



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

“LA INTEGRACIÓN DE SISTEMAS EN EL PROCESO DE OTORGAMIENTO
DE CRÉDITO EN FINANCIERA INDEPENDENCIA”

TRABAJO PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA:

MAURICIO GABRIEL PALMA JIMÉNEZ

DIRECTOR DE TRABAJO PROFESIONAL:

ING. CINDY ARACELI RAMOS PÉREZ



Ciudad Universitaria

2009

A mi alma mater la “Universidad Nacional Autónoma de México”

A la gran Facultad de Ingeniería

A la inigualable Escuela Nacional Preparatoria

***A Miguel y Mireya, por siempre haberme apoyado en todas y cada una de mis decisiones, porque con su amor me han respaldado y defendido de la adversidad, por enseñarme siempre el camino de la honestidad y rectitud, por brindarme siempre lo mejor de ellos, por permitirme realizar una gran meta, por enseñarme a triunfar...
GRACIAS MAMÁ Y PAPÁ.***

A mi hermano Miguel por darme su apoyo y cariño incondicional, por ser mi guardián, por escucharme y guiarme con sus consejos. Gracias por estar a mi lado.

A Martha que ha sido mi hermana y me ha enseñado que nunca debes de rendirte aunque te nieguen lo más querido.

A Eva, Yola, Karlita, Gus y Alejandro, por tener el gusto de haberlos conocido y porque sé que en donde quiera que se encuentren me apoyan igual que como lo hicieron en vida.

A toda mi familia que ha sido cómplice en la realización de todas mis metas, que me ha alentado para que sea una mejor persona día a día.

A mis amigos por enseñarme y permitirme caminar al lado de ellos en la vida, por brindarme su confianza y apoyarme en momentos difíciles.

A la Ing. Cindy Ramos, por todo su apoyo y valiosa asesoría en la elaboración de este trabajo de titulación.

A Financiera Independencia, que me brindó su confianza y me abrió las puertas para iniciar mi carrera profesional. En especial a Pedro García, por proporcionarme la oportunidad de laborar ahí.

A mi país, por invitarme a superar diariamente para contribuir a su mejoramiento.

A Dios por darme vida y salud para llegar hasta donde ahora estoy...

ÍNDICE

Introducción	5
Capítulo 1. – Contexto y Vista General del Sistema de Proceso de Otorgamiento de Crédito	
Otorgamiento de Crédito	
1.1.- Antecedentes de la Integración de Sistemas	7
1.1.2 – Arquitectura Orientada a Servicios (SOA)	11
1.2 – Contexto	12
1.3 – Vista general del sistema	14
1.4 – Definición de la problemática	16
Capítulo 2. - Arquitectura Actual	
2.1 - El Bus de Servicios Empresariales (ESB) de Financiera Independencia	17
2.2 - Subsistemas involucrados	23
2.3 - Objetivos de la arquitectura	27
Capítulo 3. – Flujo general de una solicitud en el Proceso de Otorgamiento de Crédito y mi participación profesional.	
3.1 - Flujo general de una solicitud en el Proceso de Otorgamiento de Crédito	28
3.2 - El Ingeniero de Soporte a Producción en el Proceso de Otorgamiento de Crédito.	51
Conclusión	64
Bibliografía	66
Mesografía	67
Glosario	68

Introducción

Hoy en día se necesita de profesionistas con una formación que permita responder a las diversas necesidades que se presentan en el campo de trabajo de la Ingeniería en Computación. Para que pueda analizar, diseñar, construir y dar soporte técnico a los sistemas electrónicos de cómputo y de programación, considerando el aseguramiento de la calidad en los mismos, además de evaluar, comparar y seleccionar los equipos que ayudaran a cumplir los objetivos del trabajo.

Lo anterior, es exactamente el perfil de egresado que ofrece la carrera de ingeniería en Computación en la Universidad Nacional Autónoma de México, en este trabajo explicaré como un egresado se enfrenta a un reto como lo es *la integración de sistemas*.

Durante la segunda mitad del 2006 y todo el 2007 estuve en el programa de formación de Becarios de la Unidad de Cómputo Académico de la Facultad de Ingeniería (UNICA). En dicho programa, complementé la parte teórica de las asignaturas de la carrera ya que obtuve capacitación en muchas ramas de la ingeniería en computación, que van desde la arquitectura física de las computadoras, hasta desarrollo de software de alto nivel con lo último en lenguajes de programación, así como un amplio panorama en redes de computadoras.

Fue gracias a mi formación integral la que me permitió a principios del 2008 agregarme al entorno laboral. Financiera Independencia, fue la que me brindó una oportunidad al evaluar mis conocimientos y supo que tenía las bases sólidas para abordar la integración de sistemas. De esta forma, me contrataron para la vacante de Ingeniero de Soporte a Producción, la función principal del puesto es mantener todos los sistemas que se involucran en el Proceso de Otorgamiento de Crédito (POC) funcionando óptimamente, encontrar errores en los sistemas (bugs) y proponer soluciones a ellos, así como desarrollar nuevos módulos o aplicaciones complementarias para el POC.

En este trabajo explicaré la transformación de una arquitectura de sistema basada en procesos aislados a una arquitectura basada en servicios (SOA) que son orquestados a través de un Bus de Servicios Empresariales (Enterprise Service Bus – ESB), el cómo es que la formación del ingeniero en computación es indispensable para la integración de sistemas, en un caso práctico, en el POC. Además de explicar cómo se va avanzando la información de una solicitud de crédito en todo el POC hasta llegar al objetivo principal que es la satisfacción de nuestros clientes.

Para concluir este trabajo expondré los resultados que he obtenido en el tiempo que he estado laborando en Financiera Independencia, así como mis aportaciones a la empresa.

Capítulo 1. – Contexto y Vista General del Sistema de Proceso de Otorgamiento de Crédito

1.1.- Antecedentes de la Integración de Sistemas

Según Jeffrey O. Grady¹, la integración de sistemas se define como la interacción de componentes de Hardware, Software y Humanos para alcanzar el propósito de todo el sistema o la satisfacción del cliente.

De acuerdo con Judith M. Myerson² la integración de sistemas es la combinación de tecnologías divergentes y comúnmente incompatibles, aplicaciones, datos y comunicaciones para formar una arquitectura de información uniforme y una estructura de trabajo funcional.

La integración de sistemas se puede definir como la combinación de procesos, software, estándares y hardware resultando en la mejor incorporación posible de dos o más sistemas empresariales permitiéndoles funcionar como uno solo.

En la década de 1980, varias empresas estaban empezando a comprender el valor y la necesidad de integración de sistemas. Los problemas surgieron, cuando muchos miembros del personal de TI (Tecnología de la Información) corporativa trataron de rediseñar las aplicaciones que ya se habían desarrollado para hacerlos aparecer como si estuvieran integrados.

Como las aplicaciones ERP (Enterprise Resource Planning) se habían hecho mucho más frecuentes en la década de 1990, hubo una necesidad de las empresas para poder aprovechar las aplicaciones ya existentes y los datos en los sistemas ERP, lo que sólo podría hacerse mediante la introducción de la integración de sistemas. El traslado hacia la integración era lógico, las empresas, utilizaron tecnología cliente / servidor para crear aplicaciones departamentales, pero más tarde se dieron cuenta de la importancia de vincular los beneficios en múltiples procesos de negocio.

Las empresas comúnmente están compuestas por cientos de “aplicaciones hechas a la medida”, adquiridas de algún tercero, partes de sistemas legados o desarrolladas propiamente, que funcionan en diferentes sistemas operativos. Esto se convierte en arquitecturas muy difíciles de mantener y muy enredadas, sin embargo esto tiene explicación.

¹ Jeffrey O. Grady, *System Integration*, CRC Press, 1994

² Judith M. Myerson, *Enterprise Systems Integration*, CRC Press, 2002

Primero, desarrollar aplicaciones de negocio es muy difícil. Crear una única gran aplicación que soporte todo el flujo del negocio es casi imposible. Las empresas desarrolladoras de aplicaciones ERP han tenido algo de éxito al crear gigantescas aplicaciones de negocio, sin embargo, la realidad es que empresas como SAP, Oracle, PeopleSoft, etc., abarcan solo una fracción de las funciones de negocio que necesita típicamente una empresa.

Segundo, separar las funciones de negocio entre múltiples aplicaciones proporciona flexibilidad al negocio para seleccionar por ejemplo, la mejor forma de hacer su contabilidad, la mejor manera de manejar a sus clientes (software CRM), así como la mejor manera de procesar cada orden de trabajo. Generalmente el área de sistemas de la empresa, no está interesada en una aplicación empresarial que “lo haga todo”, sino en un número de aplicaciones que cubran los requerimientos individuales de negocio.

Para tener procesos de negocio y datos comunes entre aplicaciones, éstas últimas deben estar integradas. La integración de aplicaciones³ debe proporcionar eficiencia, confiabilidad y seguridad en el intercambio de datos entre las múltiples aplicaciones de la empresa.

Los fabricantes de software ofrecen suites de Integración de sistemas que proveen diferentes plataformas, integración de múltiples lenguajes, así como la capacidad de interactuar con muchas aplicaciones populares. Sin embargo, estas características técnicas sólo abordan una pequeña parte de la complejidad de la integración. Los verdaderos retos de la integración van más allá de los problemas técnicos y de las reglas de negocio.

- La integración de sistemas requiere cambios significativos en las políticas de la compañía. Las aplicaciones de negocio generalmente se enfocan en un área en específico tales como, manejo de los clientes, cobranza o crédito. Una integración de sistemas exitosa establece comunicación no sólo entre las computadoras, también debe hacerlo entre las áreas de negocio y los equipos de sistemas.
- Debido a su gran alcance, los esfuerzos de integración típicamente tienen amplias implicaciones en los negocios. Una vez que las funciones críticas de la empresa son incorporadas a la solución integrada, el buen funcionamiento de la solución se convierte en algo vital para la empresa. Las fallas de la solución pueden costar mucho dinero o clientes.

³ En este trabajo se utilizará indistintamente “*integración de sistemas*” e “*integración de aplicaciones*”

- A pesar de la demanda por la solución que proporciona la integración de sistemas, sólo un poco de estándares se han establecido en este dominio. XML; XSL y los Web Services son el mayor avance de estándares para una solución de integración.
- Con los XML Web Services se resuelven sólo una parte de los retos de integración. La existencia de la misma presentación de los datos (por ejemplo XML) no implica que tengan la misma semántica, connotación, restricciones y consideraciones en cada sistema participante. Resolver estas diferencias semánticas puede llevar aparte de tiempo, tomar decisiones técnicas y de negocio importantes.
- Mientras que desarrollar una solución de integración implica un gran reto, operarla y mantenerla puede ser aún más exhausto. La combinación de tecnologías y la naturaleza distribuida de las soluciones de integración hacen que el despliegue, el monitoreo y el soporte a los problemas que surjan requieran una gran combinación de conocimientos y habilidades. En muchos casos, estas habilidades y conocimientos se distribuyen entre muchos miembros del equipo o ni siquiera existen entre la operación de IT.

La integración de sistemas es muy compleja, e incorpora todos los niveles del sistema de una empresa - la arquitectura, hardware, software y procesos, implica la integración en los siguientes niveles:

Integración de los Procesos de Negocio (Business Process Integration): una corporación deberá definir, activar y gestionar los procesos para el intercambio de información entre las diversas empresas de sistemas comerciales. Esto permite a las organizaciones racionalizar las operaciones, reducir costos y mejorar la capacidad de respuesta a las demandas del cliente. Dentro de estos elementos se incluyen la gestión de procesos, modelado de procesos, y el flujo de trabajo, lo que supone la combinación de las tareas, procedimientos, organizaciones, requiere de entrada y salida de información, y herramientas necesarios para cada paso en un proceso de negocio.

Integración de Aplicaciones: En este nivel de integración, el objetivo es reunir datos o la función de una aplicación junto con la otra aplicación que en conjunto proporcionan cerca de la integración en tiempo real. Integración de Aplicaciones se utiliza para, por nombrar unos pocos, integración B2B (Business to Business), la aplicación de gestión de relaciones con clientes (CRM), sistemas que

están integrados con una compañía de aplicaciones backend, integración Web, y la creación de sitios Web que aprovechan múltiples sistemas empresariales.

Integración de Datos: Para que ambos la integración de aplicaciones y Integración de Procesos de Negocios tengan éxito, la integración de datos y sistemas de bases de datos deben ser analizados. Antes de la integración, los datos deben ser identificados (en donde se encuentra), catalogado, y un modelo de metadatos debe construirse (un modelo para varios almacenes de datos). Una vez que estos tres pasos estén terminados, los datos se pueden compartir a través de bases de datos distribuidas.

Estándares de Integración: Con el fin de lograr la plena integración de datos, se deben elegir formatos estándar para los datos. Los estándares de integración son los que promueven el intercambio y la distribución de la información, y de datos de negocios – los estándares son la parte central de la integración de sistemas. Estos incluyen COM+ / DCOM, CORBA, EDI, Java RMI, y XML.

Plataforma de Integración: Para completar la integración de sistemas, la arquitectura, software y hardware, las necesidades de una red heterogénea debe integrarse. La plataforma de Integración se ocupa de los procesos y herramientas que se requieren para permitir que estos sistemas de comunicación, tanto óptima y segura, de modo que los datos pueden transmitirse a través de diferentes aplicaciones sin problemas. Por ejemplo, averiguar una forma de que una máquina NT para transmitir información fiable a una máquina UNIX es una gran tarea para la integración de todo un sistema corporativo.

Una parte fundamental para la integración de sistemas es la arquitectura utilizada, en este caso la que es orientada a servicios (SOA) es la mejor solución ya que su utilidad está en la propia filosofía de dicha arquitectura.

1.1.2 - SOA

La arquitectura de software es una descripción de un sistema en términos de la mayoría de sus componentes, sus relaciones, y la información que se transmite entre ellos. En esencia, la arquitectura es un plan para desarrollar sistemas que cumplan con requerimientos bien establecidos y dicho sistema debe cumplir los requerimientos ahora y seguirlo haciendo en el futuro.

SOA es un estilo arquitectónico para construir soluciones empresariales basadas en servicios. Se enfoca en la construcción independiente de servicios alineados al negocio que pueden combinarse para formar grandes procesos de alto nivel y soluciones en el contexto de la empresa. Crear servicios no es el objetivo de SOA, el verdadero objetivo es reusar servicios y combinarlos para crear procesos de negocio ágiles y flexibles.

En términos de SOA, las partes importantes de la arquitectura son las siguientes:

- **Los Procesos** – funciones de negocio de alto nivel, a menudo abarcan aplicaciones o Líneas de Negocio.
- **Los Servicios** – Unidades modulares de la funcionalidad del negocio.
- **La Integración** – Conexión a la publicación de aplicaciones existentes y/o datos como servicios.
- **Los Sistemas Existentes** – Sistemas legados, aplicaciones comerciales, etc.
- **Los Documentos** – Unidades de información de alto nivel, así como solicitudes de trabajo o Intercambio electrónico de datos (EDI).
- **La Semántica** – El inconfundible significado de la información que es intercambiada entre procesos.
- **La Transformación** – La conversión de información de un formato o semántica a otra.
- **La Comunicación** – La habilidad de los servicios para comunicarse con otros.

SOA ofrece por sus características un bajo acoplamiento entre los servicios y una alta interoperabilidad, así como la facilidad de cambiar a modelos de negocio con la participación de entes externos a la organización (partners o proveedores).

1.2 Contexto

Financiera Independencia es una Sociedad Financiera de Objeto Múltiple (SOFOM) dedicada al otorgamiento de créditos orientados a un mercado específico. Financiera Independencia a lo largo de los años ha implementado procesos de negocio que le permiten otorgar estos créditos y contar con una cartera de clientes importante.

Financiera Independencia contaba en el 2005 con 65 sucursales y un estimado de 43,000 solicitudes al mes. Para los siguientes años se tenía una meta de poder procesar 90,000 solicitudes a partir de 70 sucursales.

La arquitectura tecnológica con que contaba Financiera Independencia estaba basada en procesos y sistemas independientes que, para hacerlos trabajar en conjunto, requerían de un esfuerzo tecnológico grande y que no cumplía con los requerimientos mencionados anteriormente. En el esquema de operación anterior se tenía un control de los recursos de red disponibles. Con el fin de mejorar la productividad y administrar mejor dichos recursos, se propuso cambiar el diseño de tareas en “polling” a un diseño de procesos basado en eventos.

Anteriormente se tenían 4 entidades de negocio que colaboraban en el proceso de la solicitud:

- **Sucursales**, se capturaban las solicitudes de crédito y se digitalizaban los documentos probatorios que acompañan la solicitud y se realizan las verificaciones físicas de las solicitudes aprobadas. Cuando la solicitud ha pasado por todo el proceso de autorización, vuelve a la sucursal para la generación del cheque y del contrato.
- **FISA (Oficina Central)**, ubicada en la ciudad de México, se tenían procesos de recolección de los datos de las sucursales, el repositorio de solicitudes para verificación física se encontraba en FISA.
- **Buró de Crédito (BNC)**, esta entidad es externa a Financiera Independencia, sin embargo, se encarga de proporcionar información crediticia acerca de los clientes que ingresan alguna solicitud de crédito.
- **Centro Operativo de León (COL)** se tiene el sistema SIAC (Sistema Integral de Análisis de Crédito), con el que se valida la información de la solicitud con respecto a los documentos

digitalizados. También se encuentra localizado el centro de llamadas en que los operadores verifican las solicitudes telefónicamente y notifican a los usuarios del otorgamiento del crédito.

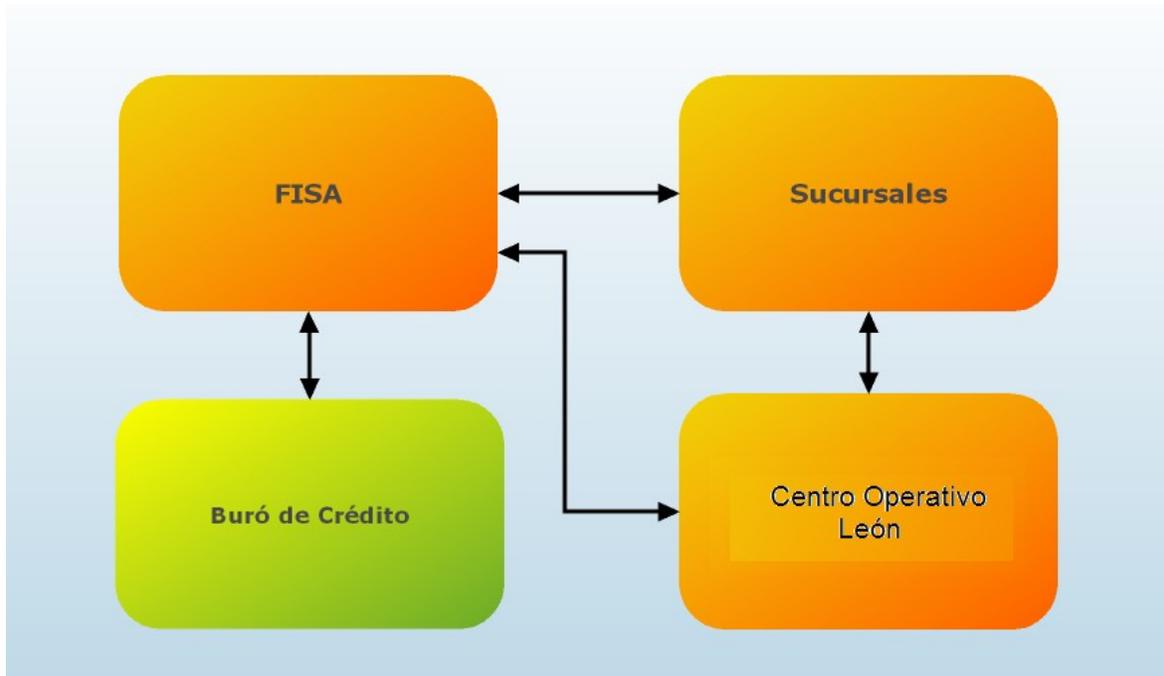


Figura 1

Las aplicaciones se ejecutaban mediante procesos tipo daemon que cada determinado tiempo replicaban los datos.

En la Figura 2 se describe la organización general anterior de los componentes del sistema.

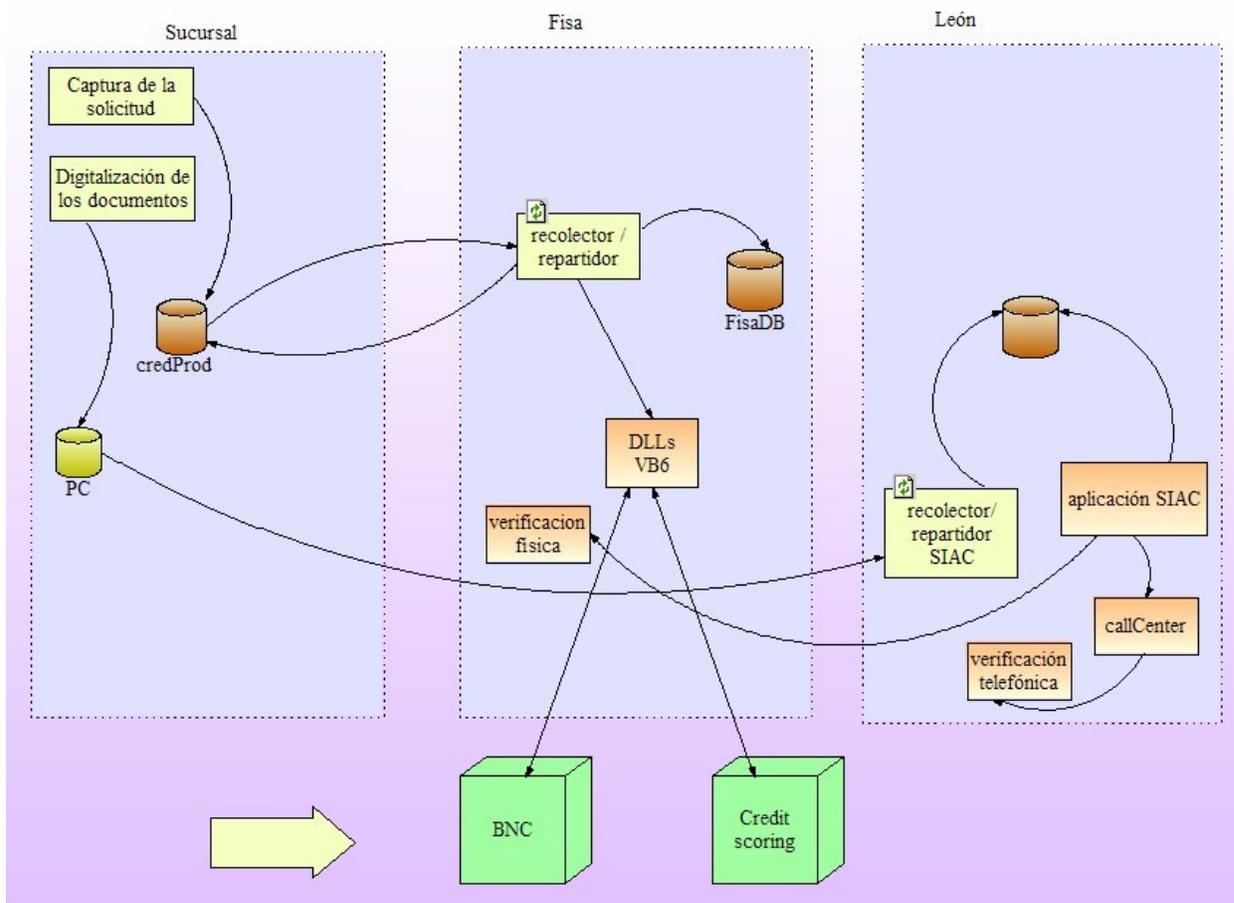


Figura 2

1.3 Vista General del Sistema

En la arquitectura anterior existían las siguientes aplicaciones

Buró Nacional de Crédito

Se cuenta con una aplicación donde mediante un componente nativo (Windows Dynamic linking library) se envían los datos de la solicitud en un formato de archivo especial. El archivo es enviado a través del canal dedicado a BNC y éste regresa la información crediticia de cada solicitud, es un mecanismo síncrono y no están implementados los Callbacks.

Credit Scoring

Se tiene una aplicación desde la que se envían los datos de información crediticia del solicitante; el sistema regresa la calificación de la persona. De la misma manera, una vez que se ha otorgado o

rechazado el crédito, se enviará información de retroalimentación para confirmar o cambiar la recomendación hecha por el Scoring.

SIAC

En la aplicación del SIAC se consultan periódicamente solicitudes disponibles a un servicio Web que regresará la siguiente, el SIAC se encarga de distribuir la tarea de captura al operador.

El operador de SIAC envía un resultado de la verificación de los documentos contra los datos capturados; envía un código de error que servirá para determinar el encargado de corregir dicho error o si se continúa el proceso.

Aplicación de digitalización de imágenes

Cada sucursal digitaliza los documentos que acompañan a una solicitud de crédito. Las imágenes digitalizadas se guardan en un repositorio de archivos local.

Aplicación de transferencia de archivos

La aplicación para transferencia de archivos es responsable de “mover” los archivos digitalizados y almacenados en el repositorio local de las sucursales hacia el repositorio central. La aplicación también es responsable de administrar el ancho de banda disponible.

Generación de Contratos en Sucursales.

Las sucursales utilizan un repositorio de datos localizado en FISA, mediante una aplicación, generan los contratos para los créditos otorgados.

Generación de cheques

Se accede a este proceso en la sucursal mediante una llamada a una aplicación en FISA, que genera un número consecutivo de cheque para el crédito otorgado. Las sucursales imprimen el cheque.

Verificación telefónica

Se registra una verificación telefónica mediante una aplicación del centro de llamadas, estas se programan mediante otra aplicación que consulta dichos registros.

Verificación física

Se registra una verificación física mediante una aplicación en FISA, en la sucursal se consultan las verificaciones y se realizan mediante una aplicación que actualiza la base de datos.

1.4 Definición de la problemática

Financiera Independencia es una empresa comprometida con un servicio de calidad para el cliente, por eso busca disminuir tanto los tiempos de respuesta como el riesgo, en cada solicitud de crédito que era ingresada. De la misma manera, está enfocada en permanecer con presencia sólida en el mercado. Para lograr lo anterior, es necesario evitar duplicidad de trabajo, optimizar recursos tanto nuevos como existentes y simplificar al máximo la administración de todos los sistemas involucrados en el Proceso de Otorgamiento de Crédito.

Es por ello que la Arquitectura de la manera descrita en éste capítulo no era suficiente para cumplir con los objetivos de Financiera Independencia.

En el escenario anterior se detectaba la siguiente problemática a resolver:

- La comunicación por muestreo periódico (polling) estaba generando períodos de inactividad, puesto que no garantizaba la entrega sincronizada de las imágenes de solicitud y los datos capturados.
- No se contaba con un orquestador central donde se pudieran incorporar nuevas reglas de negocio o afinar escenarios del flujo.
- Existían puntos donde los datos de la solicitud se podían perder. En dado caso no existía un mecanismo para notificar a las personas o sistemas involucrados de que había ocurrido un error.
- Existía una necesidad de un mejor uso de la conexión con la sucursal, ya que el ancho de banda es limitado y se comparte con otros servicios.

Es por eso que se implementó la Arquitectura expuesta en el Capítulo 2.

Capítulo 2 . - Arquitectura Actual

2.1 - El ESB de Financiera Independencia

Para la arquitectura actual, se quería llegar a un escenario en el que en el COL se tenga el control central y monitoreo de todo el proceso de la solicitud, desde ahí se orquestarían los servicios involucrados y se enviarían notificaciones a las sucursales en caso de error.

La arquitectura tecnológica que se propuso como solución para este sistema es una Arquitectura Orientada a Servicios (SOA), que permitió la interacción desacoplada y eficiente de los diferentes sistemas que requerían ser integrados. La arquitectura SOA está basada en un Bus de Servicios Empresariales (Enterprise Service Bus – ESB), que es responsable de recibir todos los eventos generados en los diferentes sistemas y enrutarlos a los sistemas interesados en dichos mensajes.

El ESB a implementarse es un canal común que cubre las necesidades de comunicación, fiabilidad, independencia y flexibilidad que requiere Financiera Independencia. El ESB permite también a Financiera Independencia la integración de nuevos sistemas a su arquitectura de manera rápida.

Financiera Independencia - ESB



Figura 3

El ESB implementado está basado en BEA WebLogic Integration 8.1 (WLI), el que permitió que los mensajes que sean depositados en el ESB sean enrutados a cada uno de los destinatarios y es responsable también de la orquestación de los servicios relacionados a cada proceso de negocio. El ESB es responsable también de la traducción de formatos de datos requerida para poder realizar el intercambio de mensajes entre los diferentes sistemas.

El ESB está basado en WebServices, que son soportados de forma nativa por WLI, y que ofrece una implementación de los estándares SOAP, WSDL (Web Services Description Language) y UDDI (Universal Description, Discovery and Integration). Los WebServices son un elemento clave para la integración de nuevos sistemas al ESB y finalmente puedan interactuar con el resto de los sistemas.

WLI provee los mecanismos de mensajería necesarios para el intercambio y entrega de mensajes de asíncronos de forma confiable, elemento indispensable para poder integrar diferentes tipos de sistemas al ESB.

La arquitectura de un ESB está centrada en un canal de comunicación central. El canal central provee servicios de entrega de mensajes, basados en estándares como SOAP, HTTP (HyperText Transfer Protocol) y Java Message Service (JMS), y típicamente está diseñado para ofrecer un nivel de procesamiento de transacciones alto (high-throughput), entrega de mensajes garantizada hacia diversos consumidores y productores.

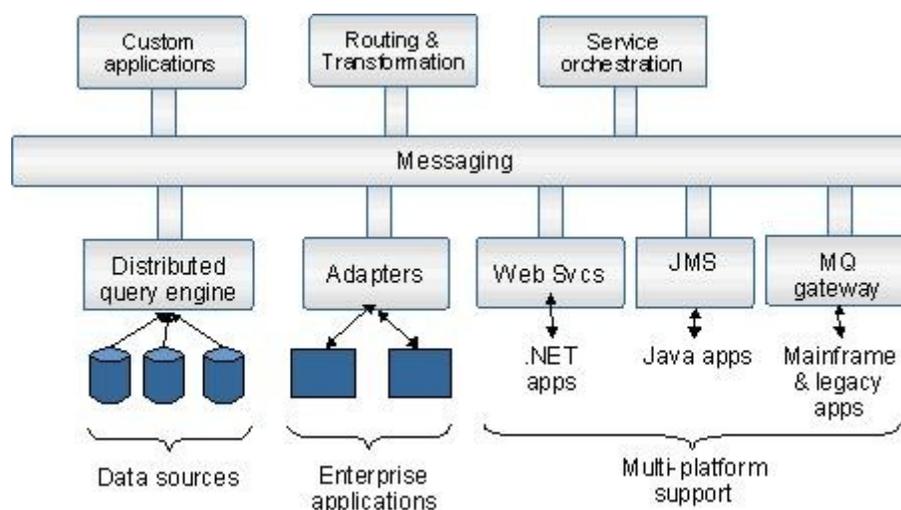


Figura 4

El ESB soporta XML (Extensible Markup Language) como tipo de datos nativo, sin embargo, permitirá manejar también otros tipos de datos. Como se muestra en la Figura 4, es posible conectar diferentes tipos de componentes al ESB:

Ruteo y Transformación: un Corredor de Mensajes (Message Broker) es un componente fundamental de un ESB. Permite el ruteo de mensajes en base a su contenido y transformación de datos, utilizando estándares como XQuery y XSLT (Extensible Stylesheet Language Transformations).

Adaptadores, típicamente implementados basados en el estándar JCA (J2EE Connector Architecture), que permiten conectarse a una gran variedad de aplicaciones empresariales.

Motores de consulta distribuidos, que típicamente están basados en XQuery o SQL (Structured Query Language), que permiten la creación de servicios de datos que abstraen la complejidad de las fuentes de datos originales.

Aplicaciones a la medida, basadas en J2EE o en Struts, que pueden conectarse al ESB para proveer una interface a los servicios empresariales.

Motores de Orquestación de Servicios (BPM), que pueden secuenciar la ejecución de los servicios y mantener el estado de los procesos de larga duración, basados en estándares como Process Definition for Java (PD4J) o Business Process Execution Language (BPEL).

En la Figura 5 se muestran los puntos mencionados anteriormente a mayor detalle.

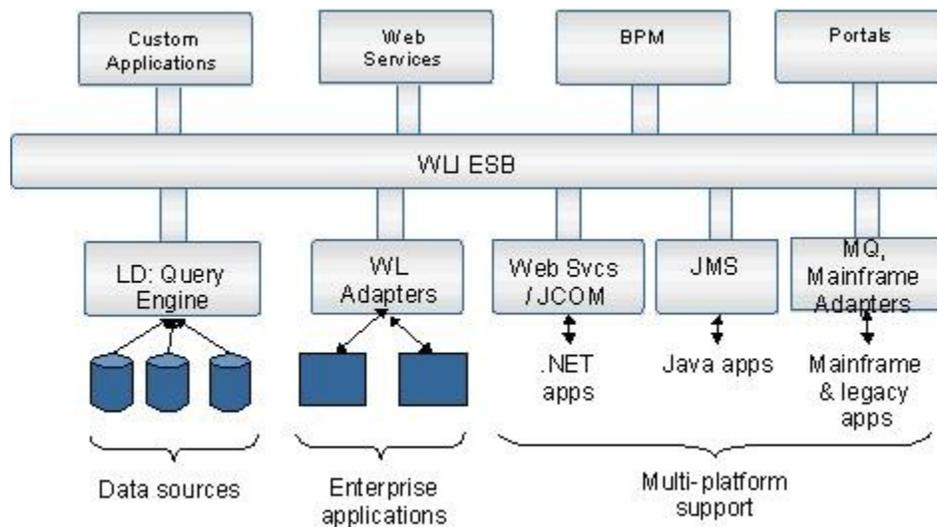


Figura 5

El núcleo del ESB es WLI, que se basa en el WLI Message Broker y en el servidor de JMS de WebLogic Server para poder soportar el alto procesamiento de mensajes, la entrega garantizada de mensajes y el ruteo de los mismos. El Message Broker de WLI utiliza canales y generadores de eventos para enviar los datos a las diferentes aplicaciones y dispara la ejecución de los servicios basados en eventos (por ejemplo, timers, web services, eventos de aplicación, etc.)

El ESB consta de de una aplicación que es desarrollada incrementalmente, basada en las capacidades de WLI para mensajes (JMS), comunicación (Web Services) y acceso a datos. La orquestación de los servicios y la comunicación con cada uno de los componentes del sistema es construida mediante los siguientes elementos:

Java Process Definition (JPD)

Permiten definir de manera gráfica los procesos de negocio y la orquestación de los servicios. De manera gráfica, es posible definir el flujo de eventos e interacciones que están involucradas en un proceso de negocios. El JPD contiene el flujo del proceso de negocio. Un JPD es modificado mediante el workshop.

Transformaciones de datos (DTF)

WebLogic Integration incluye herramientas para realizar Transformaciones de Datos, componente que es fundamental para la integración e intercambio de datos entre los diferentes sistemas.

Con WLI se pueden también generar de manera dinámica reglas de validación de los datos recibidos desde una aplicación. Es posible también aplicar funciones y búsqueda de información sobre los datos utilizando el estándar de XQuery.

Eventos

WebLogic Integration cuenta con mecanismos para generar eventos en respuesta a sucesos externos. El Bus de negocio puede generar un evento como respuesta a diversos escenarios externos, como puede ser la transferencia de un archivo, la llegada de un correo electrónico, mediante un timer se pueden generar eventos de manera periódica; así como las aplicaciones externas pueden generar eventos.

Controles Custom (JCS)

Un Control Custom es un control Java que encapsula funcionalidad reutilizable. Los Controles Custom son creados mediante un archivo JCS (Java Control Source) y un archivo Java asociado. Un Control Custom puede ser utilizado en una aplicación o empaquetado para ser utilizado en múltiples

aplicaciones. En este sentido, un Control Custom es similar a un EJB. Sin embargo, los Controles Custom tienen varias ventajas con respecto a un EJB, incluyendo las siguientes:

- Los Controles Custom pueden utilizar otros controles Java
- Los Controles Custom pueden ser extendidos
- Los Controles Custom soportan asincronía mediante callbacks

BEA ha abierto el código del framework para Controles Custom hacia la Fundación de Software Apache, haciéndolos disponibles para cualquier desarrollo Java, y no solo para el desarrollo basado en WebLogic.

Con estos componentes, se contempla un diseño del ESB construido de la siguiente manera:

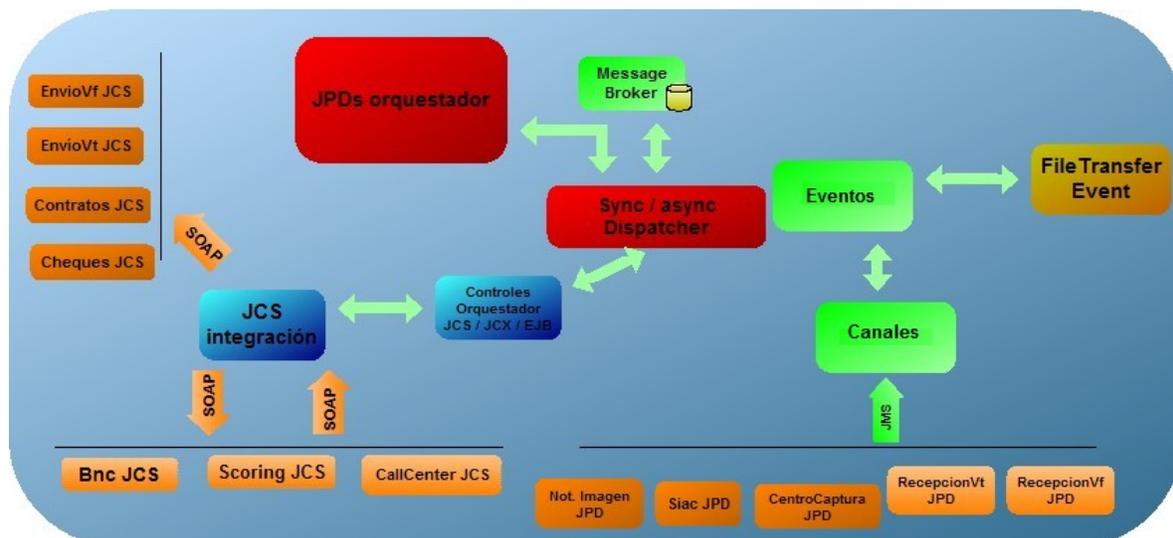


Figura 6

Uno de los principales beneficios de un ESB es que soporta la interoperabilidad de diversas plataformas. Además de soportar la tecnología de web services, WLI ofrece conectividad nativa hacia sistemas legados o tecnologías de integración propietarias, como MQSeries, CICS, .NET y aplicaciones C/C++.

Esta arquitectura aprovecha al máximo los beneficios de conectividad con que cuenta el ESB y WebLogic Integration. En la Figura 6 se muestran la forma en que se está comunicando el ESB y los subsistemas que tiene integrados.

2.2 - Subsistemas Involucrados

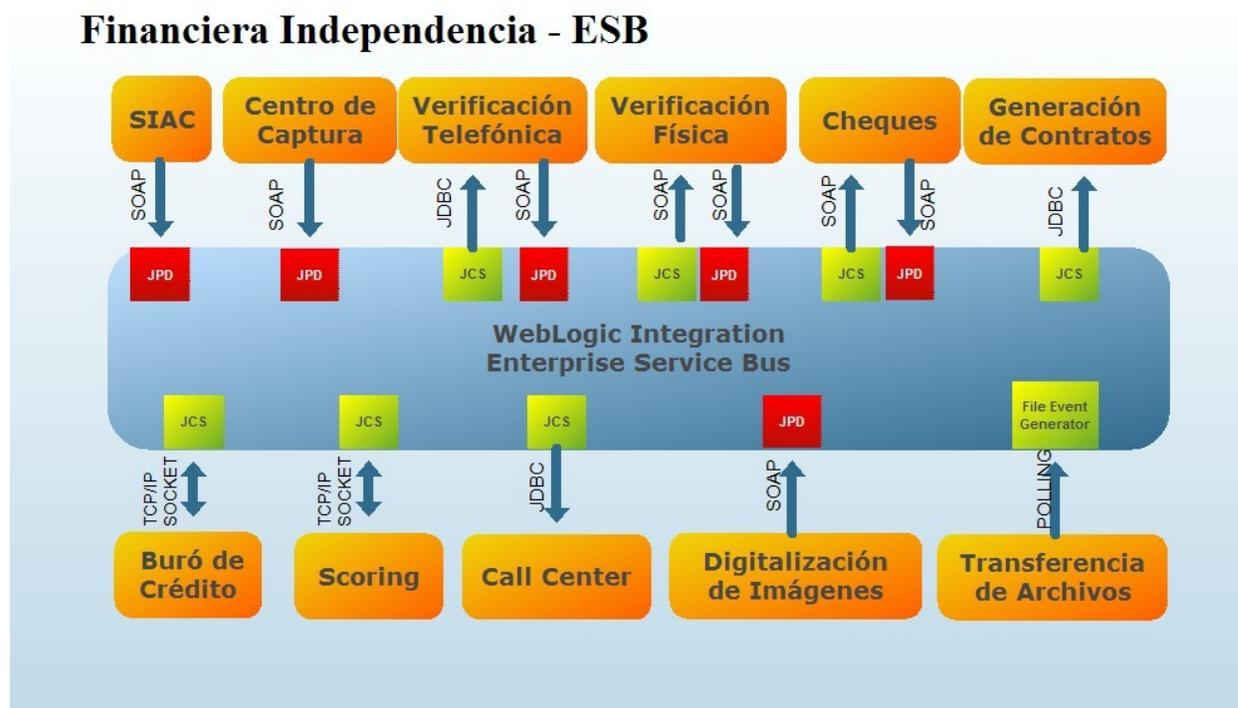


Figura 7

Como se muestra en la Figura 7, el ESB tiene interacción con diferentes subsistemas que están involucrados en los procesos de negocio de Financiera Independencia. Los mecanismos que utiliza para comunicarse con los diferentes subsistemas son los siguientes:

SIAC: La comunicación con el Sistema SIAC se realiza sólo en una dirección, es decir, el Sistema SIAC se comunicaría hacia el ESB utilizando el protocolo SOAP, sin embargo, no se cuenta con un mecanismo para que el ESB envíe acciones a SIAC. El ESB proveerá un Web Service que recibe la petición y genera los eventos relacionados.

Centro de Captura: La comunicación del Centro de Captura y el ESB se realiza también en una sola dirección. El Centro de Captura se comunica hacia el ESB utilizando el protocolo SOAP, no se cuenta con un mecanismo para que el ESB se comunique con el Centro de Captura. El ESB provee un Web Service que recibe la petición y genera los eventos relacionados.

Verificación Telefónica: El Sistema de Verificación Telefónica se comunica hacia el ESB utilizando el protocolo SOAP, no se cuenta con un mecanismo para que el ESB se comunique con el Sistema de Verificación Telefónica. El ESB provee un Web Service que recibe la petición y genera los eventos relacionados.

Verificación Física: El Sistema de Verificación Física, tiene la capacidad de recibir peticiones de forma directa, por lo que las notificaciones hacia él se realizan a través de un servicio web. El Sistema de Verificación Física implementa el servicio y detecta cuando hay una nueva notificación.

Cheques: El Sistema de Cheques requiere recibir notificaciones de parte del ESB. La comunicación con él se hace mediante un servicio web por SOAP implementado en el sistema de cheques; el ESB es cliente de dicho servicio, posteriormente la notificación es procesada por el Sistema de Cheques.

Generación de Contratos: El Sistema para Generación de Contratos requiere recibir notificaciones de parte del ESB. La comunicación con él se realiza de forma indirecta mediante una base de datos, esto es, el ESB registra la notificación en una base de datos (mediante JDBC) y posteriormente la notificación es procesada por el Sistema para Generación de Contratos.

Buró de Crédito: La comunicación con el Buró de Crédito se realizará de forma indirecta, mediante un servidor intermedio, y en una sola dirección, es decir, el ESB hace consultas de manera síncrona, el BNC responde en el mismo momento. El ESB se comunica utilizando un Socket hacia un servidor intermedio que hospeda una DLL de Windows, desarrollada por Financiera Independencia, que se conecta al Buró de Crédito y otorga una respuesta de forma síncrona con los resultados obtenidos del Buró de Crédito.

Scoring: La comunicación con el Sistema Scoring se realiza en una sola dirección, del ESB hacia Scoring. La comunicación se realizará mediante un Socket TCP(Transmission Control Protocol)/IP (Internet Protocol) que se conecta a Scoring para enviar las peticiones, que son respondidas de forma síncrona por Scoring.

Digitalización de Imágenes: El Sistema para Digitalización de Imágenes requiere notificar eventos al ESB. Para poder realizar dicha comunicación, se comunica al ESB utilizando el protocolo SOAP. El ESB cuenta con un Web Service que recibe dicha notificación y que inicia los procesos relacionados a la notificación.

Transferencia de Archivos (Valija Electrónica): El sistema para la Transferencia de Archivos es responsable de transferir los archivos digitalizados en las sucursales de Financiera Independencia. El ESB, mediante WLI, estará revisando continuamente el sistema de archivos (polling) para detectar cuando exista un nuevo archivo a procesar. Una vez detectado el nuevo archivo, se iniciarán los procesos de negocio relacionados.

La implantación del proceso de negocio se especifica con un diagrama de flujo principal.

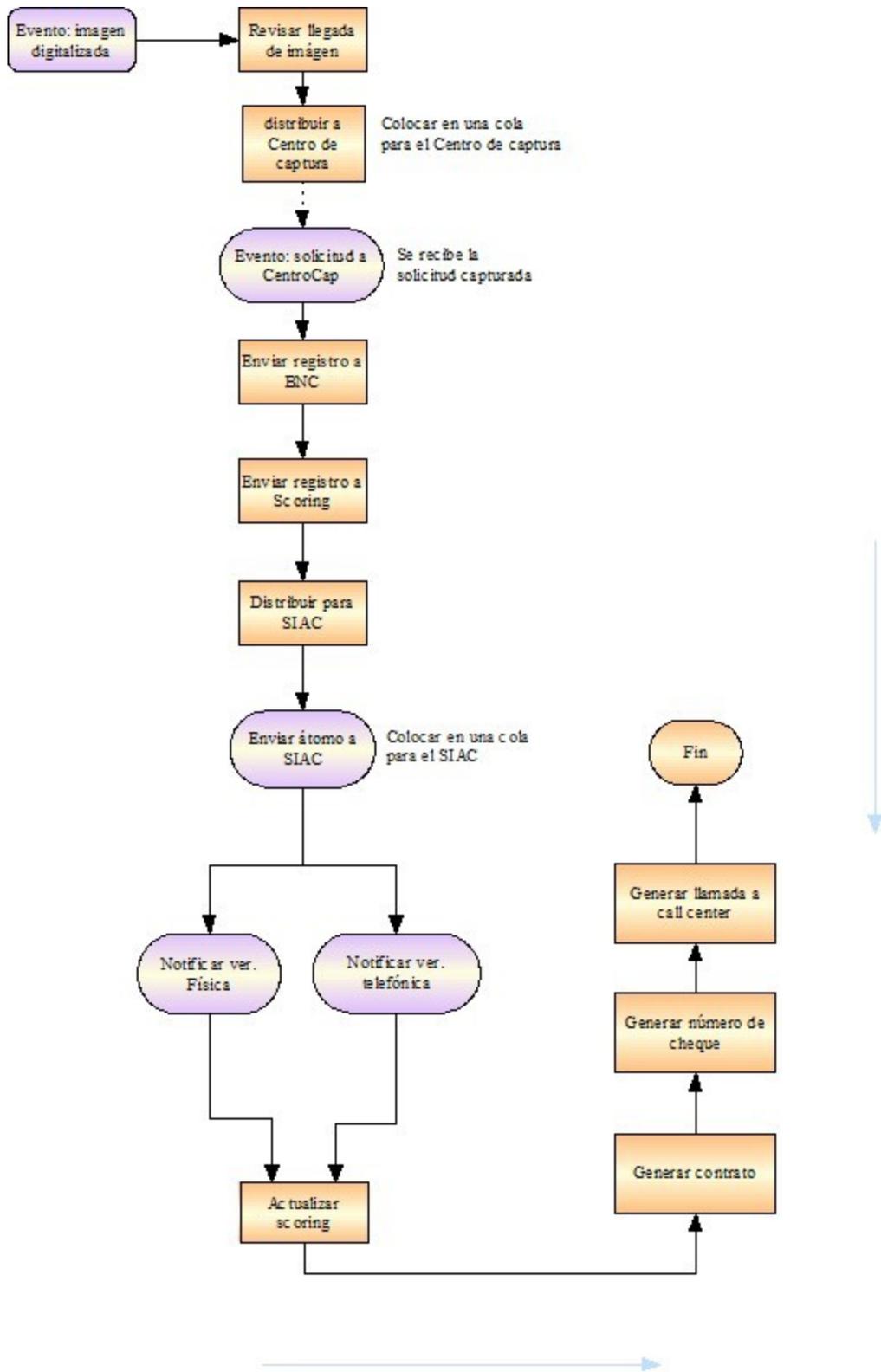


Figura 8

2.3 - Objetivos de la arquitectura

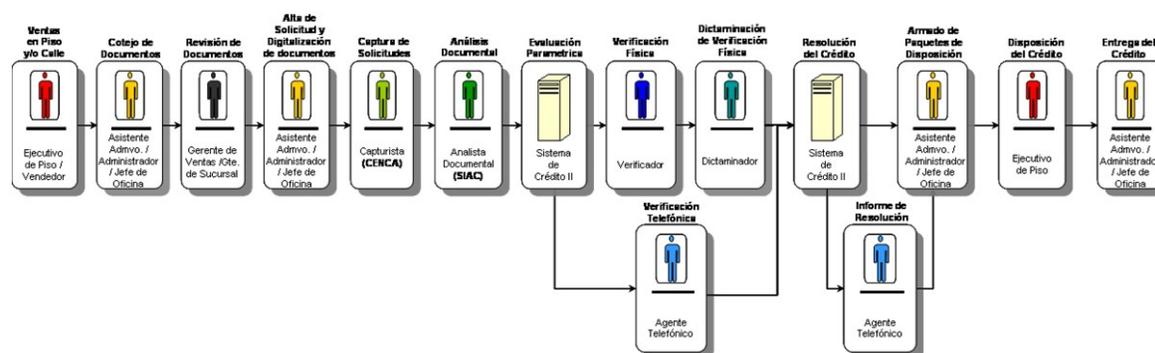
- Minimizar el tiempo de proceso de un registro, mediante las notificaciones.
- Llevar un control centralizado de la captura de las solicitudes
- Sincronizar la recepción de la imagen con la captura de los datos.
- Llevar control del estado de cada proceso y detectar errores.
- Optimizar el uso de ancho de banda en las sucursales.

El orquestador reside en un cluster de WLI bajo Linux; los sistemas nativos a integrar se encuentran sobre una plataforma Windows y están desarrollados en VB6 (Visual Basic 6), por lo que todos los componentes de integración deben ser distribuidos.

CAPÍTULO 3. – FLUJO GENERAL DE UNA SOLICITUD EN EL POC Y MI PARTICIPACIÓN PROFESIONAL

3.1 Flujo general de una solicitud en el POC

Sucursal / Oficina de Servicio



Mapa del Proceso de Otorgamiento de Crédito, utilizado para capacitación comercial.

En el proceso de otorgamiento de crédito el proceso comienza cuando el cliente es atendido por algún(a) ejecutivo(a) de piso y le pide sus documentos para iniciar el trámite del crédito, pero para nuestro interés se empezará el proceso desde la primer intervención del sistema en el proceso, esto es cuando se digitaliza la solicitud. Cabe resaltar que trataré únicamente un flujo sin errores, es decir, que no abordaré el tema de reactivación de solicitudes erróneas.

- **SUCURSAL**

El primer paso es el ingreso por parte de alguna ejecutiva de ventas de datos generales en el sistema de crédito II (Que posteriormente son transmitidos con tecnología XML), este sistema está desarrollado con Power Builder y es el que inicia el ciclo de una solicitud dentro del POC (desde el punto de vista de los sistemas). Después de esto el paso siguiente es digitalizar los documentos cotejados por el área de administración, es decir, se obtiene mediante una interfaz del sistema de crédito, una imagen digital de cada uno de los documentos del cliente y que son requeridos para la solicitud del crédito.

Una vez terminada la digitalización de los documentos, se envían los datos y las imágenes digitales al SEO (Sistema Electrónico de Originación), mediante un programa desarrollado en tecnología .NET, se encarga de mover las imágenes localmente para que un poleador (valija electrónica) mueva hasta el repositorio de León las imágenes para ser capturadas y analizadas.

Cuando hay problemas en esta y en las partes subsecuentes una de mis funciones como Ingeniero de Soporte a Producción es identificar el problema, estudiar los patrones del problema y darle solución.

Problemas en el proceso

En esta parte, los problemas que comúnmente ocurren es que los parámetros de entrada no coinciden (Porque en las tablas los valores están nulos) con los parámetros de entrada del WS del SEO, en este caso lo que hago es identificar el nodo del xml que no tiene valor y le actualizo el valor correcto.

Por ejemplo en el siguiente xml que se envía al SEO:

```
<presolicitud>
  <solicitud>19626881</solicitud>
  <sucursal>1</sucursal>
  <empresa>1</empresa>
  <personaDigitalizador></personaDigitalizador>
  <personaVendedor>010546508</personaVendedor>
  <numeroPaginas>4</numeroPaginas>
  <producto>ASP1</producto>
  <tipoSolicitud>TRIN0006</tipoSolicitud>
  <numeroCliente>00000000</numeroCliente>
  <clavePromocion>APER</clavePromocion>
  <montoPromocion>0</montoPromocion>
  <tipoDocumento>N</tipoDocumento>
  <pctFinan>100</pctFinan>
  <numeroCoacreditado>00000981</numeroCoacreditado>
  <folioSubsidio>0</folioSubsidio>
  <montoProyecto>0</montoProyecto>
  <montoSolicitado>0</montoSolicitado>
  <montoSubsidio>0</montoSubsidio>
  <montoAhorro>0</montoAhorro>
  <montoMaterial>0</montoMaterial>
  <lugarDisposicion>1</lugarDisposicion>
  <agenteTelefonico>00000000</agenteTelefonico>
</presolicitud>
```

En este ejemplo, se ve que en el elemento `personaDigitalizador`, no tiene ningún valor entonces lo que hago en este caso es actualizar el registro de esa solicitud:

```
DECLARE @documento text
SELECT @documento=replace(doctor_xml,'<personaDigitalizador></personaDigitalizador>','<personaDigitalizador>123456789</personaDigitalizador>') FROM buzon_seo WHERE documento='19626881'
UPDATE buzon_seo SET docto_xml=@documento where documento='19626881'
```

Otro de los problemas que ocurren en este proceso es que la valija electrónica que se encarga de llevarse las imágenes a León, se queda detenida por alguna razón. Para solucionar esto, reinicio el servicio para que continúe enviando las imágenes.

- **SEO (Sistema Electrónico de Origenación)**

A partir de este momento la solicitud entra en interfaces llamadas “átomos” las cuales son sistemas y procesos independientes, sin embargo llevan un orden específico de ejecución, según el tipo de solicitud del cual se trate (nueva, renovación, etc), dicho orden es orquestado por el `weblogic integration`.

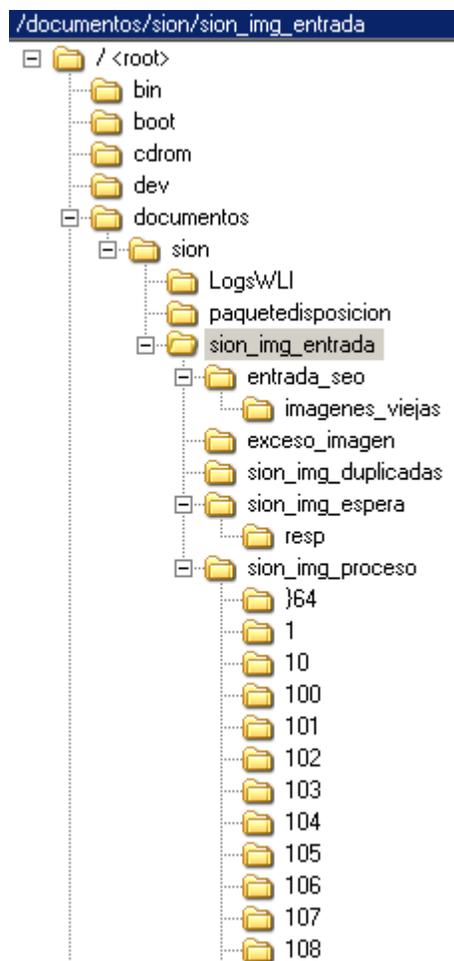
En primer lugar tenemos el átomo de imagen. Las imágenes llegan a un repositorio en el COL (Centro Operativo León), entonces la acción de este proceso es registrar en base de datos el nombre de la imagen para que después sea encontrada por los demás átomos que consultan dichas imágenes. Una vez que está registrada la imagen, este proceso notifica al SEO que el proceso está completado, el SEO guarda en sus propias tablas el status de la solicitud y envía la petición al siguiente átomo el cual es el de Central de Captura.

Problemas en el proceso

Este proceso es uno de los más problemáticos ya que intervienen varios factores para la resolución del mismo. El nombre de la imagen debe ser guardado en las tablas del SIAC para que la aplicación local pueda localizar físicamente la imagen, por lo tanto en una de las etapas (Por dentro del proceso), el SEO toma el nombre de la imagen y la manda a un Web Service el cual se encarga de

registrar la imagen. Pero puede ser que la imagen ya no se encuentre en la carpeta de donde el SEO la toma debido a que un cron de depuración de imágenes la movió por haber excedido su tiempo, dejando el proceso pendiente. Lo que hago en este caso es buscar la imagen en la carpeta de depuración y regresarla a la carpeta de operación para que el SEO la pueda tomar y registrar su nombre.

La estructura de las carpetas dentro del servidor es la siguiente:



Estructura de Carpetas dentro del Servidor FISAIMG

Donde las importantes son las siguientes:

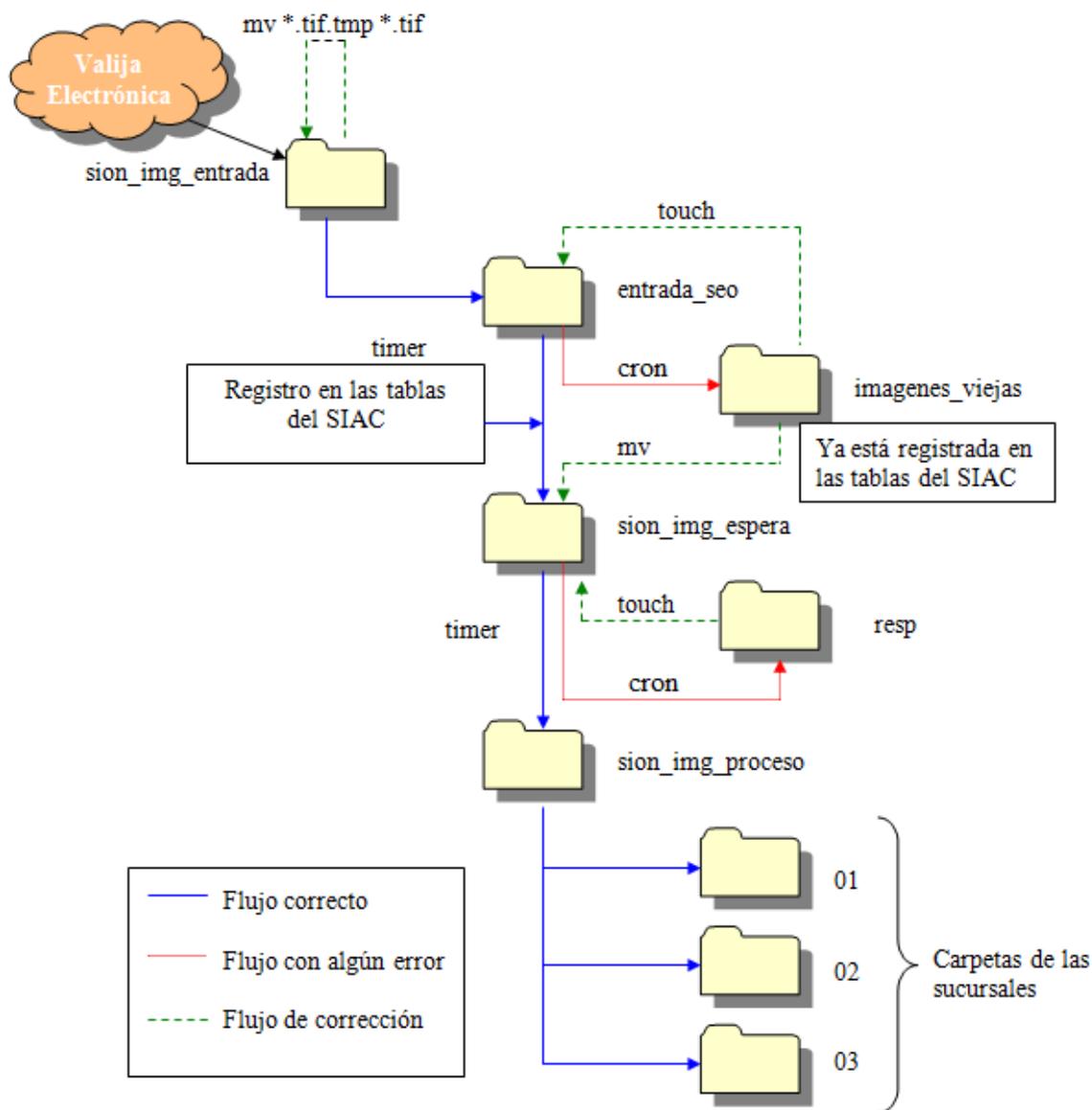
- **Sion_img_entrada:** aquí llegan todas las imágenes que trae la valija electrónica
- **Entrada_seo:** Un cron dentro del servidor traslada las imágenes a esta carpeta.
 - *Imagenes_viejas:* llegan las imágenes que excedieron su tiempo de 5 minutos en la carpeta entrada_seo
- **Sion_img_espera:** aquí llegan las imágenes que ya fueron registradas en las tablas del SIAC y están en espera a pasar a la siguiente carpeta.
 - Resp: si excedieron el tiempo (5 minutos) en la carpeta sion_img_espera llegan aquí las imágenes
- **Sion_img_proceso:** dentro de esta carpeta llegan las imágenes que están en proceso.
 - 1 : contiene las imágenes de las solicitudes que están en proceso y son de la sucursal 1.
 - 2: contiene las imágenes de las solicitudes que están en proceso y son de la sucursal 2.
 - 3

El flujo de las imágenes en el servidor es el siguiente:

1. La valija electrónica trae las imágenes digitalizadas de las solicitudes desde las sucursales hasta sion_img_entrada.
2. Una vez en sion_img_entrada un cron las traslada a la carpeta de entrada_seo y cambia la extensión de tmp a tif. Si esto no ocurre debe a:
 - a. El proceso no cambio la extensión tmp de la imagen a tif, por lo tanto no es trasladada a la siguiente carpeta.
3. Una vez en entrada_seo un timer las mueve a sion_img_espera pero antes las registra en las tablas del SIAC. Si esto no ocurre debe a :
 - a. La solicitud excedió el tiempo de 5 minutos en esta carpeta y es movida por un cron de respaldo a imágenes_viejas.
4. Después llegan las imágenes a la carpeta sion_img_espera donde esperan al timer a ser recogidas para ser enviadas a cenca y seguir en el POC. Si esto no ocurre, son movidas por exceso de tiempo a la carpeta resp.

- Una vez que cambiaron al átomo de captura las imágenes son movidas a la carpeta de su sucursal correspondiente (identificada por número) dentro de la carpeta sion_img_proceso.

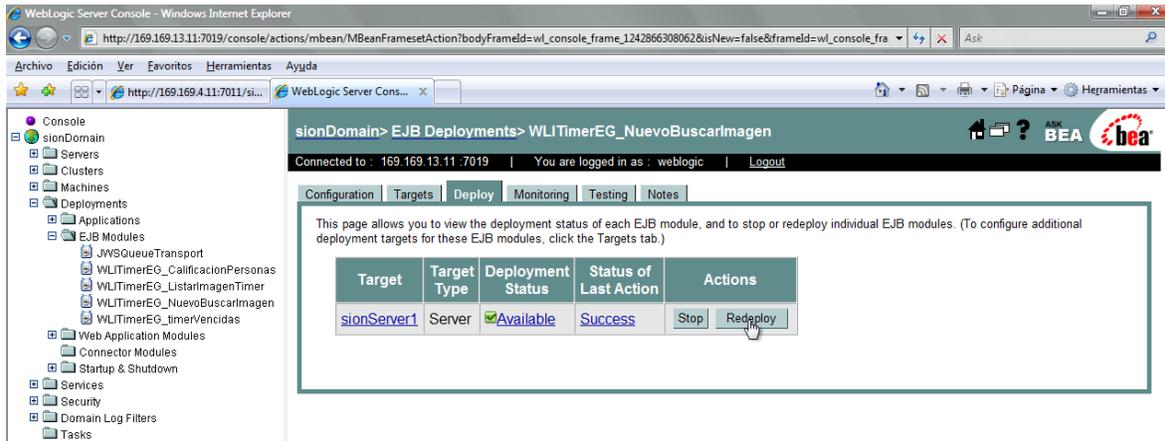
A continuación se muestra gráficamente el flujo de las imágenes dentro del servidor.



Explicación gráfica del flujo de las imágenes dentro del servidor, con tres flujos, el correcto, con errores y el flujo de corrección.

Otro problema común en este proceso son los “timers” que ejecuta el SEO para registrar las imágenes en el SIAC y para cambiarlas entre carpetas, ya que a veces no están bien

“desplegados” (deployed) en el servidor, es decir, que no están activos. Para solucionar esto, desde la consola de administración del SEO activo los timers haciendo un redesplicue (redployment). De esta manera se activan y empiezan a procesar las imágenes.



WebLogic Server Console, Ejemplo del Redeploy del timer BuscarImagenDigitalizada.

- **CENCA (Centro Nacional De Captura)**

El SEO como orquestador hace la petición al Web Service de CENCA enviando la ruta de la imagen y otros datos generales de la solicitud. Una vez que los datos son recibidos, la aplicación de CENCA reparte las solicitudes según su antigüedad a las PC's clientes que residen en el Centro Operativo León donde capturistas digitalizan la información escrita en las imágenes de los documentos del cliente. Cuando se termina de capturar una solicitud, el cliente envía una petición por la próxima solicitud a capturar. Una vez que los datos fueron capturados, una parte de la aplicación de CENCA llamada "FISA exporter requests" envía los datos capturados al SEO por medio de un XML el cual contiene todos los datos del cliente (propios, referencias, etc) y de la solicitud. El seo recibe los datos y envía la solicitud al siguiente átomo el cuál es Cálculo de Crédito.

Problemas en el proceso

En este proceso el problema más general, son los caracteres "raros" en el documento generado en por los capturistas, caracteres que no son soportados por la codificación de caracteres en el XML, el carácter más común es el "&" cuando esto sucede en las tablas de CENCA se queda pendiente por exportar (enviar al seo) la solicitud ya capturada. Lo que hago para corregir esto es sustituir el

carácter del XML generado y guardarlo nuevamente en las tablas de CENCA, de esta manera, se vuelve a exportar la solicitud y es recibida por el seo.

A continuación escribo un ejemplo de un XML con carácter raro:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<recibirSolicitudScc xmlns="sion.fisa.solicitudFormal">
<solicitud>
  <idSolicitud>17980896</idSolicitud>
  <datosSolicitante>
    <curp errorCampo="" modificado="false" ></curp>
    <numDependientes errorCampo="" modificado="false" >1</numDependientes>
    <entidadFederativa errorCampo="" modificado="false"
>MOR</entidadFederativa>
    <antiguedadAnios errorCampo="" modificado="false" >5</antiguedadAnios>
    <antiguedadMeses errorCampo="" modificado="false" >0</antiguedadMeses>
    <ladaTelefonoParticular errorCampo="" modificado="false"
>777</ladaTelefonoParticular>
    <ladaTelefonoCelular errorCampo="" modificado="false"
>777</ladaTelefonoCelular>
    <ladaTelefonoRecados errorCampo="" modificado="false"
>777</ladaTelefonoRecados>
    <nombre errorCampo="" modificado="false" >NOHEMI </nombre>
    <paterno errorCampo="" modificado="false" >OLIVER </paterno>
    <materno errorCampo="" modificado="false" >ALDERETE </materno>
    <rfc errorCampo="" modificado="false" >OIAN860612</rfc>
    <fechaNacimiento errorCampo="" modificado="false"
>1986-06-12T12:00:00.0000000-05:00</fechaNacimiento>
    <correoElectronico errorCampo="" modificado="false"
></correoElectronico>
    <sexo errorCampo="" modificado="false" >F</sexo>
    <nacionalidad errorCampo="" modificado="false" >M</nacionalidad>
    <pais errorCampo="" modificado="true" >MX</pais>
    <estadoCivil errorCampo="" modificado="false" >C</estadoCivil>
    <tipoVivienda errorCampo="" modificado="false" >F</tipoVivienda>
    <domicilio errorCampo="" modificado="false" >ISIDRO FABELA 5</domicilio>
    <entreCalles errorCampo="" modificado="false" >CALLE DE LA LUZ Y 2DA PRIV LA LUZ
</entreCalles>
    <cp errorCampo="" modificado="false" >62450</cp><delegacionMpio
errorCampo="" modificado="false" >CUERNAVACA</delegacionMpio>
    <ciudad errorCampo="" modificado="false" >CUERNAVACA</ciudad>
    <colonia errorCampo="" modificado="false" >CHAPULTEPEC</colonia>
    <telefonoParticular errorCampo="" modificado="false"
></telefonoParticular>
    <telefonoCelular errorCampo="" modificado="false" ></telefonoCelular>
    <telefonoRecados errorCampo="" modificado="false"
>3158156</telefonoRecados>
    <recaderoNombre errorCampo="" modificado="false" >OSCAR
</recaderoNombre>
    <recaderoPaterno errorCampo="" modificado="false" >ARCE
</recaderoPaterno>
    <recaderoMaterno errorCampo="" modificado="false" >HERNANDEZ
</recaderoMaterno>
    <recaderoRelacionCliente errorCampo="" modificado="false"
>20</recaderoRelacionCliente>
```

```

        <tipoIdentificacion errorCampo="" modificado="false"
>CIFE</tipoIdentificacion>
        <numeroIdentificacion errorCampo="" modificado="false"
>0222098728065</numeroIdentificacion>
        <emisionIdentificacion errorCampo="" modificado="false"
>00</emisionIdentificacion>
    </datosSolicitante>
    <datosEmpleo>
        <idEmpresa errorCampo="" modificado="false" >0</idEmpresa>
        <lada errorCampo="" modificado="false" >777</lada>
        <entidadFederativa errorCampo="" modificado="false"
>MOR</entidadFederativa>
        <tipoTrabajo errorCampo="" modificado="false" >EMP</tipoTrabajo>
        <centroTrabajo errorCampo="" modificado="false" >C & A </centroTrabajo>
        <razonSocialEmp errorCampo="" modificado="false" ></razonSocialEmp>
        <rfc errorCampo="" modificado="false" ></rfc>
        <giro errorCampo="" modificado="false" >FFW</giro>
        <puesto errorCampo="" modificado="false" >1K6</puesto>
        <fechaIngreso errorCampo="" modificado="false"
>2002-02-03T12:00:00.0000000-06:00</fechaIngreso>
        <ustedCotizaEn errorCampo="" modificado="false" >M</ustedCotizaEn>
        <sueldoMensual errorCampo="" modificado="false" >6000</sueldoMensual>
        <nss errorCampo="" modificado="false" ></nss>
        <frecuenciaCobro errorCampo="" modificado="false" >Q</frecuenciaCobro>
        <telefono errorCampo="" modificado="false" ></telefono>
        <extension errorCampo="" modificado="false" ></extension>
        <domicilio errorCampo="" modificado="false" >VICENTE GUERRERO
760</domicilio>
        <entreCalles errorCampo="" modificado="false" >X SAM</entreCalles>
        <cp errorCampo="" modificado="false" >62320</cp>
        <delegacion errorCampo="" modificado="false" >CUERNAVACA</delegacion>
        <ciudad errorCampo="" modificado="false" >CUERNAVACA</ciudad>
        <colonia errorCampo="NCAT" modificado="false" ></colonia>
        <nombreJefe errorCampo="" modificado="false" >RAFAEL </nombreJefe>
        <apellidoPaternoJefe errorCampo="" modificado="false" >ARCE
</apellidoPaternoJefe>
        <apellidoMaternoJefe errorCampo="" modificado="false" >CRUZ
</apellidoMaternoJefe>
        <puestoJefe errorCampo="" modificado="false" >SUPERVISOR </puestoJefe>
    </datosEmpleo>
    <datosConyuge>
        <nombre errorCampo="" modificado="false" >JOSE </nombre>
        <paterno errorCampo="" modificado="false" >ORTIZ </paterno>
        <materno errorCampo="" modificado="false" >CARRANZA </materno>
        <sexo errorCampo="" modificado="false" >M</sexo>
    </datosConyuge>
    <referencia1>
        <ladaTelefono errorCampo="" modificado="false" >777</ladaTelefono>
        <nombre errorCampo="" modificado="false" >RICARDO </nombre>
        <paterno errorCampo="" modificado="false" >MARTINEZ </paterno>
        <materno errorCampo="" modificado="false" >CRUZ </materno>
        <telefono errorCampo="" modificado="false" >3163830</telefono>
        <extension errorCampo="" modificado="false" ></extension>
        <casaoempleo errorCampo="" modificado="false" >C</casaoempleo>
        <relacionCliente errorCampo="" modificado="false" >16</relacionCliente>
    </referencia1>
    <referencia2>
        <ladaTelefono errorCampo="" modificado="false" >777</ladaTelefono>
        <nombre errorCampo="" modificado="false" >GLORIA</nombre>

```

```

    <paterno errorCampo="" modificado="false" >ARCE </paterno>
    <materno errorCampo="" modificado="false" >AVILA </materno>
    <telefono errorCampo="" modificado="false" >5120294</telefono>
    <extension errorCampo="" modificado="false" ></extension>
    <casaoempleo errorCampo="" modificado="false" >C</casaoempleo>
    <relacionCliente errorCampo="" modificado="false" >21</relacionCliente>
  </referencia2>
  <creditoSolicitado>
    <montoSolicitado errorCampo="" modificado="false"
>6000</montoSolicitado>
    <plazo errorCampo="" modificado="false" >44</plazo>
    <diaPago errorCampo="" modificado="false" >3</diaPago>
    <porcentajeFinanciamiento errorCampo="" modificado="false"
>100</porcentajeFinanciamiento>
    <destinoCredito errorCampo="" modificado="false" >P</destinoCredito>
    <lugarDisposicion errorCampo="" modificado="false" >1</lugarDisposicion>
    <origenSolicitud errorCampo="" modificado="false" >L</origenSolicitud>
    <esPromocion errorCampo="" modificado="false" >N</esPromocion>
    <contratoRecomienda errorCampo="" modificado="false"
></contratoRecomienda>
  </creditoSolicitado>
  <preguntas>
    <funcionarioPublico errorCampo="" modificado="false"
>N</funcionarioPublico>
    <relacionFuncionario errorCampo="" modificado="false"
>N</relacionFuncionario>
  </preguntas>
  <capturista>720526207</capturista>
  <comentarios></comentarios>
</solicitud>
</recibirSolicitudScs>

```

En este caso por ejemplo en el campo “centro de trabajo” existe un carácter ‘&’ el cual debe ser reemplazado para que la aplicación pueda enviar la captura nuevamente al SEO.

```

DECLARE @documento text
SELECT @documento=replace(documentogenerado,'&','&amp;') FROM datosregistrados WHERE
folio='17980896'
UPDATE datosregistrados SET documentogenerado =@documento WHERE folio='17980896'

```

De esta forma queda resuelto el problema del carácter no soportado por la aplicación en XML generado y puede enviar la captura al SEO.

- **CÁLCULO DE CRÉDITO**

EL átomo de cálculo de crédito es sencillo, pero es una parte crucial del proceso, ya que en él se calculan todos los montos del crédito según el producto. Los parámetros que recibe el stored procedure son los datos de la solicitud que ya fueron capturados previamente. Este átomo se ejecuta del lado del seo, ya que es un stored procedure residente en la base de datos del SEO. Este proceso es síncrono y una vez que el seo recibe la respuesta envía la solicitud al siguiente átomo que es el de validación de políticas de crédito.

Problemas en el proceso

En este proceso lo que les sucede a las solicitudes es que faltó algún dato de capturar en el proceso anterior y por tal motivo cuando se llama el stored procedure de cálculo de crédito no puede dar los resultados requeridos ya que hacen falta parámetros. En este proceso hago un “debug” de la solicitud para ver cuál es el campo al que le hace falta el valor o tiene un valor erróneo. Una vez actualizando dicho campo, la solicitud resuelve el proceso sin problemas.

Parameter_name	Type
@p_solicitud	varchar
@p_monto	float
@p_monto_pago	float
@p_monto_ult_pago	float
@p_fecha_prim_venc	fecha_type
@p_fecha_prox_pago	fecha_type
@p_gastos_admon	float
@p_comision_apertura	float
@p_seguro	float
@p_monto_pcd	float
@p_cod_monto_2	Char
@p_monto_2	Float
@p_pct_iva_gastos_admon	Float
@p_pct_iva_comision	Float
@p_pct_iva_seguro	Float
@p_pct_iva_pcd	Float
@p_pct_iva_monto_2	Float
@p_tasa	Float
@p_codigo	Char
@p_comentario	Varchar
@p_fecha_esp_liq	fecha_type

Estos son los parámetros que recibe el stored procedure *calcula_credito* el cual cuando hace falta algún valor se corrige con una actualización a la solicitud parecida a la siguiente:

```
UPDATE solicitudes SET monto=3500 WHERE solicitud='12345678'
```

En este caso se corrige la solicitud en el campo monto para que pueda seguir su recorrido hacia validación de políticas de crédito.

- **VALIDACIÓN DE POLÍTICAS DE CRÉDITO**

Este átomo es igual síncrono ya que el SEO por una parte ejecuta un stored procedure residente en su propia base de datos y después consume un webservice publicado en el SIC (Sistema Integral de Crédito) el cual verifica los teléfonos de la referencias así como el número de contratos que ha tenido el cliente, que el cliente se encuentre dentro de la edad, que el domicilio del cliente se encuentre dentro de la zona autorizada para la sucursal, entre otras cosas. Una vez que el SEO recibe la respuesta del web service, dirige el flujo de la solicitud hacia el siguiente átomo el cual es SIAC.

Problemas en el proceso

Al igual que el proceso anterior el problema más común con el que me encuentro es la falta de valores en los parámetros o valores erróneos que se pasan al stored procedure. Para corregir esto hago un “debug” de la solicitud y completo o corrijo los valores que se están pasando al stored procedure.

Parameter_name	Type
@p_solicitud	char
@p_fecha_verificacion	fecha_type
@p_cliente	char
@p_sucursal	int
@p_no_avales	int
@p_aval_1	char
@p_tipo_aval	char
@p_producto	char
@p_tipo_credito	char
@p_efectivo	money
@p_plazo	int
@p_frecuencia	char
@p_monto_pago	money
@p_monto_pago_cp	money
@p_pct_finan	float
@p_contrato_ant	no_contrato_type
@p_vendedor	no_persona_type

@p_tipo_codeudor_1	char
@p_segmento	char
@p_tipo_producto	char
@p_prospectos	char
@fecha_contrato_ant	datetime
@p_status_contrato_ant	char
@p_monto_liq_ant	money
@p_producto_ant	char
@p_monto_promocion	money
@p_monto_pago_ant	money
@contrato_recomienda	char
@p_fecha_solicitud	fecha_type
@p_formato	char
@p_rfc	rfc_type
@p_fecha_nac	fecha_type
@p_calle	varchar
@p_calle_emp	varchar
@p_cp	codigo_postal_type
@p_ciudad	ciudad_type
@p_cp_emp	codigo_postal_type
@p_ciudad_emp	ciudad_type
@p_puesto_emp	char
@p_negocio_propio	char
@p_fecha_ing_emp	fecha_type
@p_ingresos	money
@p_calificacion	int
@p_rfc_calculado	rfc_type
@p_municipio	varchar
@p_municipio_emp	varchar
@p_entidad	char
@p_entidad_emp	char
@p_pensionado	char
@p_asalariado	char
@p_patrimonio	char
@p_p1	char
@p_p2	char
@p_numSolicitudes	int
@p_existe_persona	int
@p_colonia	varchar
@p_colonia_emp	varchar
@p_cp_obra	codigo_postal_type
@p_colonia_obra	varchar

@dias_atraso	fecha_type
@retorno	char
@comentario	varchar
@tipo_analisis	char
@tipo_respuesta_politicas	int
@tipo_telefono_inactivar_dir e	char
@tipo_telefono_inactivar_rec a	char
@tipo_telefono_inactivar_cel u	char
@tipo_telefono_inactivar_emp l	char
@tipo_telefono_inactivar_ref 1	char
@tipo_telefono_inactivar_ref 2	char
@sucursal_cobranza	Int

Es común en este proceso que el campo @p_cp (código postal) venga en blanco y esto ocasiona que el stored procedure envíe errónea la solicitud con el comentario “Domicilio fuera de la zona autorizada”, para corregir esto, ejecuto la siguiente sentencia SQL en la base de datos del SEO y la solicitud continúa su camino al siguiente átomo.

```
UPDATE personas SET cp=11800 WHERE persona='123456789'
```

- **SIAC**

Una vez que el seo hace la llamada al web service del SIAC (Sistema de Análisis Central), la aplicación del SIAC funciona en forma similar a la de CENCA, se reparten las solicitudes tomando en cuenta la antigüedad de la misma en la base de datos para ser analizadas por un analista documental. El analista utiliza el sistema SIAC para verificar que la información digitalizada con la capturada por el CENCA sea la misma, además de verificar que esté en forma con respecto a las políticas dictaminadas por la empresa. Por ejemplo, que tenga la capacidad de pago acorde al producto que solicita, que no sea una actividad indeseable, etc.

El analista al final decide si la revisión es aprobada, cuando no se encontró ninguna observación en la revisión; autorizada condicionada, cuando la aprueba pero fuerza a que se aplique la verificación o bien con algún código de error.

Con un servicio de windows que radica en el SIAC se envía las solicitudes ya revisadas al SEO, para ello se ayuda del campo estado_envio y toma las que se encuentran con valor S (pendiente de salir), adicionalmente le manda una plantilla con los campos donde se localizó los errores, para que el SEO posteriormente las procese y se las envíe a CENCA.

Problemas en el proceso

En este proceso el problema más común es que las solicitudes tienen valor de exportado "E" sin embargo los datos no fueron recibidos por el SEO. Para este caso lo que hago es actualizar el valor del registro a pendiente por enviar ("S") y se vuelve a enviar para que el SEO reciba la respuesta del análisis.

```
UPDATE siac_bandeja SET estado_envio='S' WHERE solicitud='12345678' and estado_envio='E'
```

- **CONSULTA A BURÓ DE CRÉDITO**

SEO llama a un web service con los datos generales de la solicitud, el web service guarda esa información en una base de buró de crédito propia de FISA y devuelve el control al SEO.

Un servicio de windows se encarga de monitorear la información almacenada en tabla indicada el punto anterior y si detecta nuevos registros entonces realiza la consulta al buró nacional de crédito y la respuesta la almacena en la misma base de datos local.

Otro servicio de windows exportará los registros hacia el SEO con un llamado a un web service tomando los datos de la base local que cumplan con el status de pendiente.

El SEO recibe la información del web service y guarda en la base de datos local la respuesta recibida, esto es; una S cuando se encontró a la persona en buró de crédito y una N en caso contrario.

Problemas en el Proceso

El problema más común con el que me encuentro en este proceso es, que el servicio de Windows que reside en el servidor de aplicaciones, se queda detenido debido a un consumo excesivo de memoria por otras aplicaciones, en este caso resuelvo el problema reiniciando el servicio de Windows encargado de la exportación de solicitudes pendientes.

Otro de los problemas que ocurren frecuentemente es que en la consulta a buró se manda el XML pero falta algún dato, como por ejemplo:

```
<envioburo xmlns="com/fisa/sion/schema/servicioburo">
  <Solicitud>16743205</Solicitud>
  <IdAtomo>47020742</IdAtomo>
  <Origen>SE0</Origen>
  <Sucursal>89</Sucursal>
  <Persona>722125972</Persona>
  <Prioridad>2</Prioridad>
  <Nombre>PAULO CESAR DE JESUS</Nombre>
  <paterno>ALCARAZ</paterno>
  <materno>LOPEZ</materno>
  <FechaNacimiento>1977-09-05T00:00:00</FechaNacimiento>
  <Rfc>AALP770905</Rfc>
  <Sexo>M</Sexo>
  <EstadoCivil>C</EstadoCivil>
  <Nacionalidad>M</Nacionalidad>
  <Calle>C RTNO SANTA MARIA 6 D</Calle>
  <Colonia>LAS AMERICAS</Colonia>
  <Cp>99030</Cp>
  <Ciudad></Ciudad>
  <Entidad>ZAC</Entidad>
  <Municipio>FRESNILLO</Municipio>
  <Importe>4800.0000</Importe>
  <Segmento>INFO</Segmento>
  <TipoProducto>TRAD</TipoProducto>
  <Categoria>POPN</Categoria>
</envioburo>
```

En este ejemplo, vemos que el elemento “Ciudad” en el xml no tiene valor asignado y provoca que la solicitud no pueda hacer la consulta a buró. El paso a seguir es completar el Valor en la tabla de personas en el SEO y volver a enviar la solicitud a consulta con todos los datos.

```
UPDATE personas SET ciudad='FRESNILLO' WHERE persona='722125972'
```

Otro de los problemas que se dan, pero con menos frecuencia, son las causas externas (el enlace a buró no está disponible, por ejemplo), cuando esto sucede el seo tiene mecanismos de reintentos, los cuales congelan en tablas locales la parte de código que estaba ejecutando, generando “tareas” para que posteriormente desde una aplicación web se puedan poner nuevamente en marcha y retome la ejecución de dichos procesos desde donde se quedó, de esta forma pueden continuar al siguiente proceso las solicitudes.

- **CONSULTAR CREDIT SCORE.**

Este átomo tiene como objetivo devolver un puntaje del cliente con base en la información que se le ha recolectado hasta el momento; como es, sus datos personales y de empleo, datos de buró de crédito, datos del producto que solicita y datos generales.

SEO llama a un stored procedure en la base de datos local para obtener datos del cliente como sexo, edad, estado civil, tipo vivienda, número de dependientes, antigüedad en el empleo, tipo de actividad, y todos los datos que obtuvo del buró de crédito. Después llama a un web service, al cual se le enviará la información recolectada anteriormente, esta llamada es síncrona.

El web service recibe la información del SEO y llama a una dll desarrollada por FISA, esta librería en general se encarga de armar la trama correspondiente y de hacer la conversión de los caracteres de control que requiere la siguiente dll, de igual forma cuando recibe la respuesta.

Problemas en el proceso

El problema más común en este átomo es la aplicación que puntea al solicitante, ya que son dll's que se llaman entre sí. Los logs de la aplicación (Fair Isaac) se llenan y lo que hago es limpiar los logs, es decir, se respaldo los logs antiguos y los sustituyo por unos vacíos.

Después de hacer esto lo que hago es reiniciar la aplicación con las herramientas de la misma y de ésta forma continúa atendiendo peticiones del web service.

A continuación se muestran las pantallas para el reinicio del Fair Isaac:

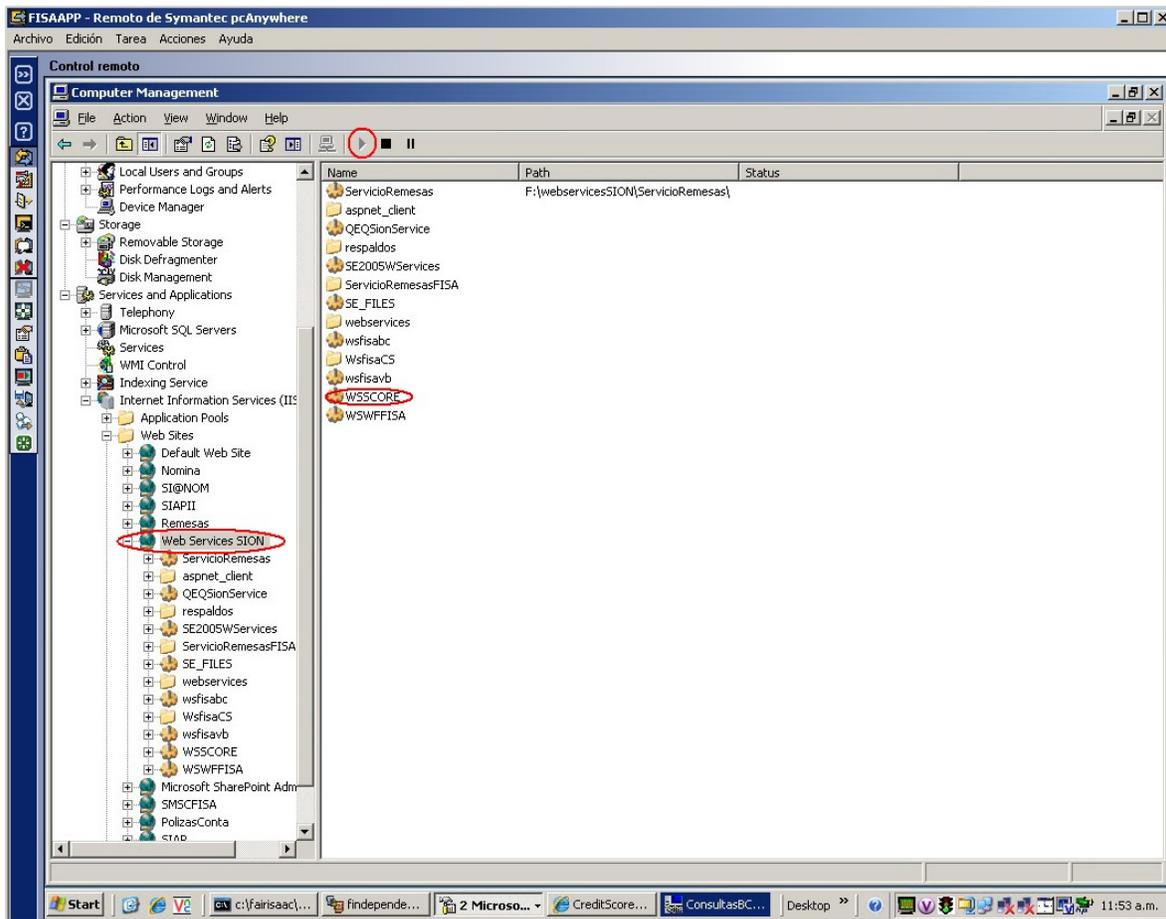


Figura 9. Reinicio del Servicio en el servidor FISAAPP (169.169.4.6)

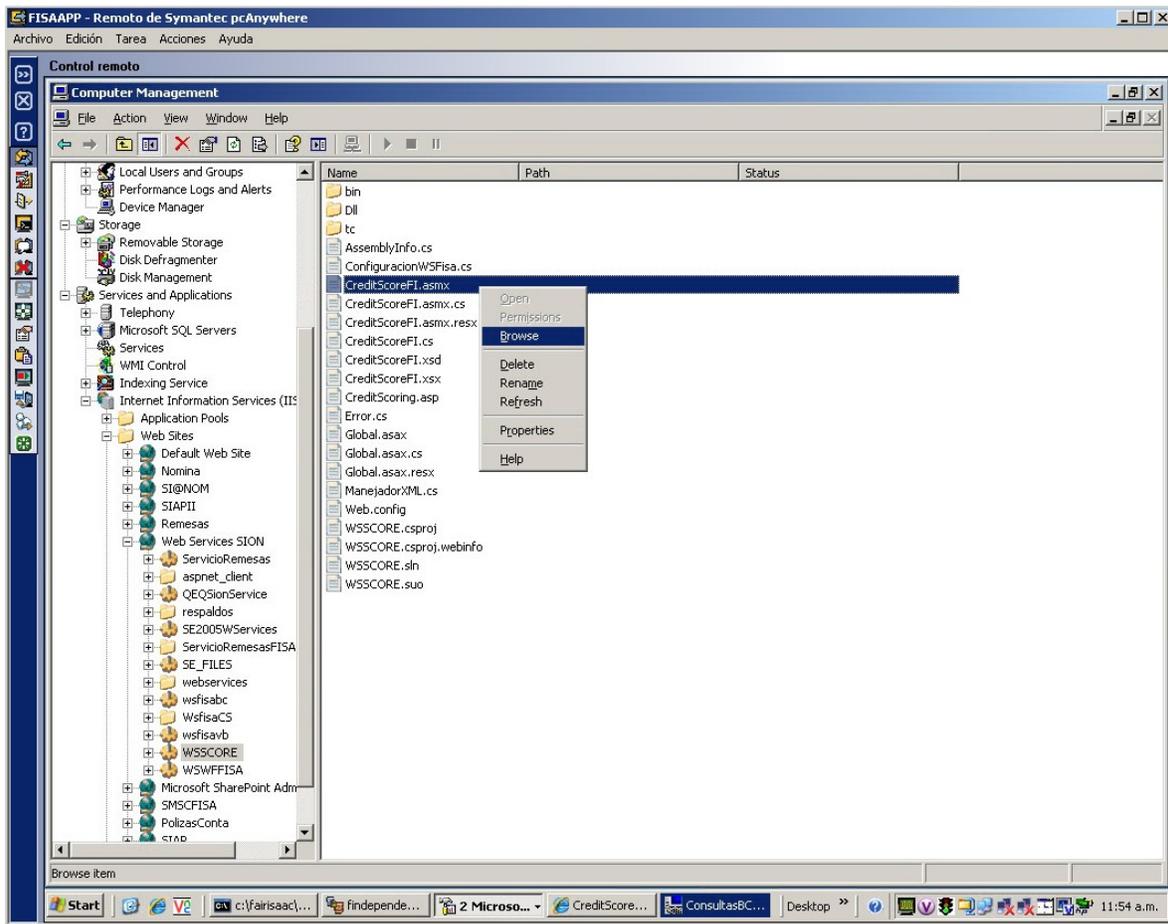


Figura 10

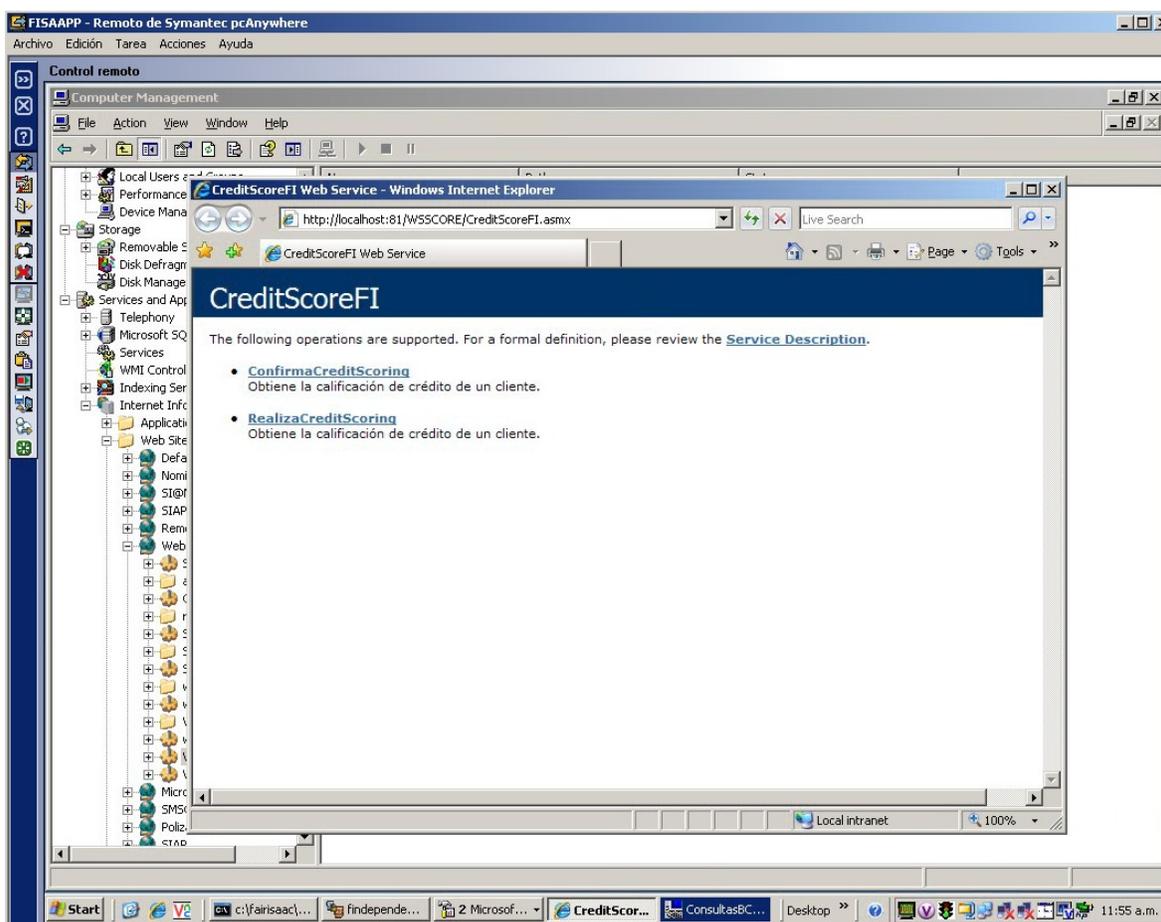


Figura 11. Verificación del servicio (El servicio está atendiendo peticiones)

• VERIFICACIONES

El átomo de verificación es una llamada asíncrona a los webservices de verificación telefónica y física.

Física: En este átomo el seo manda los datos con las direcciones del cliente (domicilio y laboral) para que un verificador compruebe que las direcciones son verídicas. Para asignar las verificaciones, el web service de este proceso genera registros en la base de datos local y mediante la sincronización de datos con las pockets pc's que tienen asignadas los verificadores obtienen todos los datos necesarios para poder acudir a los domicilios.

En la verificación el agente toma fotografías del domicilio o empleo, hace una entrevista a los vecinos y solicitante, llenando formularios en la aplicación de las pockets, la información obtenida se

puede enviar inalámbricamente utilizando la red gsm de nuestro proveedor o en la sucursal sincronizando datos nuevamente.

La aplicación genera un XML con la información necesaria para que después los dictaminadores determinen si se aprueba, se manda errónea o se rechaza la verificación física. Los dictaminadores evalúan las fotografías, así como los comentarios de los verificadores.

Telefónica: En este átomo el seo hace la llamada al web service de verificación telefónica pasándole los datos de los teléfonos del solicitante (directo, laboral y referencias) una vez que los datos están en la base de datos local, una aplicación de marcación va asignando a los operadores las llamadas, estos a su vez, entrevistan por teléfono al solicitante o a la persona que toma los recados para aceptar, enviar errónea o rechazar la solicitud.

Una vez que el seo recibe las respuestas de estos web services, evalúa las respuestas después de guardarlas en su base de datos local. Esto lo hace mediante un stored procedure y toma la decisión de aprobar, enviar errónea o rechazar la solicitud.

Problemas en el Proceso

Este proceso de verificación es en el que me encuentro con más problemas, ya que muchas de las solicitudes se detienen aquí. Esto se debe a los numerosos pasos que conlleva este proceso, sobre todo las verificaciones físicas. En estas últimas, el problema más frecuente, son los caracteres no soportados por la codificación de los xml's, ya que el verificador al ingresar los datos obtenidos escribe caracteres especiales, los cuales hacen que no se transfiera correctamente a través de la aplicación de la pocket a la base de datos el xml formado en la pocket. Para solucionar esto, lo que hago es encontrar en el xml el carácter especial y lo sustituyo o lo borro.

```
<Direccion1 num_persona="654602042" Tipo="DD" Direccion="12 297 X 21 Y 23 CASA CON
VIRGEN EN LA PTA." cp="97199" Zona="AZUL" Colonia="MARIA LUISA" Municipio=""
Ciudad="MERIDA" Estado="YUCATAN" Fecha="29/04/2009" CveCiudad="01" CveMunicipio=""
Entre_Calles="21 Y 23" Cita="" Nombre="ANA LUISA BRICENO PEREZ" Contrato="73953499"
Oid="3" tipo_objeto="D" Visitado="FALSE" Seleccionado="FALSE" Orden="4" ></Direccion1>
<Bitacora Tipo="VERIFICACION FISICA" Fch="06/04/2009" Hora_Gestion="13:35:05"
Contacto="EXITOSO" Resultado="PIDE MAS INFORMACION" Causa="INSOLVENCIA PREVIA"
Observaciones="DOM OK SE AUT PAGO DE 1 LETRA DD DA FPP A LA SIG QUINCENA / LUPITA"
Fch_Promesa=" " Fch_Prox_Accion="06/04/2009" ></Bitacora>
```

En este fragmento de XML podemos ver un ejemplo de un carácter especial, lo que se hace aquí es sustituir el carácter especial de `INFORMACIØN` por `INFORMACION` una vez hecho esto regresamos el XML a la pocket y el proceso continúa.

En el caso de las verificaciones telefónicas el problema más común con el que me encuentro es cuando el SEO llama al web service de Verificación telefónica y le contesta un status de éxito, sin embargo en la base de datos del call center, no se generó ningún registro y el seo si abrió el canal para escuchar por la respuesta de la verificación. Para dejar consistentes los datos en ambas bases, lo que hago es reiniciar el proceso de la solicitud en verificación telefónica, esto es, vuelvo a ejecutar el proceso correspondiente de esta solicitud para que se vuelva a hacer el llamado al web service de verificación telefónica, se cancelen los registros que se generaron en la base del seo e inserte unos nuevos registros en ambos lados del proceso (SEO y call center).

- **CONFIRMAR A CREDIT SCORE**

Este átomo hace básicamente lo mismo que la Consulta a score, pero su intención es “alimentar” la base de datos del score para hacer más precisa la evaluación del perfil, así que se puede decir que este átomo “aprende” los diferentes perfiles que se van calificando. Es importante mencionar que a este átomo van tanto las solicitudes aceptadas y rechazadas, de esta manera se tienen más datos sobre los perfiles.

Problemas en el Proceso

En este proceso ocurren exactamente los mismos problemas que en Consulta a Score, ya que es el mismo servicio.

- **GENERACIÓN DE CONTRATO**

En este átomo el SEO llama al Servicio de Generación de contrato el cual, llama un stored procedure en la sucursal de donde se haya ingresado la solicitud. El stored genera un registro con todas las variables del crédito calculadas en el proceso, para que puedan ser impresas en el

paquete de disposición que se le entregará al cliente. Una vez que está dado de alta en sucursal el contrato se notifica al SEO para que guarde en la base de datos local el número de contrato. Una vez que concluye el átomo síncrono, el SEO hace el llamado al servicio de Generación de Cheques en el SIC.

Problemas en el proceso

En este proceso es de los menos problemáticos. El único problema que se presenta el servicio de generación de contrato no puede llamar al stored procedure que genera el contrato en sucursal, debido a la comunicación o por causas externas (cierre emergente). Cuando esto sucede el SEO tiene mecanismos de reintentos y genera tareas, “congelando” el código que estaba ejecutando para que cuando se tengan las condiciones para continuar se vuelva a lanzar el proceso desde donde se quedó.

Lo que hago es que una vez que ya no existe el problema por el cual se detuvo el proceso, lanzo nuevamente la tarea del contrato para que continúe el proceso de la solicitud.

- **GENERACIÓN DE CHEQUES**

En este átomo es síncrono, genera en el SIC un registro con el cual se imprime un cheque con folios otorgados por una entidad bancaria, lleva los datos del crédito, así como los datos del cliente.

Problemas en el proceso

En este proceso el problema que se presenta es la comunicación con el servicio de cheques, ya que en ocasiones en enlace está abajo y no es posible llevar a cabo la llamada al servicio. Al igual que en átomos anteriores, el SEO hace reintentos de y genera tareas del código ejecutado para que posteriormente sean lanzadas. La solución que pongo en este proceso es volver a lanzar las tareas cuando las condiciones de comunicación así lo permiten.

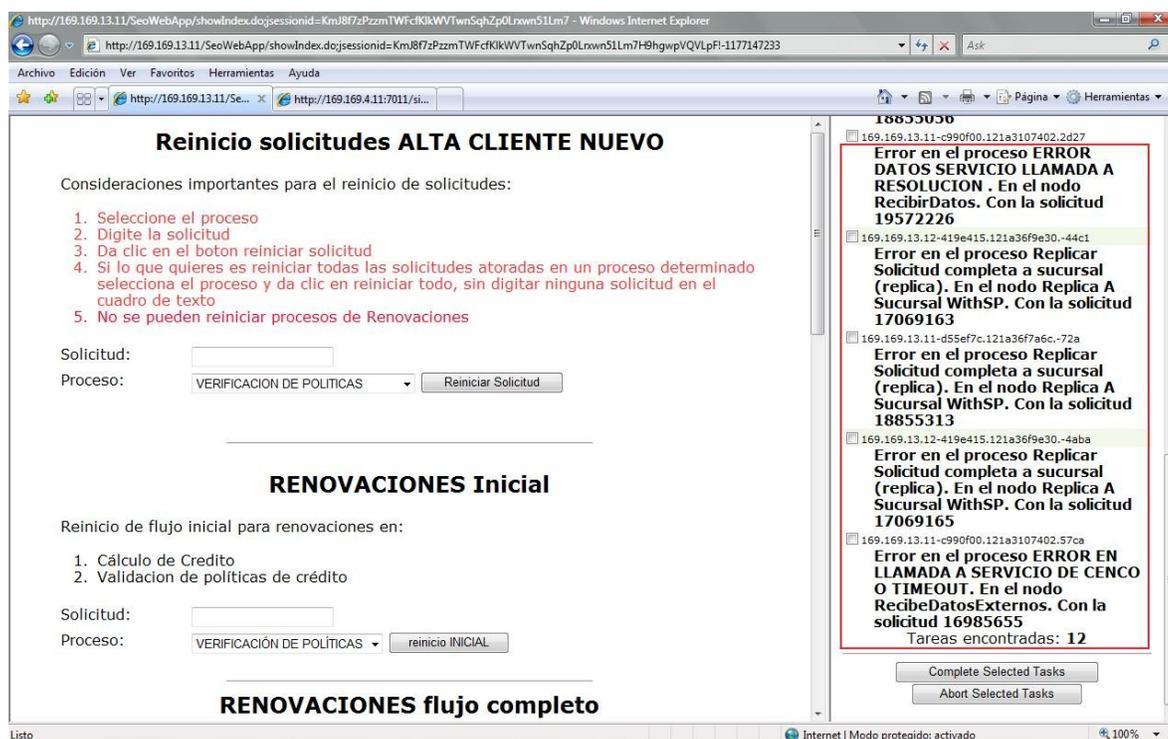


Figura 12. Página de reinicios de solicitudes y de Solución de tareas creadas en los diferentes átmos.

- **NUEVAMENTE EN SUCURSAL**

Una vez que ya se generó el contrato y el cheque, en sucursal, el área de administración se encarga de imprimir tanto el cheque como el contrato, esto para que el cliente firme de conformidad y se le entregue el cheque que puede ir a cambiar al banco al que pertenezca el cheque.

Con este proceso finaliza el POC y se inician nuevos procesos (caja y cobranza) en los cuales se tienen otras arquitecturas e implementaciones.

3.2 - El Ingeniero de Soporte a Producción en el POC

Como podemos ver, mi participación en el proceso es muy importante, pues ayudo a que las solicitudes que tienen algún problema y se detienen en algún proceso, continúen con su flujo. De no intervenir para corregir el problema que tengan, se produciría un efecto de “bola de nieve” ya que se van acumulando las solicitudes detenidas y puede llegar un momento en el que el sistema ocupe

más procesamiento para intentar avanzar las que tienen un problema, que las que no lo tienen, si esto ocurre el sistema puede empezar a generar “deadlocks” y o bloqueos en las diferentes bases de datos.

Anteriormente he comentado la solución en cada proceso en caso de que alguna solicitud tenga problema, sin embargo hacer lo que ya he comentado manualmente para cada proceso, requiere de mucho tiempo, es por eso que he desarrollado herramientas externas al SEO para corregir los problemas antes mencionados cuando estos ocurren.

Un ejemplo es el proceso de imagen, he programado una aplicación que hace una consulta a la base de datos del SEO, para conocer cuales solicitudes en este proceso son las más antiguas. La aplicación obtiene una lista, de esta última consulta en sucursal el nombre de la imagen para cada solicitud y crea un shellscript de Linux, su fin es mover las imágenes de la lista que se encuentren en alguna carpeta de respaldo dentro del servidor del seo, para ponerlas nuevamente en operación. Esta miniaplicación ha ayudado mucho en general a este proceso porque tenía tiempos muy largos de respuesta y siendo la primera parte del proceso de otorgamiento de crédito, la calidad y la satisfacción del cliente se estaba viendo afectada.

A continuación escribo la clase principal de la aplicación desarrollada que se menciona anteriormente.

```
package comunes.utilerias.seo;

import java.io.*;
import java.sql.*;
import java.util.HashMap;
import java.util.StringTokenizer;

/**
 *
 * @author mpalmaj
 */

public class MueveImagenes {
/**
 *
 * @param archivo Contiene las sentencias a subir
 * @param sucursal -1 Fisa
 *                 -2 Seo
 *                 -3 Conexia
 *                 -4 CENCA
 *                 -5 CENCACOA
 */
}
```

```

*/
private Connection conexion;
private Statement stmt;
private HashMap hMapaQuery;
private int sucursales=Integer.parseInt(LeeProperties.getPropiedad("NO_SUCURSALES"));

public MueveImagenes(){}
public MueveImagenes(String archivo,int sucursal){
    BufferedReader buffer=null;
    FileInputStream fis=null;
    try{
        iniciaConexion(sucursal);
        File f=new File(archivo);
        fis=new FileInputStream(f);
        buffer=new BufferedReader(new InputStreamReader(fis));
        String update="";
        conexion.setAutoCommit(false);
        Log.escribe("Leyendo archivo "+archivo);
        Log.escribe("Estado del autocommit= "+conexion.getAutoCommit());

        stmt=conexion.createStatement();
        ResultSet rs=stmt.executeQuery("");
        int i=1;
        int afectados=0;
        while((update=buffer.readLine())!=null){
            afectados=stmt.executeUpdate(update);

            Log.escribe(i+"-"+afectados+" registros actualizados");
            System.out.println(i+"-"+afectados);
            i++;
        }
        conexion.commit();
        Log.escribe("haciendo commit...");
        Log.escribe("~~~~~Fin de lectura y carga de "+archivo+" ~~~~~");

    }catch(FileNotFoundException fnf){
        Log.escribe(fnf);
        fnf.printStackTrace();
    }catch(IOException ioe ){
        Log.escribe(ioe);
        ioe.printStackTrace();
    }catch(SQLException sql ){
        Log.escribe(sql);
        sql.printStackTrace();
        try{
            conexion.rollback();
            Log.escribe("haciendo rollback...");
        }catch(Exception ex){
            Log.escribe(ex);
            ex.printStackTrace();
        }
    }

}finally{

    try{

```

```

        stmt.close();
        conexion.close();
        buffer.close();
        fis.close();
    }catch(Exception ex){
        Log.escribe(ex);
        ex.printStackTrace();
    }
}

}

}

/**
 *
 *
 * @param archivo String ya sea la lista de solicitudes pendientes o un query dentro
de un archivo.
 * @param tipo boolean false es lista, true es query
 * @return boolean verdadero si pudo crear archivo conteniendo solicitudes y
sucursales
 *         falso si ocurrió algún error.
 *
 */
public boolean getPendientes(String archivo,boolean tipo){
    BufferedReader buffer=null;
    FileInputStream fis=null;
    ResultSet rs=null;
    PrintStream imprime=null;
    StringBuffer query=new StringBuffer();
    boolean respuesta=false;
    FileOutputStream fos=null;

    try{
        File f=new File(archivo);
        fis=new FileInputStream(f);
        buffer=new BufferedReader(new InputStreamReader(fis));
        iniciaConexion(-2);
        Log.escribe("Leyendo archivo "+archivo);
        String temp="";
        //Log.escribe("Estado del autocommit= "+conexion.getAutoCommit());
        if(tipo){
            do{
                query.append(temp+"\n");
            }while((temp=buffer.readLine())!=null);
        }else{
            query.append(LeeProperties.getPropiedad("QUERY_PART_1"));
            do{
                if(!temp.equals(""))
                    query.append("'" +temp+"',");
            }while((temp=buffer.readLine())!=null);
            query.deleteCharAt(query.lastIndexOf(","));
            query.append(LeeProperties.getPropiedad("QUERY_PART_2"));
        }
    }
}

```

```

    }
    Log.escribe(query.toString());
    stmt=conexion.createStatement();
    rs=stmt.executeQuery(query.toString());

    File resultado=new File("resultado.txt");
    fos=new FileOutputStream(resultado);
    imprime=new PrintStream(fos);
    temp="";
    int rows=1;
    while(rs.next()){
        rows++;
        if(!temp.equals(rs.getString(1))){
            temp=rs.getString(1);
            imprime.println("---");
        }
        imprime.println("<" +rs.getString(1)+"|" +rs.getString(2)+"',>");
    }
    Log.escribe("Archivo Creado \"resultado.txt\" con "+rows+" solicitudes para
    buscar en sucursal");

    respuesta=true;
}catch(FileNotFoundException fnf){
    Log.escribe(fnf);
    fnf.printStackTrace();
}catch(IOException ioe ){
    Log.escribe(ioe);
    ioe.printStackTrace();
}
catch(SQLException sql ){
    Log.escribe(sql);
    sql.printStackTrace();
}
finally{
    try{

        if(conexion!=null)
            conexion.close();
        if(rs!=null)
            rs.close();
        if(stmt!=null)
            stmt.close();
        imprime.close();

        buffer.close();
        fis.close();

        Log.escribe("Conexiones y flujos cerradas para crear archivo resultado.txt");
    }catch(Exception ex){
        Log.escribe(ex);
        ex.printStackTrace();
    }
}

return respuesta;

```

```

}
/**
 *
 * @description Este método busca en sucursal el nombre completo de la imagen para
 generar un Shell que busque en el servidor dicha imagen y la mueva a las carpetas de
 operación.
 *
 *
 **/
public boolean getQuerySucursal(){
    boolean respuesta=false;
    FileInputStream fis=null;
    BufferedReader buffer=null;
    HashMap hMapa=new HashMap();
    StringBuffer resultados=new StringBuffer();
    //int sucursales=Integer.parseInt(LeePropiedades.getPropiedad("NO_SUCURSALES"));
    String inicioQuery,finQuery;
    //StringBuffer solicitudes=new StringBuffer();
    try{
        File f=new File("resultado.txt");
        fis=new FileInputStream(f);
        buffer=new BufferedReader(new InputStreamReader(fis));
        String linea="";
        do{
            resultados.append(linea);
        }while((linea=buffer.readLine())!=null);
        linea="";
        StringTokenizer token= new StringTokenizer(resultados.toString(),"<|>\n\t");
        String sucursalActual="";
        resultados.delete(0, resultados.length());
        int counter=0;
        while(token.hasMoreTokens()){
            linea=token.nextToken();
            if(linea.equals("---") && !sucursalActual.equals("")){
                resultados.deleteCharAt(resultados.lastIndexOf(","));
                hMapa.put(sucursalActual,resultados.toString());
                resultados.delete(0, resultados.length());
            }else{
                if(counter%2==0 && !sucursalActual.equals("")){
                    resultados.append(linea);
                }else
                    sucursalActual=linea;
                counter++;
            }
        }
        resultados.deleteCharAt(resultados.lastIndexOf(","));
        hMapa.put(sucursalActual,resultados.toString());
        resultados.delete(0,resultados.length());
        inicioQuery=LeePropiedades.getPropiedad("QUERY_PART_SUC_1");
        finQuery=LeePropiedades.getPropiedad("QUERY_PART_SUC_2");
        hMapaQuery =new HashMap();
        for(int i=1;i<=sucursales;i++){
            if(hMapa.containsKey(""+i)){
                resultados.append(inicioQuery+hMapa.get(""+i)+finQuery);
                hMapaQuery.put(""+i,resultados.toString());
                Log.escribe("Sucursal: "+i+" Query: "+resultados.toString());
                resultados.delete(0,resultados.length());
            }
        }
    }
}

```

```

    }
    respuesta=true;

}catch(Exception ex){
    Log.escribe(ex);
}finally{
    try{
        fis.close();
        buffer.close();
        Log.escribe("Flujos cerradas para crear archivo resultado.txt");
    }catch(Exception ex){
        Log.escribe(ex);
    }
}

}
return respuesta;
}

/**
 *
 * @param String nombreShell el nombre del Shell que se va a generar que contendrá
los comandos en Linux para buscar y mover las imágenes en sucursal.
 *
 */
public boolean getShellFile(String nombreShell){
    ResultSet rs=null;
    StringBuffer commands=new StringBuffer();
    int lugarDisposicion;
    String preliberacion=LeeProperties.getPropiedad("FOLDER_SEARCH_1");
    String trash=LeeProperties.getPropiedad("FOLDER_SEARCH_2");
    String destUno=LeeProperties.getPropiedad("FOLDER_DEST_1");
    String destDos=LeeProperties.getPropiedad("FOLDER_DEST_2");
    String pathImagen="";
    File shell=new File(nombreShell);
    PrintStream escribe=null;
    FileOutputStream fos=null;
    try{
        fos=new FileOutputStream(shell);
        escribe=new PrintStream(fos);
        escribe.println("#!/bin/sh");
        for(int i=1;i<=sucursales;i++){
            lugarDisposicion=0;

            if(hMapaQuery.containsKey(""+i)){

                iniciaConexion(i); //Para producción
                stmt=conexion.createStatement();
                rs=stmt.executeQuery(hMapaQuery.get(""+i).toString());

                while(rs.next()){
                    Log.escribe("Sucursal:"+i+" Documento: "+rs.getString(2));
                    lugarDisposicion=rs.getInt(4);
                    commands.delete(0,commands.length());
                    if(lugarDisposicion==1)
                        commands.append(LeeProperties.getPropiedad("SHELL_COMMAND_1")
+i+" ");
                    else
                        commands.append(LeeProperties.getPropiedad("SHELL_COMMAND_1")
+i+"."+(lugarDisposicion-1)+" ");
                }
            }
        }
    }
}

```

```

        pathImagen=rs.getString(3);
        commands.append("'find "+preliberacion+" -name \""+pathImagen+"\"
");
        commands.append("-exec cp {} "+destUno+" \\;; ");
        commands.append("find "+preliberacion+" -name \""+pathImagen+"\"
");
        commands.append("-exec cp {} "+destDos+" \\;; ");
        commands.append("find "+trash+" -name \""+pathImagen+"\" ");
        commands.append("-exec cp {} "+destDos+" \\;; ");
        commands.append("find "+trash+" -name \""+pathImagen+"\" ");
        commands.append("-exec cp {} "+destUno+" \\;');");
        commands.append("\necho 'VALOR DE LA EJECUCION<
"+rs.getString(2)+" >: '$?");
        escribe.println(commands.toString());
    }

    }

    if(conexion!=null)
        conexion.close();
    if(rs!=null)
        rs.close();
    if(stmt!=null)
        stmt.close();

}
escribe.close();
}catch(Exception ex){
    Log.escribe(ex);
}finally{
    try{
        if(conexion!=null)
            conexion.close();
        if(rs!=null)
            rs.close();
        if(stmt!=null)
            stmt.close();
        escribe.close();
        Log.escribe("Flujos Cerrados para conexion a sucursal");
    }catch(Exception ex){
        Log.escribe(ex);
    }
}
return false;
}
/**
 *
 * @param String shellFile nombre del Shell que se va a ejecutar después de haber
generado los comandos
 *
 */
public boolean ejecutaShell(String shellFile){
    boolean termino=false;
    Process p=null;
    BufferedReader input=null;
    BufferedReader inputError=null;

```

```

String linea="";
try {
    System.out.println("Ejecutando proceso");
    p = Runtime.getRuntime().exec("/bin/bash "+shellFile);
    p.waitFor();

    input =new BufferedReader(new InputStreamReader(p.getInputStream()));
    inputError =new BufferedReader(new InputStreamReader(p.getErrorStream()));
    while ((linea = input.readLine()) != null) {
        System.out.println(linea);
        Log.escribe(linea);
    }
    while ((linea = inputError.readLine()) != null) {
        System.out.println(linea);
        Log.escribeError(linea);
    }
    input.close();
    inputError.close();
    termino=true;
}catch (Exception err) {
    System.out.println(err.getMessage());
    Log.escribe(err);
}finally{
    try{
        input.close();
        inputError.close();
        Log.escribe("Valor de finalizacion para el proceso: "+p.exitValue());
    }catch(Exception ex){
        Log.escribe(ex);
        ex.printStackTrace();
    }
}
return termino;
}

public void iniciaConexion(int cual) throws SQLException {
    switch (cual){
        case -1:
            conexion=ConnectionFactory.getConnection("driverCentral",
"servidorBDFISA3", "usuarioFISA3", "passwordFISA3");
            break;
        case -2:
            conexion=ConnectionFactory.getConnection("driverSybase", "servidorBDSE0",
"usuarioBDSE0", "passwordBDSE0");
            break;
        case -3:
            conexion=ConnectionFactory.getConnection("driverSybase","servidorBDFISA2",
"usuarioFISA2","passwordFISA2");
            break;
        case -4:
            conexion=ConnectionFactory.getConnection("driverSucursal","servidorCENCA",
"usuarioCENCA","passwordCENCA");
            break;
        case -5:
            conexion=ConnectionFactory.getConnection("driverSucursal","servidorCENCACOA",
"usuarioCENCACOA","passwordCENCACOA");
            break;
        default:
            conexion=ConnectionFactory.getConnectionSucursal(cual);
    }
}

```

```

        break;
    }
}

public static void main(String args[]){
    try{
        MueveImagenes mueve=new MueveImagenes();

        if(mueve.getPendientes(args[0],args[1].toUpperCase().equals("QUERY"))
            mueve.getQuerySucursal();
        if(LeeProperties.getPropiedad("ATOMO_PENDIENTE").equals("ATOI")){
            Log.escribe("ATOMO PENDIENTE: ATOI");
            mueve.getShellFile(args[2]);
        }
        if(LeeProperties.getPropiedad("RUN_PROCESS").equals("S")){
            Log.escribe("SI EJECUTA EL SHELL!!!");
            if(mueve.ejecutaShell(args[2]))
                Log.escribe("SE EJECUTO CON EXITO EL SHELL");
        }
    }catch(ArrayIndexOutOfBoundsException aiobe){
        Log.escribe(aiobe);
        System.out.println("Los parametros que se deben pasar a la clase son:");
        System.out.println("args[0]: Nombre de archivo donde se va a leer");
        System.out.println("args[1]: Tipo de archivo, QUERY o LISTA");
        System.out.println("args[2]: Nombre del Shell que va a generar");
    }
}
}
}

```

Como este caso, he desarrollado soluciones para cada proceso del SEO, sin embargo, éstas son temporales, porque otro de mis deberes como ingeniero de soporte a producción es corregir el código que no esté funcionando debidamente dentro del SEO o en alguna de sus interfaces, por eso es que se toman de manera temporal dichas soluciones.

Un ejemplo de la corrección del código que no estaba funcionando debidamente, era en la parte de replicar la solicitud completa a las sucursales, en este caso, los teléfonos de los clientes se estaban replicando de las tablas del SEO a las de sucursal con status erróneos (Inactivos en lugar de activos) y la fecha en la que se daban de alta esos teléfonos en financiera independencia se replicaba en blanco. Por lo tanto las estrategias del área comercial para contactar a los clientes por teléfono no podrían funcionar y no podríamos ofrecerles una mayor gama de servicios a nuestros clientes. Por mi parte yo corregí esta parte del código para que los teléfonos de los clientes se repliquen de forma correcta a las sucursales, ya que es en el SEO, en el átomo de captura, en

donde se ingresan los teléfonos de los clientes y estos deben estar correctos para que las sucursales los puedan utilizar correctamente.

A continuación escribo el método que generé para replicar correctamente los teléfonos a sucursal dentro del JPD “ReplicaSucursal”.

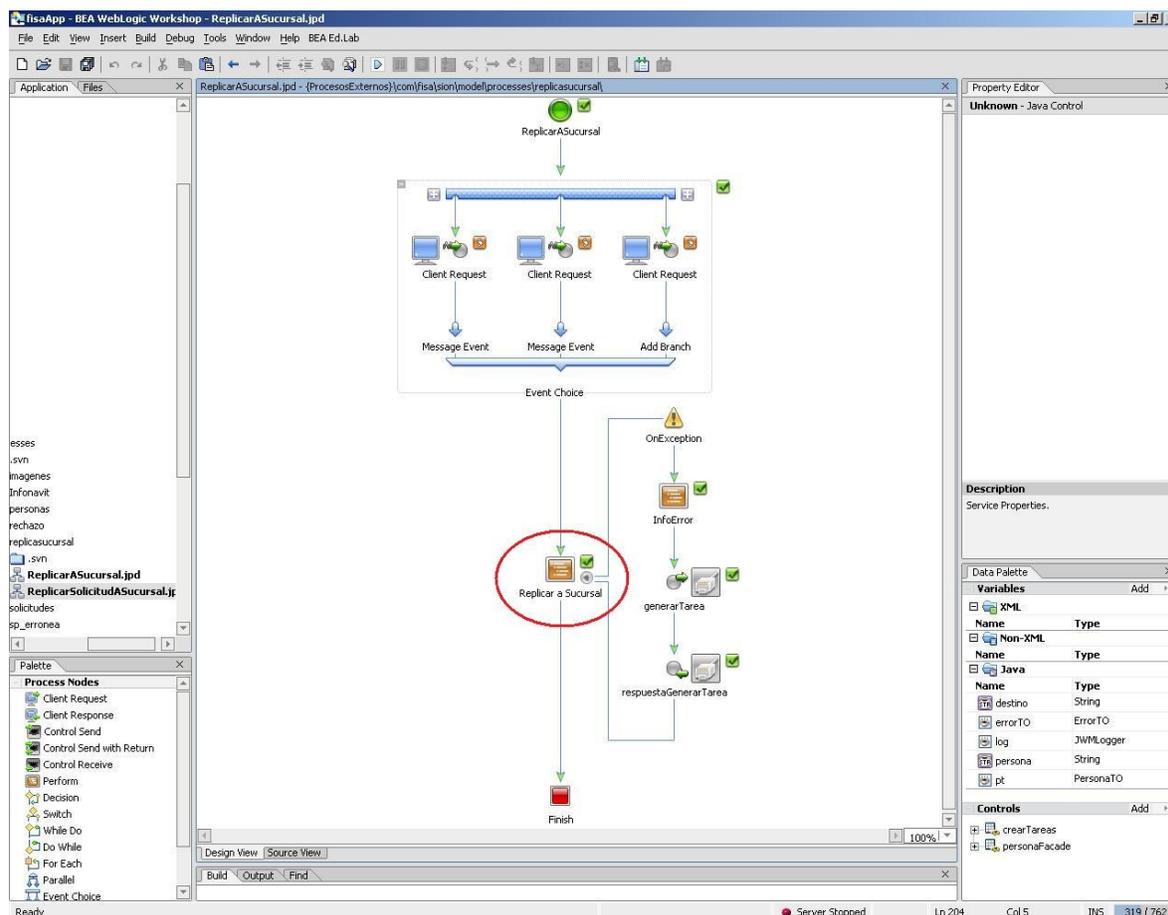


Figura 13. JPD ReplicaASucursal – Se indica el punto en donde se implementó el método dentro del flujo para replicar correctamente los teléfonos.

```

/**
 * Método que sustituye a los métodos para insertarTelefono y actualizar telefono
 * CUIDAR EL NÚMERO DE PARAMETROS EN CICLO, ENTRADA , SALIDA
 */
public ReplicaTelefonoResponse replicaTelefono(TelefonoTO to) throws
SionSystemException
{
    CallableStatement pstmt = null;
    ReplicaTelefonoResponse response = new ReplicaTelefonoResponse();

    if(con == null)
        getConnection();

    if(con != null)
    {

```

```

try {

    log.debug("Datos del telefono [{"+ to.toString() +"}]");

    //Se construye la lista de parametros
    //CUIDAR EL NÚMERO TOTAL DE PARAMETROS 23 DE ENTRADA + 2 DE SALIDA
    String parametrosSP = "";
    for(int i = 0 ; i < 22 + 2 ; i++)
        parametrosSP += "?,";

    String llamadaSP = "{call sp_registra_telefono("+
parametrosSP.substring(0,parametrosSP.length()-1) +")}";

    pstmt = con.prepareCall(llamadaSP);
    log.debug("Se prepara llamada de sp_registra_telefono con timeOut
["+Globals.STATEMENT_TIMEOUT+" ]");
    pstmt.setQueryTimeout(Globals.STATEMENT_TIMEOUT);

    pstmt.setString(1, to.getPersona() );
    pstmt.setString(2, to.getTipoTelefono() );
    pstmt.setString(3, to.getLada() );
    pstmt.setString(4, to.getTelefono() );
    pstmt.setString(5, to.getTipoRed() );
    pstmt.setString(6, to.getExtension() );
    pstmt.setString(7, to.getNombre() );
    pstmt.setString(8, to.getApellidoPat() );
    pstmt.setString(9, to.getApellidoMat() );
    pstmt.setString(10, to.getRelacionCliente() );
    pstmt.setString(11, to.getCodigoResolucion() );
    pstmt.setString(12, to.getStatus() );
    if(to.getFechaAlta() != null)
        pstmt.setDate(13, new java.sql.Date(to.getFechaAlta().getTime()));
    else
        pstmt.setDate(13, new java.sql.Date(new Date().getTime()));
    pstmt.setString(14, to.getUsuarioAlta() );
    pstmt.setDate(15, new java.sql.Date(to.getFechaUltimaMod().getTime() ) );
    pstmt.setString(16, to.getUsuarioMod() );
    pstmt.setDate(17, to.getFechaBaja() != null ? new
java.sql.Date(to.getFechaBaja().getTime()) : null );
    pstmt.setString(18, to.getUsuarioBaja() );

    if (to.getFechaUltimaVerif() != null)
        pstmt.setDate(19, new
java.sql.Date(to.getFechaUltimaVerif().getTime() ) );
    else
        pstmt.setDate(19, null );

    pstmt.setString(20, to.getLugar() );
    pstmt.setString(21, to.getOrigen() );
    pstmt.setString(22, to.getCambioTelefono() );

    //Parametros de salida
    pstmt.registerOutParameter(23, java.sql.Types.INTEGER);
    pstmt.registerOutParameter(24, java.sql.Types.VARCHAR);

    pstmt.execute();

    response = new ReplicaTelefonoResponse();

```

```
        response.codigo = pstmt.getInt(23);
        response.comentario = pstmt.getString(24);

        pstmt.close();

        log.debug("Se termina la llamada sp_registra_telefono con timeOut
["+Globals.STATEMENT_TIMEOUT+"] ");
    }
    catch(Exception ex)
    {
        response.codigo = -1;
        response.comentario = "Error en el SEO al ejecutar
{sp_registra_telefono}";
        log.error("Error llamando SP sp_registra_telefono", ex);
        ex.printStackTrace();
        throw new SionSystemException(ex);
    }
    finally
    {
        this.cleanCallableUp(pstmt, null);
    }
}

return response;
}
```

CONCLUSIÓN

A lo largo del trabajo que realicé puedo notar que se cumple el objetivo de explicar el flujo de una solicitud a través de todos los procesos que componen el POC. Así como la migración de los sistemas aislados a una integración de ellos, aplicando tecnologías que sin la formación como ingeniero en computación, no sería fácil implementar para un funcionamiento óptimo.

Resaltaré los siguientes puntos.

Ingeniería en Computación: La carrera de ingeniería en computación brinda bases sólidas e integrales sobre TI, suficientes para que los egresados nos sumemos al entorno laboral, sin embargo, es nuestro deber desarrollar dichas bases para complementar los conocimientos recibidos durante la carrera y de esta forma seremos globalmente competitivos.

Trabajo Profesional: En el día a día de mi desempeño profesional en el puesto de ingeniero de soporte a producción, he aplicado muchos de los conocimientos adquiridos en la carrera de ingeniería en computación. Las materias que han sido de gran ayuda por mencionar algunas son, Bases de Datos, Lenguajes de Programación, Ingeniería de Software, Redes de Datos, Sistemas Operativos, entre otras. Todas ellas me han ayudado a cumplir con mi trabajo diario, dejando en el mismo una gran calidad en el aspecto técnico y humano.

La Integración de Sistemas: Las exigencias de hoy en día requieren en pocas palabras hacer más con menos y aprovechar los recursos existentes. Lo anterior es precisamente la filosofía de la integración de sistemas, ya que involucra todos los niveles de los sistemas de una empresa, obteniendo como consecuencia un aumento en la productividad y disminuye el gasto relacionado con TI.

Financiera Independencia: La arquitectura actual, basada en la arquitectura SOA y en el modelo del ESB, permite que Financiera Independencia cuente con un canal de interconexión común que cubre las necesidades de comunicación, fiabilidad, independencia y flexibilidad que requiere. El ESB permite también a Financiera Independencia la integración de nuevos sistemas a su arquitectura de manera rápida.

La arquitectura así mismo permite a Financiera Independencia llevar a cabo sus procesos de negocio de una manera más ágil, rápida y contribuye a la disminución de los costos relacionados a su operación. Permite también contar con una infraestructura tecnológica que les permite continuar con su proceso de expansión de forma exitosa y adaptarse a los continuos cambios de negocio que el entorno demanda.

Aportaciones a Financiera Independencia: Como ingeniero de soporte a producción le doy mantenimiento y monitoreo el desempeño al SEO, que es el sistema encargado de generar los créditos para cada solicitud, además corrijo los errores que surgen en el mismo. Con esto prevengo fallas que puedan impactar a nivel nacional en la operación diaria de la empresa.

Finalmente quiero resaltar que es para mí, un gran orgullo haber egresado de la Universidad Nacional Autónoma de México, por ser una gran casa de conocimientos, formadora de excelentes profesionales, que están comprometidos a mejorar diariamente México.

Bibliografía

Capítulo 1

1. **ARTIGES**, Mark “BEA Weblogic Server 8.1 Unleashed”. Indianapolis. SAMS. 2004
2. **O. GRADY**, Jeffrey “System Integration”. CRC Press, 1994
3. **MYERSON**, Judith M. “Enterprise Systems Integration”. CRC Press, 2002
4. **CERAMI**, Ethan “Web Services Essentials” O'Reilly,2002
5. **NYBERG**, Greg “Mastering BEA WebLogic Server: best practices for building and deploying J2EE applications” John Wiley and Sons, 2003
6. **HOHPE**, George “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions” Addison-Wesley, 2005
7. **ROSEN**, Michael “Applied SOA: Service Oriented Architecture and design strategies” Wiley-India, 2008

Mesografía

Capítulo 1

1. **ENTERPRISE APPLICATION INTEGRATION**, Overview. Fecha de creación 2005, disponible en http://eai.ittoolbox.com/pub/eai_overview.htm Consulta Marzo 2009.
2. **ORACLE AND BEA SYSTEMS**, <http://www.oracle.com/bea/index.html> Consulta Marzo 2009.

Capítulo 2

3. **SUN DEVELOPER NETWORK**, Developer Resources for Java Technology, Fecha de Creación 1994, disponible en <http://java.sun.com> Consulta Marzo 2009
4. **BEA SYSTEMS**, Product Documentation BEA WebLogic Integration 8.1. Fecha de creación 2006, Disponible en <http://e-docs.bea.com/wli/docs81/index.html> Consulta Marzo 2009.
5. **WIKIPEDIA**, La encyclopedia libre, <http://es.wikipedia.org/> Consulta Marzo 2009

GLOSARIO

.NET: Proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

B2B: Es la abreviatura comercial de la expresión “business to business”: (comunicaciones de comercio electrónico) de empresa a empresa, por oposición a las relaciones de comercio entre empresas y consumidores (B2C), o las expresiones menos usadas empresas y gobierno (B2G) o empresas y empleados (B2E).

BPEL: es un lenguaje estandarizado por OASIS (Organization for the Advancement of Structured Information Standards) para la composición de servicios web. Está desarrollo a partir de WSDL y XLANG, ambos lenguajes orientados a la descripción de servicios Web.

BPM: metodología empresarial cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorear y optimizar de forma continua.

CICS: Customer Information Control System (en español, Sistema de control de información de clientes), es un gestor transaccional, o monitor de teleproceso, que se ejecuta principalmente en mainframes IBM con los sistemas operativos OS/390, z/OS o VSE. También existen versiones de CICS para otros entornos, como OS/400, OS/2, etc. La versión para entornos Unix recibe el nombre de TX Series.

COM+: *Component Object Model*, es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología. El término COM es a menudo usado en el mundo del desarrollo de software como un término que abarca las tecnologías OLE, OLE Automation, ActiveX, COM+ y DCOM.

CORBA: *Common Object Request Broker Architecture*, Es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

CRM: *Customer Relationship Management*, es parte de una estrategia de negocio centrada en el cliente. Una parte fundamental de su idea es, precisamente, la de recopilar la mayor cantidad de información posible sobre los clientes, para poder dar valor a la oferta.

CVS: Concurrent Version System, producto de control de versiones que permite proteger los cambios hechos al código.

DLL: *Dynamic Linking Library*, término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.

EDI: *Electronic Data Interchange*, es un software Middleware que permite la conexión a distintos sistemas empresariales como ERP o CRM.

EJB: *Enterprise Java Beans*, son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Oracle - Sun Microsystems (ahora JEE 5.0).

ERP: *Enterprise Resource Planning*, son sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios.

ESB: *Enterprise Service Bus*, Se trata de un Servicio Orientado a Infraestructuras (SOI), formado por una colección de patrones de arquitectura basados en la Enterprise Application Integration (EAI), Web services, .NET y la interoperabilidad con JAVA, integración de sistemas host e interoperabilidad con servicios de registro y repositorio de activos.

GSM: *Sistema Global para las Comunicaciones Móviles*, es un sistema estándar, completamente definido, para la comunicación mediante teléfonos móviles que incorporan tecnología digital. Por ser digital cualquier cliente de GSM puede conectarse a través de su teléfono con su ordenador y puede hacer, enviar y recibir mensajes por e-mail, faxes, navegar por Internet, acceso seguro a la red informática de una compañía (LAN/Intranet), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el Servicio de Mensajes Cortos (SMS) o mensajes de texto.

HTTP: *HyperText Transfer Protocol*, es el protocolo usado en cada transacción de la Web. HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

J2EE: *Java Enterprise Edition*, es una plataforma de programación (parte de la Plataforma Java) para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JPD: Java Process Definition, Componente de WLI que contiene la lógica en java para la orquestación de servicios, así como anotaciones que controlan el flujo del proceso de negocio.

MQSeries: Es un middleware de Mensajes de IBM, que permite a los programas comunicarse entre todas las plataformas de IBM, Windows, VMS y varias plataformas UNIX.

RPC: *Remote Procedure Calling*, Llamado remoto de procedimientos, estilo de servicio web síncrono, simula un método en un componente remoto.

SOA: Arquitectura Orientada a Servicios (*Service-Oriented Architecture*) definición basada en integrar los componentes heterogéneos mediante servicios web síncronos o asíncronos

SOAP: Simple Object Access Protocol, estándar basado en XML para la comunicación con servicios web.

Struts: es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

TCP/IP: Transport Control Protocol / Internet Protocol, Protocolo de comunicación de Internet basado en sockets.

UDDI: *Universal Description, Discovery and Integration*, es un catálogo de negocios de Internet. El registro en el catálogo se hace en XML. UDDI es una iniciativa industrial abierta entroncada en el contexto de los servicios Web.

Web service, WS, Servicio Web: Aplicación web que se comunica con clientes a través de XML (SOAP), implementa un servicio de negocio.

WLI: *WebLogic Integration*, plataforma de BEA que provee infraestructura para la implementación de procesos de negocio y Canales de comunicación con otros sistemas.

WSDL: *Web Services Definition Language*, Estándar basado en XML que sirve para especificar los parámetros y respuestas que implementará un servicio web.

XML: *Extensible Markup Language*, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML (Standard Generalized Markup Language) y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

XQuery: Es un lenguaje de consulta diseñado para consultar colecciones de datos XML. Es semánticamente similar a SQL, pero incluye algunas capacidades de programación.

XSLT: *Transformaciones XSL*, es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.