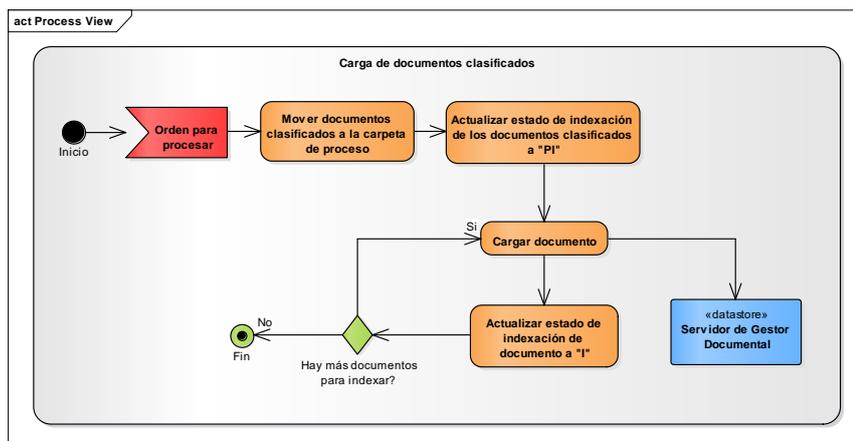


Figura 4.11. VP - Carga de documentos clasificados

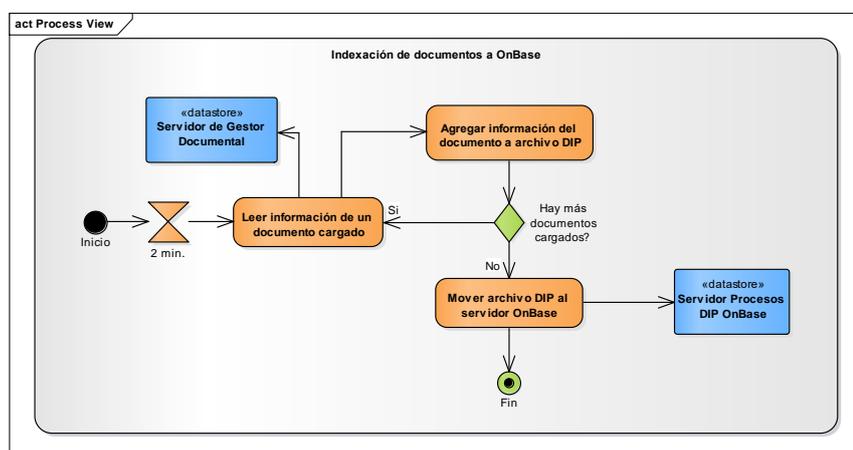


Figura 4.12. VP - Indexación de Documentos a OnBase

Tanto la Figura 4.10 como la 4.11 muestran procesos en segundo plano para subir documentos, sin embargo, no se deben confundir entre sí. El proceso de la Figura 4.10 muestra el proceso que ocurre en el cliente del Gestor Documental luego de haber clasificado todos los documentos. Y el de la Figura 4.11 muestra el proceso que se ejecuta en el servidor de Gestor

Documental cada cierto tiempo para armar el lote DIP que será enviado hacia OnBase.

#### 4.2.5 Vista Física

La vista de despliegue representa la arquitectura del sistema en tiempo de ejecución. Describe la conexión de los principales elementos de hardware y cómo los elementos de software se distribuirán sobre este hardware.

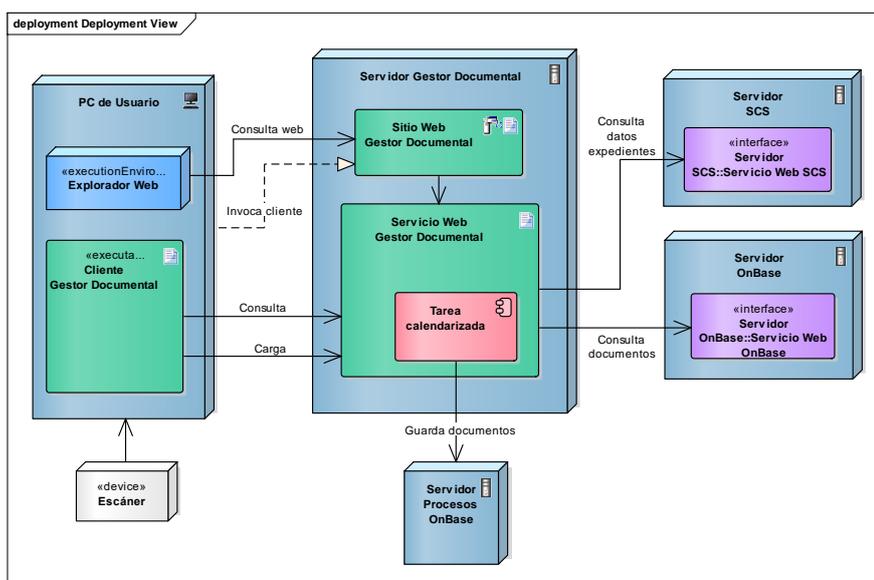


Figura 4.13. VF - Despliegue del sistema Gestor Documental

En la Figura 4.13, se muestra un Diagrama de Despliegue que describe la comunicación entre la PC del usuario, el servidor de Gestor Documental, el servidor de SCS y el servidor de OnBase. Y así mismo, la interacción de los componentes de software

desarrollados: Cliente de Gestor Documental, Servicio Web de Gestor Documental y Sitio de Gestor Documental.

## **CAPÍTULO 5**

### **DESARROLLO Y PRUEBAS DE LA IMPLEMENTACIÓN**

#### **5.1 PLAN DE DESARROLLO**

##### **5.1.1 Lenguaje de programación**

El lenguaje de programación utilizado para la implementación de la solución es Java. Los principales componentes y librerías utilizados se encuentran descritos en el apartado 2.3 “Una solución basada en Java Enterprise Edition” del presente documento. La versión utilizada fue la última correspondiente a cada uno, con la finalidad de evitar alguna incompatibilidad con la tecnología o formato actual.

### **5.1.2 Herramientas de desarrollo**

Para cada componente de software se utilizó un IDE y un ambiente de ejecución diferente debido a la necesidad de explotar funcionalidades y fortalezas de cada uno:

- Cliente de Gestor Documental: Netbeans IDE 8.1, Java Standard Edition 7.
- Sitio y Servicio Web de Gestor Documental: Eclipse Neon 4.6, Java Enterprise Edition 6

### **5.1.3 Herramienta de control de versiones**

Todo el código fuente de la aplicación construida y su respectivo control de versiones es manejado por las herramientas Git y SourceTree.

## **5.2 PLAN DE PRUEBAS**

El Plan de Pruebas de este proyecto fue desarrollado a partir de los criterios de aceptación definidos en el apartado 4.1 “Análisis de los requerimientos” del presente documento.

El detalle de los casos y su respectiva especificación se muestra a continuación en la Tabla 11.

Tabla 11. Plan de Pruebas del aplicativo Gestor Documental

Nº	CASO	ESPECIFICACIÓN
1	Búsqueda de documentos requeridos e indexados	<p><b>Descripción:</b> El usuario realizará una búsqueda de documentos requeridos para un determinado expediente.</p> <p><b>Requerimientos a probar:</b>            RF002: Presentar la lista de documentos indexados en OnBase.            RF003: Presentar la lista de documentos requeridos            RF004: Visualizar desde el cliente pesado los documentos indexados en OnBase.            RF015: Presentar el nombre del titular de la operación.            RF017: Presentar los documentos requeridos de acuerdo a la configuración de SCS.            RF024: Precargar los keywords desde SCS.</p>
2	Clasificación de documentos	<p><b>Descripción:</b> El usuario seleccionará un documento requerido y luego seleccionará un documento digitalizado. Finalmente confirmará dicha asociación de documentos.</p> <p><b>Requerimientos a probar:</b>            RF006: Que el usuario pueda asociar un documento digitalizado con un documento requerido.            RF007: Que el usuario pueda deshacer las asociaciones.</p>
3	Carga de documentos clasificados	<p><b>Descripción:</b> El usuario procederá a confirmar todas las asociaciones que haya realizado con la finalidad de indexar los documentos en OnBase.</p> <p><b>Requerimientos a probar:</b>            RF010: Cargar el documento con toda su meta información.            RF016: Actualizar el estado de la carga en SCS.</p>

Nº	CASO	ESPECIFICACIÓN
		RF018: Cargar la meta información de acuerdo a la configuración de SCS RF019: Visualizar la meta información en OnBase. RF022: Cargar el documento mediante proceso en segundo plano. RF023: Reprocesar automáticamente en caso de error. RF025: Reprocesar automáticamente en caso de cierre inesperado.
4	Visualización de documentos cargados	<b>Descripción:</b> El usuario realizará una búsqueda de documentos y seleccionará un documento indexado, luego seleccionará la opción “Ver” para poder visualizarlo en pantalla. <b>Requerimientos a probar:</b> RF002: Presentar la lista de documentos indexados en OnBase. RF004: Visualizar desde el cliente pesado los documentos indexados en OnBase. RF009: Eliminar los documentos indexados en OnBase. RF012: Visualizar desde el explorador los documentos indexados en OnBase
5	Visualización de documentos digitalizados	<b>Descripción:</b> El usuario seleccionará un documento digitalizado, luego seleccionará la opción “Ver” para poder visualizarlo en pantalla. <b>Requerimientos a probar:</b> RF001: Visualizar de los documentos digitalizados desde el escáner. RF008: Eliminar los documentos digitalizados. RF020: Que se puedan ver archivos TIFF y PDF.

Nº	CASO	ESPECIFICACIÓN
6	Reutilización de documentos cargados	<p><b>Descripción:</b> El usuario realizará una búsqueda de documentos y seleccionará un documento indexado, luego seleccionará la opción “Ver” para poder visualizarlo en pantalla. Luego seleccionará la opción “Reutilizar” para poder guardarlo en un directorio local del computador.</p> <p><b>Requerimientos a probar:</b>  RF004: Visualizar desde el cliente pesado los documentos indexados en OnBase.  RF005: Permitir la descarga de los documentos indexados en OnBase</p>
7	Ejecución de la aplicación	<p><b>Descripción:</b> El usuario accederá a una página web de prueba y seleccionará la opción “Ejecutar” para poder abrir el aplicativo.</p> <p><b>Requerimientos a probar:</b>  RF011: Ejecución desde una página web.  RF013: Validación de usuario autorizado.  RF021: Validación de acceso al directorio de trabajo.</p>

### 5.3 RESULTADOS DEL PLAN DE PRUEBAS

Luego de finalizado el desarrollo del aplicativo, cada caso del Plan de Pruebas fue evaluado para registrar la respectiva aprobación del usuario del aplicativo Gestor Documental. Los resultados de éstas pruebas se listan a continuación en la Tabla 12.

Tabla 12. Resultados del Plan de Pruebas de Gestor Documental

<b>NC</b>	<b>NT</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>
1			
	RF002	Si existiesen documentos previamente cargados a OnBase para la operación ingresada, se deben listar.	OK
	RF003	Se debe mostrar la lista de documentos requeridos	OK
	RF004	Si existiesen documentos previamente cargados a OnBase para la operación ingresada, se debe poder mostrar la imagen de cada uno.	OK
	RF015	Se debe mostrar el nombre del titular de la operación ingresada.	OK
	RF017	Si se agrega o quita un documento requerido desde la configuración de SCS, el cambio se debe reflejar en la búsqueda realizada desde Gestor Documental.	OK
	RF024	Todos los keywords obligatorios de cada documento requerido deben obtenerse desde SCS.	OK
2			
	RF006	Se debe poder seleccionar un documento requerido y una imagen digitalizada.	OK
	RF007	Se debe poder borrar una selección hecha anteriormente.	OK
3			
	RF010	Todos los keywords recibidos desde SCS para el documento deben guardarse en OnBase cuando se cargue una imagen para dicho documento.	OK
	RF016	Se debe actualizar el estado "Indexado" en SCS una vez que el documento ha sido cargado satisfactoriamente en OnBase.	OK
	RF018	Si se agrega o quita un keyword para algún documento requerido desde la	OK

<b>NC</b>	<b>NT</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>
		configuración de SCS, el cambio se debe reflejar en la carga realizada por Gestor Documental.	
	RF019	Todos los keywords cargados en OnBase deben poder consultarse en OnBase.	OK
	RF022	La carga del documento no debe bloquear la interfaz gráfica del aplicativo Gestor Documental.	OK
	RF023	Si se da un error en la carga, el documento se seguirá intentando guardar en segundo plano.	OK
	RF025	Si el aplicativo de Gestor Documental es cerrado mientras carga un documento, éste proceso no finalizará hasta que se carguen todos los documentos pendientes.	OK
4			
	RF002	Si existiesen documentos previamente cargados a OnBase para la operación ingresada, se deben listar.	OK
	RF004	Si existiesen documentos previamente cargados a OnBase para la operación ingresada, se debe poder mostrar la imagen de cada uno.	OK
	RF009	Si la configuración de SCS lo permite, se podrá seleccionar y eliminar el documento indexado en OnBase.	OK
	RF012	Se debe visualizar el documento indexado en Internet Explorer.	OK
5			
	RF001	Si existiesen documentos previamente digitalizados, se debe poder mostrar la imagen de cada uno.	OK
	RF008	Se podrán seleccionar y eliminar los documentos digitalizados.	OK

<b>NC</b>	<b>NT</b>	<b>Resultado esperado</b>	<b>Resultado obtenido</b>
	RF020	Se podrán cargar documentos en formato TIFF y PDF.	OK
6			
	RF004	Si existiesen documentos previamente cargados a OnBase para la operación ingresada, se debe poder mostrar la imagen de cada uno.	OK
	RF005	El documento indexado podrá descargarse en la PC del usuario en el formato original.	OK
7			
	RF011	Al dar clic en el enlace de Gestor Documental, éste se debe ejecutar sin inconvenientes.	OK
	RF013	Si el usuario no tiene autorización para utilizar Gestor Documental, la aplicación no se abrirá. Caso contrario, se abrirá de forma correcta.	OK
	RF021	El aplicativo creó las carpetas de trabajo y escribió en el log sin inconvenientes.	OK

## **CAPÍTULO 6**

### **ANÁLISIS DE RESULTADOS**

#### **6.1 DISMINUCIÓN DEL TIEMPO PROMEDIO DE LIQUIDACIÓN DE SOLICITUDES DE CRÉDITO**

La Tabla 11 muestra los tiempos promedio de liquidación de solicitudes entre los dos escenarios, el anterior (previo a la utilización de Gestor Documental), y el actual (luego de la implementación y adopción del Gestor por parte del usuario). El tiempo que se muestra es la cantidad de días transcurridos entre el ingreso de la solicitud de crédito (primera interacción con el cliente) y el momento en que el valor solicitado es acreditado en la cuenta del cliente.

Tabla 13. Comparación tiempos liquidación de solicitudes de crédito

<b>Tiempo promedio de liquidación de solicitudes de crédito (días)</b>		<b>Porcentaje de disminución</b>
<b>Escenario anterior</b>	<b>Escenario actual</b>	
5 (Guayaquil y Quito)	3 (Todo el país)	40%
8 (Resto del país)		62.5%

Tal como se puede apreciar, luego de la implementación del aplicativo, el tiempo en mención se redujo drásticamente; incluso eliminando la brecha que existía entre las ciudades de Guayaquil y Quito y las demás ciudades del país. Esta brecha era ocasionada por el tiempo extra que tomaba transportar los documentos físicos hacia los centros de digitalización en las ciudades de Guayaquil y Quito (ver Tabla 12).

Esta disminución influye directamente la satisfacción del cliente de forma positiva, ya que el mismo puede contar con el dinero solicitado mucho antes de lo planificado. Es decir que, en la mayoría de los casos, se logra exceder las expectativas del cliente.

A continuación, se muestra la Tabla 12, que por otra parte presenta el tiempo que transcurre desde que un asesor comercial recibe los documentos físicos por parte del cliente hasta que los mismos son digitalizados e indexados en la plataforma de OnBase, tanto en el escenario anterior (previo a la implementación de Gestor Documental),

como en el escenario actual (posterior a la implementación de Gestor Documental).

Tabla 14. Tiempo promedio de digitalización de documentos.

Tiempo promedio de digitalización de documentos (días)		Porcentaje de disminución
Escenario anterior	Escenario actual	
2 (Guayaquil)	<1 (Todo el país)	87.5%
5 (Resto del país)		95%

En la Figura 6.1 se puede apreciar que éste tiempo es el que se vio verdaderamente afectado luego de la implementación del sistema Gestor Documental, disminuyendo drásticamente.

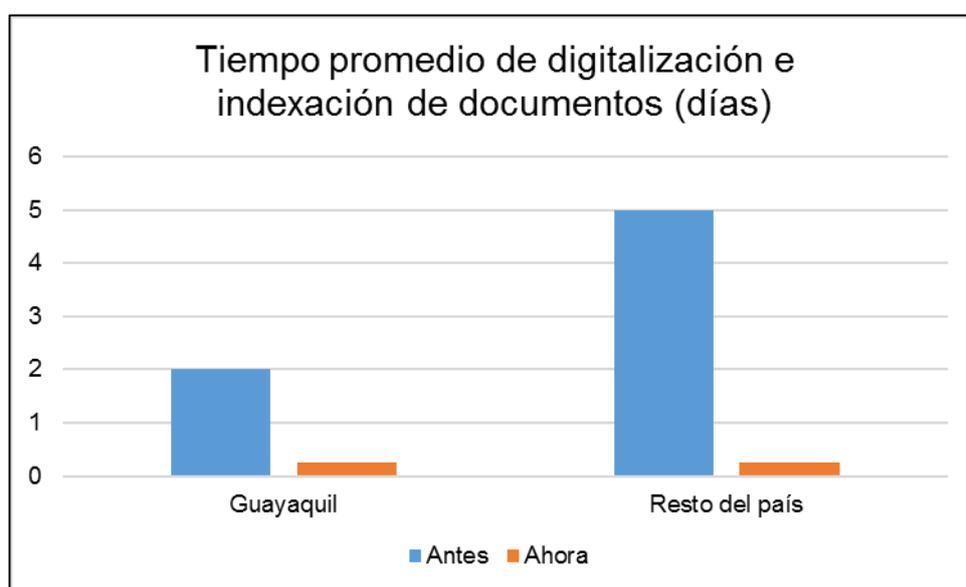


Figura 6.1. Tiempo promedio de digitalización de documentos

## 6.2 INCREMENTO DE LA CANTIDAD DE OPERACIONES DE CRÉDITO DIARIAS

La Figura 6.2 muestra el promedio diario de operaciones de crédito ingresadas cada mes en el aplicativo SCS. Los meses tomados en cuenta para el diagrama son desde marzo del año 2016 hasta noviembre del mismo año. Los tres primeros meses corresponden al momento previo de la implementación de la solución y los seis últimos al momento posterior de la implementación de la misma.

En primera instancia se pensó que asignar tareas de digitalización de documentos a los asesores comerciales disminuiría su productividad. Sin embargo, conforme la curva de aprendizaje se fue superando por parte de los usuarios, la productividad de los mismos se ha mantenido en niveles positivos.

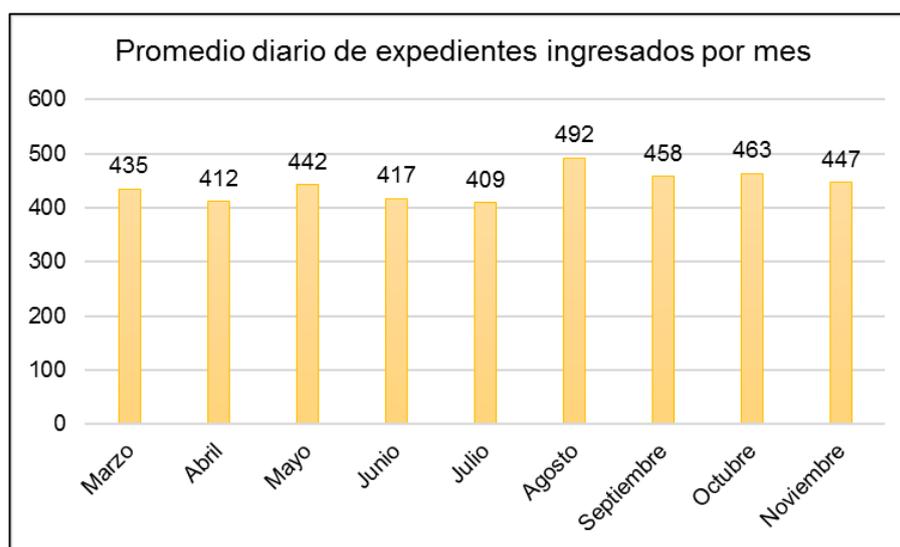


Figura 6.2. Promedio diario de expedientes ingresados por mes

La Figura 6.3 muestra el porcentaje de operaciones liquidadas por mes (expedientes liquidados contra expedientes ingresados); de acuerdo a esto, observamos que a pesar de que se están ingresando la misma cantidad de operaciones cada día, un mayor porcentaje de las mismas culminan satisfactoriamente (se liquidan). Esto se debe a que las tareas de seguimiento, validación y corrección documental pasaron de durar semanas a días, logrando un índice de deserción mucho menor.

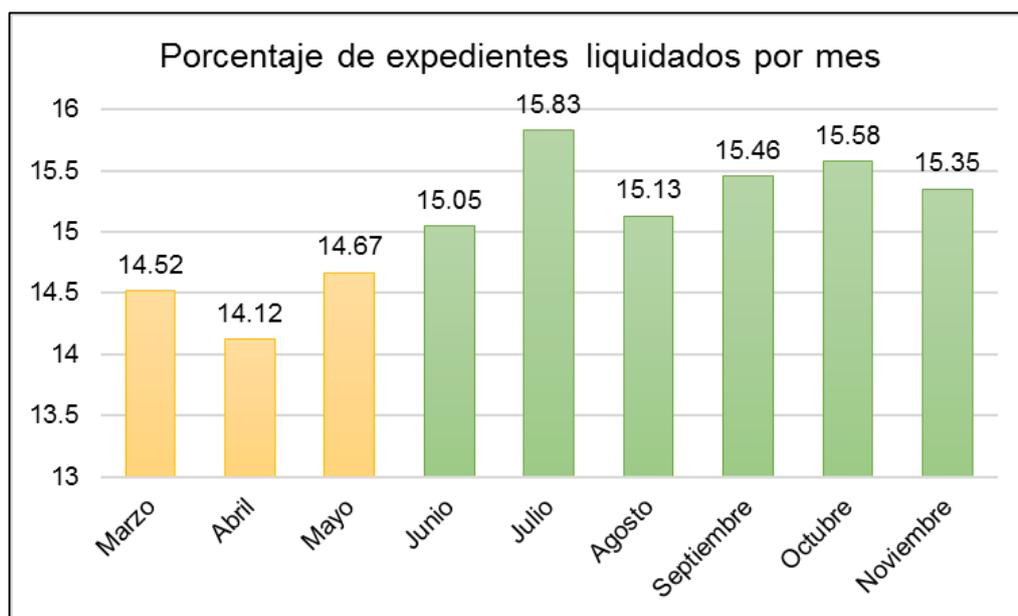


Figura 6.3. Porcentaje de expedientes liquidados por mes

Además, al conocer los datos de la Figura 6.2 podemos estimar la carga actual que está soportando el sistema de Gestor Documental. Al hablar de un promedio de 400 operaciones de crédito diarias (aproximadamente

8000 operaciones al mes), y una cantidad promedio de 12 documentos por operación, podemos decir que el sistema actualmente está soportando la indexación de 96000 documentos digitales cada mes. Esto, dejando a un lado otros productos que también utilizan el Gestor Documental y se comercializan por la plataforma SCS, tales como: cartas de referencia, cuentas de ahorro, cuentas corrientes, seguros, tarjetas de crédito, etc.

### **6.3 COSTO DE IMPLEMENTACIÓN INFERIOR AL DE LA SOLUCIÓN PROPIETARIA**

De acuerdo a lo explicado en el apartado 1.2 (Descripción del Problema), uno de los principales motivos por el que se optó por desarrollar una herramienta personalizada fue el de los altos costos de licenciamiento de módulos adicionales para la actual herramienta ECM de la empresa.

De acuerdo a conversaciones sostenidas con el proveedor, para solventar nuestra necesidad puntual, se requería adquirir licencias de tipo “Desktop Document Imaging” para OnBase. La Tabla 15 muestra el cálculo preliminar del presupuesto necesario si se hubiera decidido optar por esta alternativa, tomando en cuenta su precio y la cantidad de asesores comerciales que hubieran necesitado de esta licencia.

Tabla 15. Estimación de costo de licencias OnBase

<b>Licencia</b>	<b>Precio unitario</b>	<b>Cantidad</b>	<b>Total</b>
Desktop Document Imaging	\$500.00	500	\$250,000.00

Así mismo, la Tabla 16 muestra el cálculo del costo del proyecto tomando en cuenta los tiempos registrados en el cronograma de este trabajo de titulación, y precios referenciales actuales.

Tabla 16. Estimación de costo de desarrollo de Gestor Documental

<b>Rubro</b>	<b>Costo mensual</b>	<b>Meses</b>	<b>Subtotal</b>
Desarrollador Senior J2EE	\$1,500.00	4	\$6,000.00
Computador para desarrollo	\$16.67	4	\$66.68
Herramientas de desarrollo	\$0.00	4	\$0.00
Otros gastos	\$50.00	4	\$200.00
<b>Total</b>			<b>\$6266.68</b>

Como se puede observar, el haber optado por la opción del desarrollo personalizado, representó un ahorro del 97% para el proyecto.

### **Otros factores de ahorro indirecto**

El back-end de Gestor Documental fue diseñado para funcionar de una forma desacoplada con respecto al cliente pesado que correría en las

PCs de los asesores comerciales. Esto permitió que una vez que el back-end alcanzara un nivel de estabilidad y madurez determinado, se pudieran diseñar interfaces para que otros aplicativos puedan utilizar las funcionalidades *core* del Gestor Documental a través de otros servicios web.

La Figura 6.4 muestra cuáles son los aplicativos que pudieron aprovechar el back-end de Gestor Documental para comunicarse con el sistema ECMS de una forma fácil y confiable.

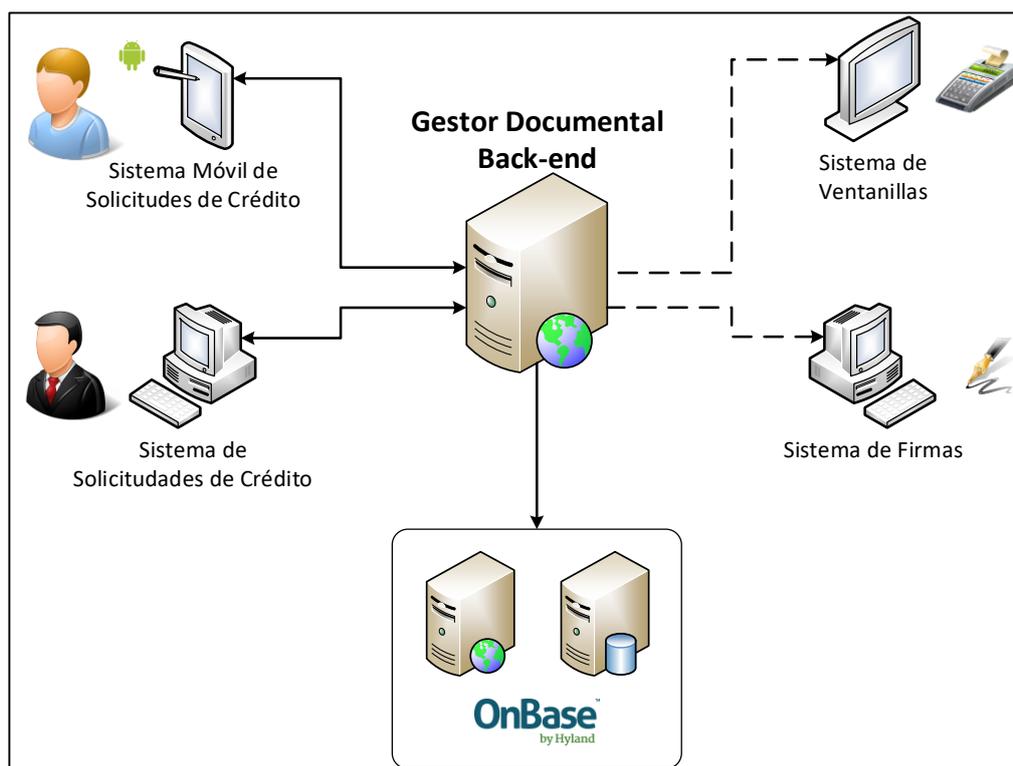


Figura 6.4. Ecosistema de clientes de Gestor Documental

A continuación, se mencionan y describen dos casos de los mostrados en la Figura 6.4.

#### Integración con Sistema Móvil de Solicitudes de Crédito

Este sistema móvil es la principal herramienta para los asesores que comercializan los productos de crédito que se orientan a microempresarios. Es un aplicativo móvil Android que permite el ingreso de solicitudes de este tipo y la digitalización de documentos habilitantes a través de su cámara fotográfica. Una vez capturados los documentos se indexan en OnBase utilizando una interfaz de Gestor Documental.

Todo el ingreso de la solicitud puede ser realizado desde cualquier lugar con acceso a Internet: Negocio del microempresario (a través de la red celular 3G), domicilio del asesor comercial (a través de su propia red WLAN), etc.

#### Integración con Sistema de Ventanillas

Existen algunas transacciones que por su naturaleza deben contar con documentos adicionales de soporte por parte del cliente, por ejemplo: Un documento de Licitud de Fondos firmado por el cliente para los casos en que el valor de un depósito sobrepasa cierto umbral. Estos documentos habilitantes deben guardarse en el ECMS. En la actualidad, la integración de nuestro Sistema de Ventanillas con el back-end de Gestor

Documental ya está implementada en ambiente productivo, dando resultados positivos.

En la actualidad, la entidad financiera cuenta con aproximadamente 150 Asesores de Micro-Crédito y 400 Cajeros de Ventanillas. Cada uno de ellos hubiera necesitado de una licencia especial de OnBase de no ser por la integración que se logró a través de Gestor Documental.

La Tabla 17 nos muestra de forma resumida la comparación de los costos entre las licencias que hubieran sido necesarias de OnBase y los costos de cada integración con Gestor Documental.

Tabla 17. Comparación costos OnBase y Gestor Documental

<b>Descripción</b>	<b>Costo de licenciamiento Onbase</b>	<b>Costo de implementación Gestor Documental</b>
Funcionalidad core para Asesores Comerciales	\$250000.00	\$6300.00
Funcionalidad para Asesores de Micro-Crédito	\$150000.00	\$1600.00
Funcionalidad para Cajeros de Ventanillas	\$200000.00	\$1600.00
Totales	\$600000.00	\$9500.00
<b>Porcentaje de Ahorro</b>		<b>98%</b>

## **CONCLUSIONES Y RECOMENDACIONES**

### **CONCLUSIONES**

El presente trabajo nos permite concluir lo siguiente:

- Antes de adquirir una solución tecnológica, se debe evaluar si realmente se está pagando por lo que se necesita, y no por características que no le agregan valor a nuestro proceso de negocio.
- Antes de construir una solución a la medida, se debe evaluar qué funcionalidad o componentes tecnológicos pueden ser reutilizados y explotados de forma correcta mediante protocolos y estándares de integración entre aplicaciones.

- La tecnología de código abierto es 100% compatible con tecnología propietaria siempre y cuando exista un estándar definido para las diferentes comunicaciones entre procesos y sistemas.
- La capacidad de poder crear componentes reutilizables no depende de la tecnología ni del protocolo de comunicación implementado sino de un buen diseño de interfaces de comunicación y el tener un dominio limitado y bien definido.
- A pesar de que en nuestro medio una “oficina cero papeles” aún parezca un sueño, el poder analizar procesos en los que intervenga información análoga y redefinirlos con la ayuda de la herramienta correcta puede permitirnos tener logros al corto plazo y conseguir ahorro en tiempo y dinero para la organización.

## **RECOMENDACIONES**

Luego de transcurrido el tiempo de implementación y estabilización de la plataforma, y de haber obtenido la retroalimentación técnica y funcional acerca del sistema Gestor Documental, podemos nombrar las siguientes recomendaciones para futuras actualizaciones del sistema, las cuales están fuera del alcance del presente trabajo de titulación.

- Durante la estabilización del aplicativo en Producción, fue necesario realizar algunas tareas para depuración y resolución de problemas. Esto se dificultó, en cierto modo, debido a que el sistema de Gestor

Documental registra la bitácora de sus operaciones en un archivo de texto plano. Para una siguiente versión se debería diseñar un modelo de datos que permita registrar las operaciones y estados de digitalización de cada documento que se ha cargado mediante el aplicativo, con la finalidad de poder rastrear errores y monitorear el servicio de una forma rápida y sencilla mediante consultas SQL.

- Los documentos que se indexan para una operación suelen ser consultados en reiteradas ocasiones durante ventanas de tiempo de aproximadamente 4 horas, de acuerdo a los estados de verificación y aprobación documental por los que pasa una solicitud. Cada consulta del documento que se realiza se delega al sistema ECMS. Para quitar esta carga al sistema se podría diseñar una capa de caché administrada por el back-end del aplicativo Gestor Documental, que permitiría que la mayor cantidad de consultas que hay sobre un documento durante el tiempo de vida de la solicitud, se ejecutaran sin la necesidad de generar un acceso al ECMS. Así mismo, cada operación de eliminación o actualización de documentos, actualizaría la información binaria del documento en la caché del servidor.

## BIBLIOGRAFÍA

[1], [5] Jesús Tramullas, Gestión de contenidos, 2005-2015: una revisión, <http://raco.cat/index.php/Hipertext/article/view/294025/389440>, Año de publicación: 2015.

[2], [4] AIIM (Association for Information and Image Management), What is Enterprise Content Management (ECM)?, <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>, Fecha de consulta: 03/06/2016.

[3] Salamntu L., Seymour L., A review of enterprise content management (ECM): Growth and Maturation of ECM from the year 2001 to 2011, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7323002>, Fecha de conferencia: 07/10/2015.

[6] Hyland, Acerca de Hyland, <https://www.onbase.com/es-MX/acerca-de-hyland/hyland-software/la-historia-de-hyland>, Fecha de consulta: 03/06/2016.

[7] Nick Geddes, Turning Documents into Data - using data capture to improve, <http://www.kiriworks.com/downloads/WPTurningDocumentintoData.pdf>, Año de publicación: 2013.

[8] Alfresco, Gestión de contenido empresarial, <https://www.alfresco.com/es/products/enterprise-content-management>, Fecha de consulta: 03/06/2016.

[9], [10] Dhouib S., Ben Halima R., Surveying Collaborative and Content Management Platforms for Enterprise, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6570631>, Fecha de conferencia: 17/06/2013.

[11] Hyland, OnBase Main Brochure, [http://www.cortland.edu/business/purchasingforms/OnBase%20main%20brochure%2020pg\(72dpi\).pdf](http://www.cortland.edu/business/purchasingforms/OnBase%20main%20brochure%2020pg(72dpi).pdf), Año de publicación: 2004.

[12] Oracle, The Java EE 6 Tutorial, <http://docs.oracle.com/javaee/6/tutorial/doc/bnaci.html>, Fecha de consulta: 03/06/2016.

[13] Shen Y., Zhang Y., Xie J., Jiang R., Han H., Study on Rainfall Information Integration System based on Java Web Start, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=7311921>, Fecha de conferencia: 23/01/2015

[14] Oracle, Java Web Start Architecture, <http://www.oracle.com/technetwork/java/javase/architecture-138566.html>, Fecha de consulta: 03/06/2016.

[15] Oracle, The Java EE 6 Tutorial, <http://docs.oracle.com/javaee/6/tutorial/doc/bnayl.html>, Fecha de consulta: 03/06/2016

[16] Mhatre H., Mehta B., Jaiswal A., Architecture for MTOM Based File Transfer,

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=65290>

[02](#), Fecha de conferencia: 21/03/2013

[17] Angel Borroy, ENI – Transmisión de ficheros en el esquema del Documento Electrónico, [https://angelborroy.wordpress.com/2012/08/31/eni-](https://angelborroy.wordpress.com/2012/08/31/eni-transmision-de-ficheros-en-el-esquema-del-documento-electronico/)

[transmision-de-ficheros-en-el-esquema-del-documento-electronico/](https://angelborroy.wordpress.com/2012/08/31/eni-transmision-de-ficheros-en-el-esquema-del-documento-electronico/), Fecha de consulta: 03/06/2016.

[18] Philippe B. Kruchten, The 4+1 View Model of Architecture,

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=44591>

[74](#), Fecha de conferencia: 18/02/200

## GLOSARIO

1. Sarbanes-Oxley Act.- Ley que involucra a todas las empresas que cotizan en la Bolsa de Valores de Nueva York y exige la regulación de la práctica financiera y el gobierno corporativo, por medio de la evaluación y divulgación de la eficacia de los procesos de auditoría interna y externa.
2. Proceso DIP. - Procesador de importación de documentos (por sus siglas en inglés), es un proceso que se encarga de la indexación masiva de documentos hacia OnBase, tomando como entrada un archivo de texto plano con la meta-información de un documento y la ruta UNC de la ubicación física del mismo.
3. Diskgroup. - Unidad de almacenamiento lógica, configurada sobre el sistema de archivos del servidor y administrada por OnBase.
4. Tiempo de retención. - Tiempo luego del cual un documento indexado en OnBase se depura automáticamente.
5. Codificación de imágenes. – Proceso mediante el cual la información binaria de una imagen es procesada por un algoritmo específico con la finalidad de: representar en una cadena de caracteres la información binaria, comprimir la información binaria de la imagen, etc.
6. Decodificación de imágenes. – Proceso mediante el cual una imagen codificada puede ser representada y entendida por el computador. Para

esto es necesario aplicar el mismo algoritmo que fue utilizado para su codificación previa.

7. JPEG.- Grupo Conjunto de Expertos en Fotografía (por sus siglas en inglés), es un comité de expertos que diseñó el algoritmo de codificación de imágenes compresas del mismo nombre.
8. G4-CCITT. - La compresión de Grupo 4 de CCITT es un algoritmo de codificación de imágenes compresas para máquinas fax de Grupo 4.
9. ORM. - Mapeo Objeto Relacional (por sus siglas en inglés), es una técnica de programación para convertir estructuras de datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.