



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**

**SISTEMA PARA ENROLAMIENTO  
DE PROVEEDORES MÉDICOS**

**DISEÑO DE UN SISTEMA PARA UNA  
ORGANIZACIÓN**

**EDWIN ERIK RODRÍGUEZ RUIZ**



**MÉXICO, D.F.**

**2007**



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**

**SISTEMA PARA ENROLAMIENTO  
DE PROVEEDORES MÉDICOS**

**DISEÑO DE UN SISTEMA PARA UNA ORGANIZACIÓN  
QUE PARA OBTENER EL TÍTULO DE:**

**LICENCIADO EN INFORMÁTICA**

**PRESENTA:**

**EDWIN ERIK RODRÍGUEZ RUIZ**

**ASESOR:**

**M. en A. LUIS EDUARDO LÓPEZ CASTRO**



**MÉXICO, D.F.**

**2007**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A Marina, por su apoyo incondicional en todo momento.

A David, por haberme dado una de las mayores alegrías en mi vida.

A mis padres Oscar y Gaby, por haberme dado el regalo de la vida y por darme una educación

Al Profesor Luis Eduardo, por que sin su ayuda esto no habría sido posible

# Índice

---

Introducción	1
Descripción del Entorno	2
Definición del Problema	5
Análisis, Enrolamiento de Proveedores	11
Diseño ProviderEnrollmentMain	30
Diseño ProviderEnrollmentCommon	41
Diseño Web Service Verificar Estatus	48
Diseño Agregar Localidad de Servicio	52
Diseño de XSD para Enrolamiento de Proveedores	68
Diseño de Web Service de Notificación	100
Diseño para desarrollo de Web Services de Lista de Trabajo	104
Diseño Interfaces NTier	114
Anexo A. Descripción de las Tablas	119
Anexo B. Glosario de Términos	122
Anexo C. Prototipos de las Interfaces Gráficas	125
Anexo D. Referencias Web	132

# 1. Introducción

---

En el presente trabajo se realiza el Diseño de un Sistema para el enrolamiento de nuevos Proveedores para el programa Medicaid del gobierno de los Estados Unidos de América.

Este Sistema basado en SOA (*Service Oriented Architecture*, Arquitectura Orientada a Servicios) se construye sobre interChange, un MMIS (*Medicaid Management Information System*, Sistema de Información para la Administración de Medicaid) desarrollado por la compañía Electronic Data Systems (EDS).

interChange es un sistema COTS (*Commercial off-the-shelf*, Producto Comercial Producido en Masa), el cual ha sido modificado para cumplir con los requerimientos específicos de cada uno de los Estados de los Estados Unidos de América donde EDS ha obtenido la concesión a través de una licitación.

Para la implementación del Sistema se utilizó como base el orquestador BPEL (*Business Process Execution Language*, Lenguaje de Ejecución de Procesos de Negocio). El Sistema que en este documento se presenta es parte de un proyecto piloto el cual permitió conocer los alcances y limitaciones de las herramientas que se utilizaron durante el ciclo de vida de este desarrollo.

El objetivo de este trabajo de Diseño es implementar BPEL y SOA en el subsistema de Proveedor (concretamente Enrolamiento de Proveedores) del programa Medicaid del gobierno de los Estados Unidos de América.

## 2. DESCRIPCIÓN DEL ENTORNO

---

### 2.1 Perfil de la Compañía

EDS es una compañía líder de servicios de tecnología global que proporciona soluciones de negocio a sus clientes. EDS fundó la industria del *outsourcing* en tecnología de información hace más de 40 años. El día de hoy, EDS ofrece un amplio portafolio de servicios de *outsourcing* en tecnología de información y procesos de negocio a sus clientes en las industrias de manufactura, servicios financieros, servicios médicos, telecomunicaciones, energía, transportación y ventas al detalle así como gobiernos al derredor del mundo.

### 2.2 ¿Qué hacemos?

Como los consejeros confiable de los CIO, EDS proporciona las mejores soluciones para que los ejecutivos puedan maximizar el retorno en sus inversiones de IT. Nuestro profundo conocimiento de la industria permite a los clientes concentrarse en los problemas específicos a su negocio, y nuestra infraestructura global sin par proporciona la capacidad y potencial para asegurarnos de atender a nuestros clientes extraordinariamente bien.

### 2.3 ¿Cómo lo hacemos?

Nuestro portafolio se construye al derredor de ofertas innovadoras en mantenimiento y desarrollo de aplicaciones, *outsourcing* de procesos

de negocio e infraestructura, incluyendo servicios de escritorio, hospedaje, almacenamiento y redes. Estos servicios se proporcionan a través de la Plataforma Corporativa Ágil de EDS, nuestro sistema global de siguiente generación que permitirá a nuestros clientes responder rápidamente a las dinámicas de mercado cambiantes y aumentar su competitividad.

### **¿Sabía qué? EDS...**

- Administra más de 65,000 servidores y da soporte a más de 3 millones de computadoras de escritorio al derredor del mundo
- Administra más de 350 millones de relaciones y más de 2,400 millones de interacciones con clientes anualmente en 43 idiomas
- Proporciona servicios de Recursos Humanos a más de 400 organizaciones en los Estados Unidos, Canadá, Europa, así como Centro y Sudamérica, y 33 millones de empleados y participantes en todo el mundo
- Tiene agentes de *help-desk* que contestan 20 millones de llamadas de soporte anualmente y dan servicio a cliente en 24 idiomas
- Procesa más de 1,000 millones de reembolsos médicos anualmente
- Proporcionó más de 78,000 horas/voluntario en 2005 para dar soporte a una variedad de proyectos comunitarios globales

- Es el patrocinador oficial del campeonato EDS Byron Nelson, que atrae anualmente a las más grandes y brillantes estrellas del golf y proporciona más dólares para la caridad que cualquier otro evento de la PGA
- Proporciona los servicios colectivos de las compañías de la Alianza Ágil de EDS (más de 200 mil millones en ingresos totales anuales y 16 mil millones en gastos de desarrollo e investigación combinados) para ofrecer las mejores soluciones en su clase para nuestros clientes

### 3. Definición del problema

---

Este documento describe el prototipo para implementar una aplicación funcional *web* para enrolamiento de proveedores utilizando la herramienta de desarrollo BPEL de Oracle. Esta versión del documento se entrega posterior al prototipo inicial. También abarca problemas relacionados con la incorporación de BPEL con interChange. Debido a que el prototipo no se realizó en base a requerimientos específicos para el estado este documento los realiza de manera aproximada para el alcance de este esfuerzo y aplica la base de iCE para los datos.

#### 3.1 INTRODUCCIÓN

La función para este prototipo es el autoenrolamiento de proveedores. El prototipo está estructurado para simular las necesidades de una interfaz *web* y proporcionar un elemento de entrada de datos. Va a interactuar con la capa de negocios N-Tier de interChange y emplear capacidades de flujo de trabajo humano para la recepción de licencia y la aprobación de enrolamiento. Es un prototipo muy enfocado en intentar completar un hilo sencillo de enrolamiento mientras evita mucha de la amplitud necesaria para proporcionar dicha característica en el subsistema de proveedores. Las representaciones en Powerpoint del prototipo están disponibles en los Apéndices A a C.

El esfuerzo de prototipado intenta asumir la calidad a nivel de producción lo mejor posible. Las áreas como la definición de xsd, manejo de excepciones, el cronometraje de las actividades asíncronas procuran asumir dicho nivel de calidad.

#### 3.2 EQUIPO DE PROTOTIPADO BPEL

John Schleicher y Erik Rodríguez realizaron las actividades del esfuerzo de prototipado. Este esfuerzo de prototipado se realizó inicialmente en la ciudad de Plano, TX para posteriormente efectuarse en El Paso, TX.

#### 3.3 PLATAFORMA DE DESARROLLO

Se utilizaron los siguientes componentes para el prototipo:

- Oracle BPEL Java Developer 10.1.2 beta 3
- Oracle Process Manager 10.1.2 beta 3
- Oracle Lite database (para soportar el *Process Manager*)<sup>1</sup>
- Oracle 10g database (para soportar los procedimientos almacenados)
- Windows XP Professional

Aunque las versiones de producción de Oracle BPEL están disponibles, se eligió la versión beta del producto debido a la funcionalidad agregada supuestamente necesaria para completar el esfuerzo de prototipado. La versión 10.1.2 beta 3 incluye flujo de trabajo humano y adaptadores de base de datos requeridos para el enrolamiento de proveedores además de avances significativos en el IDE para manipular las variables XML.

---

<sup>1</sup> La función de *Process Manager* fue también verificada con la base de datos Oracle 10g.

Las versiones de producción habrían requerido el uso del IDE Eclipse el cual era la plataforma original de la compañía Collaxa.

#### 3.4 PLATAFORMA OBJETIVO PARA EL *PROCESS MANAGER*

El BPEL *Process Manager* está diseñado para la plataforma Solaris en la solución interChange. El esfuerzo de prototipado confirmó que el sistema a nivel de producción puede ejecutarse con la base de datos y el servidor núcleo. Actualmente se ejecuta en el servidor dsscsun0. La versión de producción no era compatible con JDev 10.1.2 beta 3 sin embargo y solo una actividad superficial podría ser verificada una vez que la configuración haya sido completada. La configuración verificada fue:

- Solaris 9
- Oracle 9i (9.2.0.4)
- OC4J (Contenedor Oracle para J2EE) en modo independiente (*standalone*).
- JDK 1.4.2\_07

Esta configuración no será la versión final de implementación. El producto OC4J fue obtenido para modo independiente, ya que ultimadamente BPEL descansará sobre un Servidor de Aplicaciones de Internet Oracle Edición Empresarial el cual incluye el Monitor de Actividades de Negocio de Oracle (*Business Activity Monitor*).

#### 3.5 MONITOREO DE ACTIVIDADES DE NEGOCIO

Como se mencionó anteriormente este es un componente del Servidor de Aplicaciones Oracle. Se encuentra actualmente en beta y se espera una versión definitiva en Junio o Julio del 2005. La versión beta del producto BAM debe ser adquirida todavía para su evaluación.

#### 3.6 INTERFAZ DEL PROVEEDOR

La capa de negocios iCE N-Tier que soporta el enrolamiento de proveedores fue utilizada como interfaz por el BPEL *Process Manager* para enrolar nuevos proveedores. Esto permite que toda la lógica de negocios empleada por el desarrollo de la NTier sea invocada durante el enrolamiento de proveedores basado en *web*. Esto requirió el desarrollo de la interfaz en la plataforma .NET para poder exponer la capa de negocios N-Tier como un WSDL. El desarrolló se realizó por el equipo BPEL.

#### 3.7 REUSABILIDAD DEL PROTOTIPO

Se tiene la intención de que este prototipo contenga elementos que puedan ser aplicados a la solución final. No se tiene la intención de que sea un prototipo desechable. Su enfoque es estrecho para el enrolamiento de proveedores en el sentido de que sigue un hilo completo para una actividad unitaria de enrolamiento de médicos (proporcionando la base del autoenrolamiento de proveedores<sup>2</sup>). Esta meta va a facilitar la definición de la arquitectura de soporte y la metodología para los futuros sistemas que sean implementados. Se tiene la intención de realizar los siguientes principios:

- Los XSDs y WSDLs tienen definiciones definidas para ambientes de producción.

---

<sup>2</sup> Esto no incluye la adición de localidades de servicio lo cual es considerado separado y complementario del enrolamiento básico.

- El manejo de excepciones de BPEL asume el paradigma de niveles de despliegue.
- Las interfaces desarrolladas NTier (WSDLs) asumen una estructura de nivel de despliegue.
- El manejo de tiempo de las llamadas asíncronas está implementado para asegurar la integridad del flujo BPEL

### 3.8 CONCLUSIÓN/RESUMEN EJECUTIVO

Las actividades de prototipo para enrolamiento de proveedores utilizando Oracle BPEL 10.1.2 Beta 3 confirmaron que este producto puede utilizarse como un componente complementario dentro de la solución de interChange<sup>3</sup>. Se recomienda que BPEL sea usado de manera incremental agregando nuevos servicios a interChange de acuerdo a como vaya evolucionando los servicios de red estándar en expansión así como los demandas dictadas por el cliente. Los productos BPEL son archivos jar basados en java y están dirigidos para su inclusión en el servidor Solaris dentro de interChange.

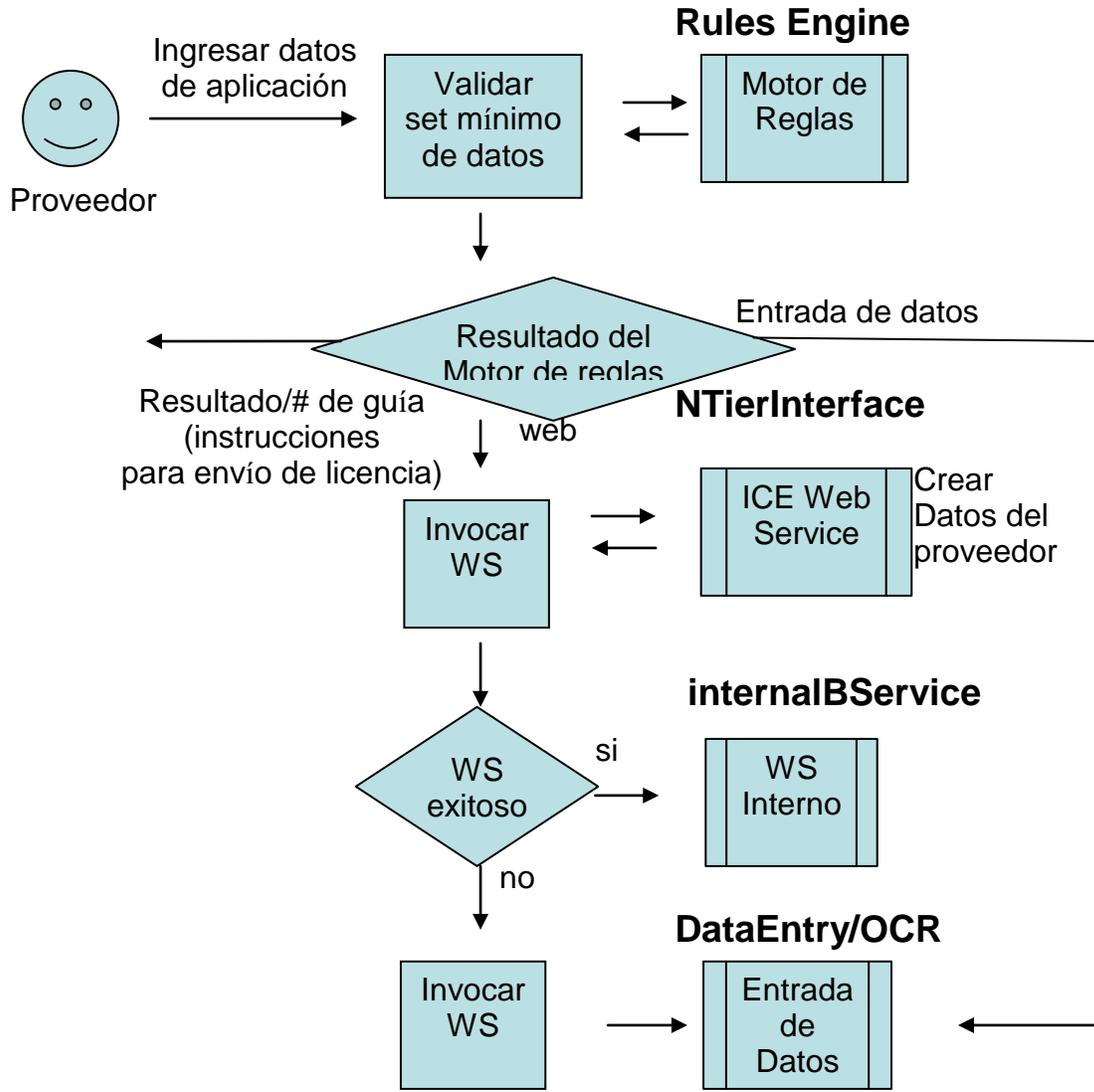
El producto beta BPEL que fue utilizado tiene dos componentes, el Administrador de Procesos que ejecuta los productos BPEL y el ambiente de desarrollo (IDE). El Administrador de Procesos es estable y el IDE tuvo falta de documentación y tenía algunos errores. Hubo algunas maneras de arreglar estos problemas del IDE. El producto se liberará en el verano y se recomienda esperar a éste para usarlo como plataforma de desarrollo para interChange debido a sus capacidades y al ambiente de desarrollo.

La introducción del componente BPEL a afectado nuestro proceso de desarrollo debido a la curva de aprendizaje, introducción a las prácticas de desarrollo, el control de código fuente, pruebas, así como la administración de la base de datos BPEL deben ser considerados. Se recomienda que se tomen pasos apropiados para definir la base apropiada para el desarrollo estructurado en BPEL debido a que se anticipa que BPEL crecerá de manera que lo que se defina ahora perdurará en futuros desarrollos. Las actividades durante este prototipo intentaron incorporar buenas prácticas para dicho fin.

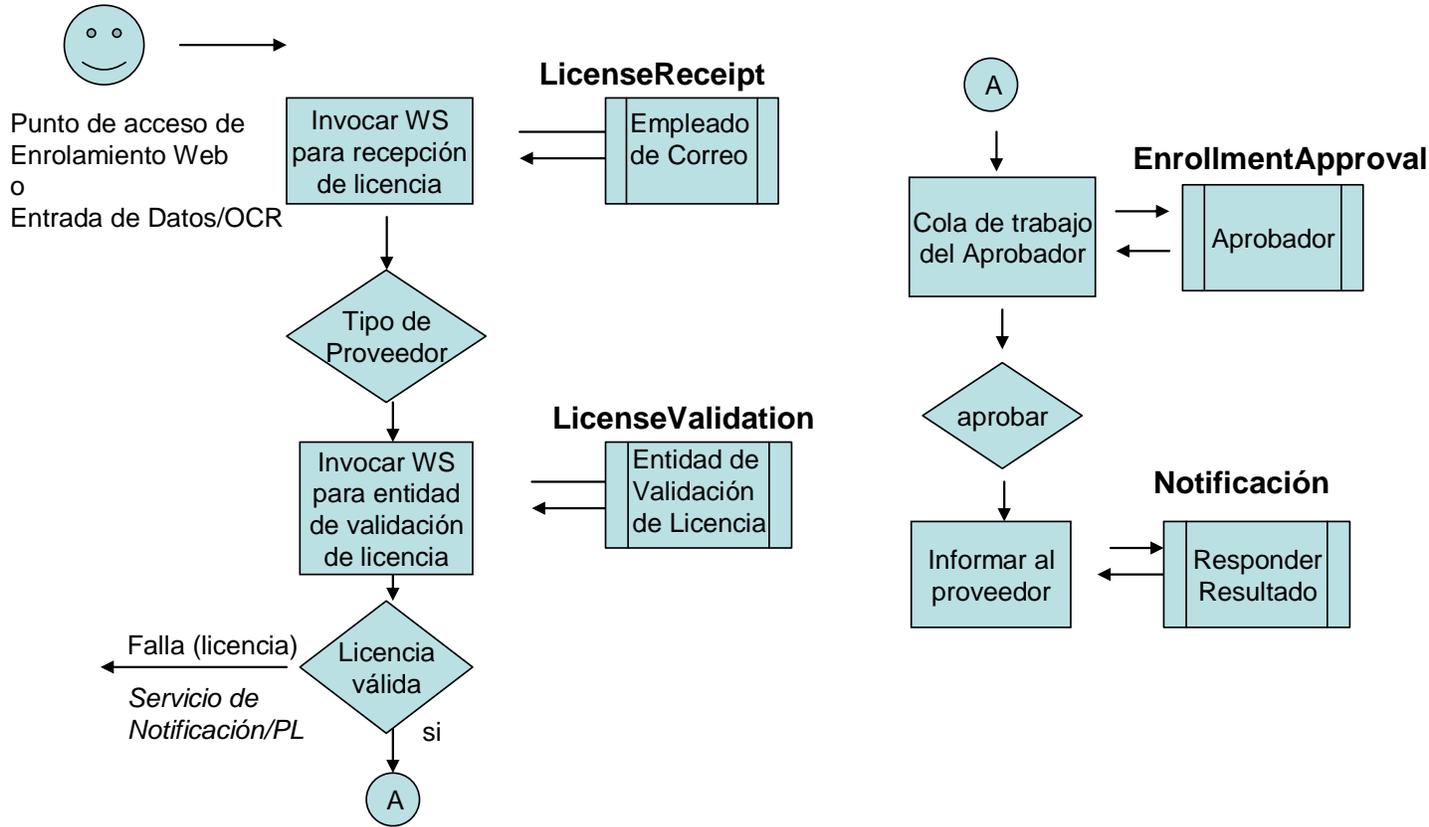
---

<sup>3</sup> BPEL no reemplazará componentes actuales ni afectara arquitecturas ya existentes.

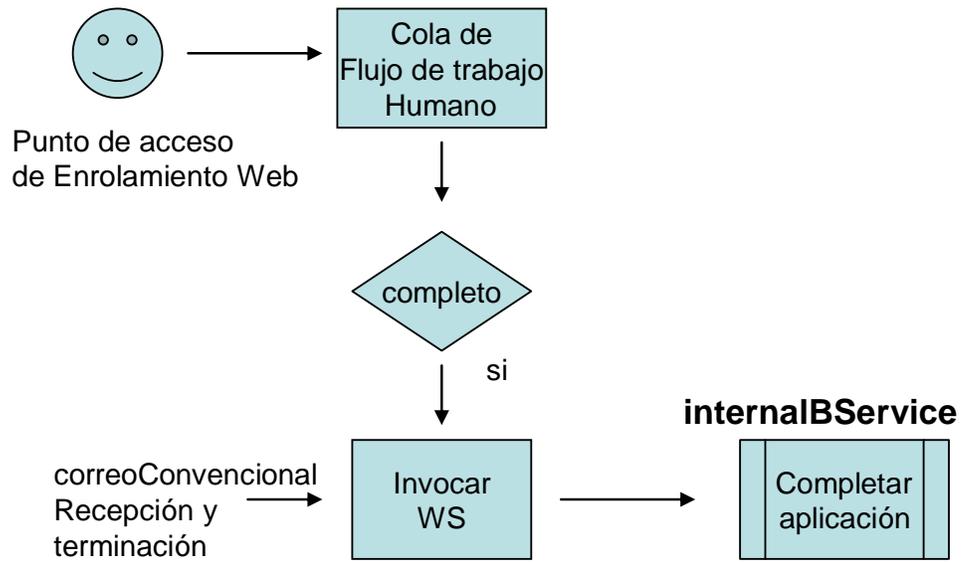
# Apéndice A: Web Service Externo



## Apéndice B: Servicio B Interno



## Apéndice C: Entrada de Datos/OCR



## **4 Análisis, Enrolamiento de Proveedores**

---

### **4.1 Descripción de requerimientos**

Una aplicación basada en *web* para el auto-enrolamiento de proveedores dentro del sistema de Medicaid.<sup>4</sup> Este documento también abarca las acciones de enrolamiento para un proveedor Médico. Este es un flujo de trabajo de principio a fin que incluye interfaces/actividades inter-agencia, elementos de trabajo humano, sincronización con elementos de papel, interacción con la capa de negocios NTier de la actividad de enrolamiento de proveedores, y finalmente la notificación de resultados del proceso hacia el proveedor. Ver Apéndices A y B para un diagrama del flujo de actividades.

### **4.2 RFP ID (n) Auto-enrolamiento de Proveedores**

Una aplicación basada en *web* que permite a los proveedores enrolarse en el sistema de Medicaid y reemplazar o complementar los sistemas basados en papel o fax que se encuentran actualmente en uso.

#### **4.2.1 Alcance**

Como se menciona arriba, el alcance es suplementario y debe integrarse con los procesos actuales y ofrecer puntos de entrada hacia el flujo de trabajo para actividades en sistemas actuales.

#### **4.2.2 BPEL Web Services**

Un mapeo directo de los requerimientos RFP para BPEL para los servicios necesarios para satisfacer los requerimientos explícitos.

##### **4.2.2.1 providerEnrollmentMain**

El punto de entrada principal para la actividad de auto-enrolamiento. Contiene la información básica de los proveedores como nombre, número de identificación del proveedor, número de licencia, dirección, contactos, etc. Se incluye por separado en servicios suplementarios la información acerca de localidades de servicio y especialidades.

### **4.3 BPEL Web Services Auxiliares**

Servicios adicionales que serían requeridos para complementar o redondear los requerimientos explícitos. Requerimientos inferidos.

#### **4.3.1 dataEntryOCR**

Servicio auxiliar que puede servir como punto de origen para el flujo de trabajo el cual permite la migración de procesos actuales y como un servicio alternativo para decisiones basadas en reglas y/o fallas en la entrada de datos automática dentro de la capa de negocios NTier. Este servicio permitirá y requerirá datos adjuntos OCR que serán incluidos dentro de los *web services*. Invocaciones basadas en NTier iniciarán los flujos BPEL al popular los datos de la interfaz WSDL e invocar el *web service* enrollB.

---

<sup>4</sup> Esta es una oración genérica no definitiva que no está asociada con ningún RFP

### **4.3.2 enrollB**

Este servicio auxiliary es un proceso únicamente interno que permite múltiples puntos de entrada dentro del flujo via los *web services* providerEnrollmentMain o alternativamente dataEntryOCR. Sirve como el flujo de control central para el proceso de enrolamiento de proveedores.

### **4.3.3 rulesEngine**

Este proceso es quien gobierna el procesamiento BPEL. Lo hace al permitir que un analista de negocios genere reglas para el procedimiento de enrolamiento. Se asume que esta interfaz está basada únicamente en datos entrantes (opuesto a una perspectiva completa histórica del proveedor o el enrolado pretendido). A este proceso se le presentarían todos los datos entrantes y cualesquiera datos generados por BPEL (como número de *tracking*) y en base a ésto decide en un intervalo muy corto de tiempo y de manera síncrona lo que el flujo de trabajo deberá llevar a cabo. La salida/respuesta necesita contener información suficiente para que el flujo de trabajo responda. Las posibles respuestas son: processElectronically (proceso BPEL), dataEntry (intervención humana), o falla. Este servicio sería invocado prematuramente en el servicio providerEnrollmentMain para dirigir ésta actividad.

### **4.3.4 NTierInterface**

Este servicio es una interfaz .NET que invoca la función de capa de negocio al agregar a un proveedor dentro del sistema. Debido a que la actividad no ha sido aprobada en esta etapa, los datos persistidos quedan marcados como pendientes y no son adecuados para continuar la actividad de proveedor a menos de que el sistema o alguno de los agentes actúen sobre ellos.

### **4.3.5 licenseReceipt**

Este servicio de flujo de trabajo humano se utiliza para “estacionar” la petición *web* mientras la licencia del proveedor se recibe. Este servicio incorpora un objetivo de grupo para que múltiples usuarios puedan capturar el elemento de trabajo y procesarlo. Esta actividad es de bajo nivel, e incorpora renovación contra escalamiento. El tiempo se asume como largo, de una semana o más. El contenido de la página de trabajo no requiere sofisticación excesiva para hacer corresponder el nombre con la aplicación. El título de la lista de trabajo incorpora un número de guía para ayudar al empleado que recibe el correo a verificar la licencia con la aplicación hecha por *web*. El *web service* regresaría un error si la licencia no es recibida o si ninguno de los usuarios especificados ejecuta alguna acción sobre este servicio. Los comentarios se pueden generar a través de la actividad de lista de trabajo licenseReceipt. Estos deben ser capturados y finalmente actualizan las tables de datos para enrolamiento de proveedores asociadas con los comentarios de la actividad de enrolamiento.

### **4.3.6 licenseValidation**

Este *web service* depende mucho de como sean implementados los procesos que se invocan desde este servicio. Algunos sistemas incorporan entidades externas para verificar las licencias de los proveedores. Para estos sistemas es necesario entender las interfaces de los sistemas destino y potencialmente capturar sus WSDLs para poder invocarlos desde *web services* o emplear adaptadores para comunicarse con estos sistemas. Cuando el personal interno lleva a cabo la validación de la licencia este servicio incorpora el mecanismo de flujo de trabajo humano. Un grupo seleccionado de usuarios recibiría una tarea con un intervalo que cuenta con un tiempo extenso para su ejecución. Otra actividad posible es el escalamiento hacia un gerente. Para los

propósitos del desarrollo CORE este *partner link* al flujo BPEL es una simple interfaz síncrona con un resultado de válido/inválido basado en algún elemento de la entrada del usuario como una etiqueta de ‘demostración’.

#### **4.3.7 enrollmentApproval**

El proceso de enrolamiento principal es revisado por una persona conocedora que aprueba dicho proceso antes de que sea activado en el sistema. Este es otro elemento de flujo de trabajo humano que potencialmente está dirigido a un grupo de aprobación para su ejecución. El escalamiento automático por intervalo terminado o por un miembro del grupo hacia una autoridad gerencial es posible a través de acciones del flujo de trabajo. Se anticipa que el tiempo para esta actividad sería de 72 horas antes del escalamiento.

#### **4.3.8 notificationService**

Esta actividad es responsable de notificar a quien lo invoca de la disposición final del servicio asíncrono. Esto incluye el servicio de enrolamiento de proveedores y el servicio para agregar localidades de servicio. Este servicio inicial de notificación está estructurado de manera genérica para poder ser aplica a cualquier actividad BPEL asíncrona. Debido a los problemas con privacidad y apelaciones potenciales la respuesta actual debe limitarse a una carta o fax. Las respuestas por correo electrónico (si son solicitadas por el usuario) deben contener solamente una respuesta interina indicando el éxito o falla de la petición, una referencia hacia la carta que pronto será recibida e información de contacto en caso de la carta no sea recibida como se programó. Ningún tipo de información personal o crítica debe ser comunicada.

#### **4.3.9 addServiceLocation**

Este *web service* tiene dos opciones de entrada. Puede ser invocado en un ciclo desde el *web service providerEnrollmentMain*, o invocado como una operación independiente para soportar un proveedor establecido al agregar una localidad de servicio. Para el punto de entrada *providerEnrollmentMain*, múltiples localidades de servicio son posibles y el número de iteraciones es dinámico. El tipo de proveedor y su especialidad son asociados con las localidades de servicio. El proveedor especifica un derechohabiente básica al que desea server. La asignación del programa para el proveedor se difiere hacia la persona que aprueba el enrolamiento. Una entrada a través de un *web service* externo requiere una respuesta intermedia.

#### **4.3.10enrollStatusCheck**

Este *web service* se proporciona para permitir que los proveedores verifiquen el estatus de su aplicación y potencialmente para obtener información de contacto para seguimiento personal. Este es un servicio síncrono que proporciona toda la retroalimentación en la respuesta sin requerimientos para la Notificación.

#### **4.3.11displayProviderData**

Este servicio permite al proveedor ver sus datos en el sistema de Medicaid. Los campos de comentarios que estén en la base de datos no se devuelven al proveedor que invoca el servicio. Este es un servicio síncrono que proporciona toda la retroalimentación en la respuesta sin algún requerimiento de Notificación.

### **4.4 BUSINESS PROCESSES**

Utilizamos los procesos de negocio de Arkansas debido a que ellos han implementado recientemente un sistema *web* de enrolamiento de proveedores.

#### **4.4.1 Proceso de Negocio A**

EDS absorbió las operaciones de enrolamiento de proveedores de Arkansas y está implementando un sistema de enrolamiento de proveedores basado en *web* que imita el sistema actualmente en uso. Básicamente al proveedor se le presentan las pantallas internas que se le presentaban anteriormente a las personas internas encargadas del ingreso de datos para transferir los datos desde una aplicación de papel.

Las páginas *web* son una operación de un solo paso que de manera selectiva ofrecen pantallas subsecuentes basadas en las selecciones previamente ingresadas durante el proceso. Las especialidades se ofrecen basándose en el tipo de proveedor ingresado, número de identificación de medicare, información de grupo, depósito bancario, etc... y las mismas se presentan de una manera condicional. Toda la información de localidades de servicio se incluye en la aplicación de una sola vez.

El proceso finalmente permite la generación de una aplicación la cual se requiere impresa por el proveedor, firmada y enviada por correo con su relativa licencia. Los datos actuales ingresados por el proveedor son, sin embargo, lo que se guarda en el sistema (nunca se vuelve a ingresar).

#### **4.4.2 Rules Engine**

Una descripción de la interacción relacionada con reglas estipulada por los requerimientos o inferida a partir de éstos.

#### **4.4.3 Human Workflow Elements**

Requerimientos/conocimiento de negocio que identifica como el flujo de trabajo humano es llevado a cabo para ser asociados con los modelos de flujo de trabajo humano de BPEL.

#### **4.4.4 NTier Business Layer Interaction**

Cuándo y cómo interactúan los *web services* BPEL con las interfaces de la capa de negocios NTier.

### **4.5 SUBSISTEMAS IMPACTADOS**

Los *web services* de enrolamiento de proveedores deben interactuar con el subsistema de proveedores a manera de cliente. No se anticipan cambios dentro del modelo de datos del subsistema los programas de proveedores. Algunos cambios menores podrían ser necesarios para mantener una correspondencia entre las acciones de aprobación de la NTier y las acciones de aprobación de la lista de trabajo de enrolamiento de BPEL. También las pantallas de entrada de datos NTier necesitan iniciar un flujo BPEL.

### **4.6 INTERFAZ DE USUARIO**

La actividad de lista de trabajo asociada con el enrolamiento de proveedores apalanca las características de lista de trabajo de Oracle debido a que son robustas y proporcionan las características necesarias para administrar trabajo controlable dirigido para compleción humana. Las interfaces de lista de

trabajo humana deben ser migradas de JSPs al ambiente .NET. Esta es la lista de tareas de las tareas pendientes más cualesquiera pantallas subsecuentes asociadas con el flujo de trabajo. Esto requiere utilizar las APIs de la lista de trabajo BPEL para crear un *web service* que sea invocada por el marco de trabajo DotNetNuke para su despliegue dentro de ese ambiente. Se asume lo siguiente acerca de las actividades de la lista de trabajo:

- Los usuarios no necesitan firmarse en la lista de trabajo una vez que se han firmado en DotNetNuke (su identidad y contraseña son conocidos)
- Las listas de trabajo se despliegan dentro de una pantalla relacionada con el usuario (desde un menú *pull down* asociado con la actividad personal del usuario)
- Al seleccionar un elemento de la lista de trabajo produce una pantalla de actividad apropiada DotNetNuke de una manera estándar para un ambiente *web*
- La compleción de una pantalla de actividad DotNetNuke invoca el API de lista de trabajo correspondiente que engloba la tarea y de manera apropiada reactiva el flujo BPEL
- Todas las acciones de lista de trabajo (aceptar, rechazar, escalar...)
- deben ser proporcionadas por la actividad basada en DotNetNuke
- Las actividades de lista de trabajo dan soporte a los grupos y roles a través del API y el servicio de identidad

## **4.7 dataEntryOCR**

### **4.7.1 WorkList**

La lista de trabajo data entry/OCR debe contener la siguiente información:

- Nombre del enrolado
- Prioridad
- Status
- Asignado a
- Fecha de expiración
- Fecha de recepción
- Acciones

### **4.7.2 Contenido GUI**

El contenido de entrada de datos debe incorporar la página inicial de entrada de datos de la capa de negocios NTier para este proveedor. El flujo normal de la página de entrada de datos no se ve afectado por la actividad de lista de trabajo. Si la actividad de la página de proveedor incluye los botones/acciones “OK” o “Submit” estos deben ser asociados con una acción “accept” de esta lista de trabajo. Ninguna acción debe ser aceptada, aun con actualizaciones en los datos, sino después de haber seleccionado el botón “Submit”. Las acciones especiales para lista de trabajo pueden ser relegadas al campo de acción de lista de trabajo dejando la pantalla de DotNetNuke para entrada de datos sin afectar.

## **4.8 licenseReceipt**

### **4.8.1 WorkList**

La lista de trabajo para recepción de licencia debe contener la siguiente información:

- Número de guía de la aplicación
- Prioridad
- Status
- Asignado a
- Fecha de expiración
- Fecha de recepción
- Acciones

### **4.8.2 Contenido GUI**

La presentación de esta GUI no requiere sofisticación y solamente necesita que el nombre o algún otro identificador sea presentado para poder hacer corresponder la licencia con la aplicación hecha en el *web*. El campo genérico de comentario de la lista de trabajo es requerido y cuando la información se ingresa aquí debe ser persistida en las tablas de enrolamiento de proveedores. Debido a que ésta es una nueva pantalla las acciones de la lista de trabajo deben estar disponibles desde la pantalla de contenido de la GUI.

## **4.9 enrollmentApproval**

### **4.9.1 WorkList**

La lista de trabajo de aprobación de enrolamiento debe contener la siguiente información:

- Nombre del enrolado
- Prioridad
- Status
- Asignado a
- Fecha de expiración
- Fecha de recepción
- Acciones

### **4.9.2 Contenido GUI**

El contenido para aprobación de enrolamiento debe incorporar la página inicial de enrolamiento de proveedores de la capa de negocios NTier. El flujo normal de la página de entrada de datos no se ve afectado por la actividad de la lista de trabajo. Si la actividad de la página de proveedores incluye un "OK" o un "Submit" entonces tiene que ser insertada dentro de la acción "accept" de la lista de trabajo. Ninguna acción debe ser posible aun cuando existan actualizaciones a los datos pero sin "Submit". Las acciones de la lista de trabajo especializada pueden ser relegadas al campo de acción de lista de trabajo dejando sin afectar la pantalla de DotNetNuke para la aprobación de enrolamiento.

## **4.10 Notificaciones de la lista de trabajo**

Estas notificaciones son opcionales dependiendo del usuario. Se proporciona un mecanismo para que a través de las pantallas DotNetNuke, un usuario

pueda inscribirse o desinscribirse de las notificaciones de la lista de trabajo. Esto debe realizarse desde el área del usuario donde las actividades de la lista de trabajo se recogen.

#### **4.11 SEGURIDAD**

El acceso a todos los *web services* de enrolamiento de proveedores deben seguir los procedimientos de autenticación a través del portal *web*. Este requerimiento no impacta el desarrollo de BPEL pero debe ser coordinado con el desarrollo del portal.

Se asume que todas las transacciones se efectúan dentro del *firewall* de Medicaid y no requieren cifrado.

#### **4.12 MEDICIONES**

El monitoreo de actividades de negocio reflejan las buenas prácticas de BPEL. Ningún requerimiento adicional de medicion ha sido definido.

#### **4.13 RESTRICCIONES DE RENDIMIENTO**

Las respuestas a los *web services* deben asumir un operador humano y deben ocurrir de una manera expedita siguiendo las prácticas aceptadas de BPEL (o de la GUI). Todos los *web services* externos asíncronos deben incluir una respuesta síncrona que indique el receptor de la petición y cualesquiera actividades de seguimiento requeridas por el usuario.

#### **4.14 ADAPTADORES**

La actividad de enrolamiento de proveedores tiene una interfaz con la base de datos para obtener un número único de guía. Esto debe ser logrado a través de un adaptador hacia un procedimiento almacenado.

#### **4.15 CONSIDERACIONES DE PRUEBAS**

Debido a que la actividad de enrolamiento de proveedores es una estructura estratificada de *web services* cada elemento de la estructura puede ser probada independientemente. Las pruebas unitarias pueden constituirse por el componente de la capa más baja. Los servicios de la capa más alta pueden constituir las pruebas de sistema. Debido a que esta es la primera iteración de BPEL no hay infraestructura para facilitar las pruebas. La consola BPEL proporciona funcionalidad básica para iniciar *web services* de una manera manual. Esta utilería se puede utilizar como herramienta para hacer las pruebas unitarias. Las pruebas unitarias a nivel de Diseño deben utilizar herramientas suplementarias de la interfaz de consola BPEL durante la fase de pruebas unitarias que puedan después ser pasadas a las pruebas de sistema.

#### **4.16 Utilerías de Prueba**

Se requiere desarrollar las siguientes utilerías para auxiliar en las pruebas automatizadas.

##### **4.16.1 Utilería de Invocación Web**

Una utilería basada en Java para invocar *web services* desde un archivo.

#### **4.16.2 Evaluador de respuestas**

Una utilidad basada en Java que lleve a cabo una comparación entre la salida de prueba y la salida esperada.

#### **4.17 Pruebas de Aceptación**

Cada componente de diseño debe identificar casos de prueba que se puedan utilizar como “pruebas de aceptación” de manera que el pasar dichas pruebas satisfactoriamente indique que el componente es adecuado para ser evaluado por el grupo de pruebas de sistema. Estas pruebas deben convertirse en parte de una biblioteca de pruebas de regresión y deben estar basadas en las utilidades automatizadas identificadas anteriormente.

#### **4.18 COMPONENTES DE DISEÑO**

Los siguientes componentes de diseño están identificados para dar soporte a la infraestructura interChange BPEL y los *web services* de enrolamiento de proveedores.

#### **4.19 Definición del XSD**

Este componente va a ser responsable de la definición del XSD para todas las actividades BPEL de enrolamiento de proveedores para asegurar la consistencia y conformidad con las prácticas aceptadas de BPEL. La parte responsable deberá solicitar requerimientos de definición de los diferentes componentes de diseño y tenerlos disponibles de una manera oportuna para estos servicios. Este componente de diseño toma la responsabilidad de la interfaz del motor de reglas.

##### **4.19.1 Asociación de requerimientos**

Asociación inferida para todos los *web services* BPEL

##### **4.19.2 Estimados de Diseño**

El estimado para la producción de los XSDs para todos los *web services* es de 40 horas/hombre.

#### **4.20 Servicio de Identidad BPEL**

Este es un componente de infraestructura necesario para cualquier *web service* que utilice una lista de trabajo. La capa por omisión JAZN que soporta el servicio de identidad de BPEL debe ser reemplazada por un *plug-in* personalizado que accesa un *web service* común que proporciona una función de servicio de identidad basada en LDAP. El *web service* proporciona toda la información necesaria de grupos y roles que no contiene suscripción a notificaciones que tendrá que ser sustituida de otra manera.

La capa de negocios NTier debe proporcionar acceso a la información de suscripción y proporcionar también un cronómetro configurable para el flujo de trabajo humano.

##### **4.20.1 Asociación de Requerimientos**

Asociación inferida de todos los *web services* BPEL que requieran una lista de trabajo.

## **4.20.2 Estimado de Diseño**

Las clases Java BPMIdentity, BPMGroup, y BPMUser deben ser extendidas para dar soporte al modelo de datos de interChange. El estimado de diseño aproximado es de 80 horas/hombre, asociados con el desarrollo de las clases Java.

## **4.21 Servicio de Notificación BPEL**

Este componente de diseño es también un elemento de infraestructura necesario para dar soporte a BPEL dentro de interChange. Como se menciona anteriormente debe proporcionar notificaciones a través de carta y fax con notificaciones interinas por correo electrónico.

El servicio de Notificación BPEL proporciona la característica actual de generación de cartas interChange el cual da soporte a cartas y fax y tiene una infraestructura completa para registrar la actividad de notificaciones. Para CORE no se sabe si existe la funcionalidad de generación de plantillas de cartas. Esta es una actividad regularmente muy específica dependiendo del estado y puede no estar disponible para el sistema CORE. Esta actividad de diseño no tiene que ser concerniente al contenido de las notificaciones pero debe tener la posibilidad de activar las notificaciones de interChange y demostrar que cada tipo es soportado para el éxito o falla de esta actividad. Cualquiera texto programable debe estar previamente llenado dentro de las tablas de interChange para soportar la generación de cartas y poder probar que esta capacidad está presente.

Si se solicita por le usuario, una respuesta interina debe ser generada por correo electrónico. El contenido principal de este correo debe ser obtenido desde un archivo con los específicos datos para éxito o falla y el tipo de petición. El notificación interina por correo electrónico NO debe contener información crítica y debe presentar la siguiente información:

- Número de guía de la petición
- Nombre de quien la origina
- Estatus de la petición
- Información de cuando esperar el recibo de la siguiente carta
- Información de contacto en caso de que la carta no sea recibida en el tiempo establecido

Cuando esta actividad es direccionada hacia una implementación actual de un estado donde existen mecanismos suficientes para generación de letras la notificación por correo electrónico puede incluir un URL dentro del sistema que muestre una version pdf de la carta generada. Para este caso sería necesario asegurarse de que la carta y el URL estén disponibles.

### **4.21.1 Asociación de Requerimientos**

Asociación inferida de todos los *web services* BPEL. Este es el camino para la respuesta a la petición asíncrona.

### 4.21.2 Estimado de Diseño

Flujo BPEL estimado a 40 horas para completar el flujo genérico necesario para dar soporte a todas las posibles respuestas y clientes<sup>5</sup>.

## 4.22 Interfaz de Componentes GUI

Este componente de diseño es responsable de todas las áreas de la interfaz GUI para dar soporte al enrolamiento de proveedores. Las interfaces identificadas incluyen:

- *Web Service* Enrolamiento de Proveedores
- *Web Service* Verificación de Estatus
- *Web Service* Desplegar Datos del Proveedor
- Inicio de Entrada de Datos para el flujo BPEL
- Interfaz para Lista de Trabajo de flujo de entrada de datos BPEL
- Interfaz para Lista de Trabajo Aprobación de Enrolamiento
- Página de contenido de Recepción de Licencia

Las páginas externas de *web services* deben ser construidas a través de pantallas .NET con los datos ensamblados de acuerdo al XSD. Una vez recibida la entrada ésta debe ser dirigida al BPEL Process Manager en Solaris. La salida del BPEL Process Manager debe ser transformada de regreso hacia la GUI que la originó y a través del servidor *firewall* NT. El inicio de la entrada de datos del flujo BPEL requiere que la pantalla de entrada de datos NTier inicie un flujo BPEL. También debe reconocer la presencia o ausencia de una licencia de manera que el *web service* `licenseReceipt` pueda ser omitido si la licencia ya se había proporcionado. Los elementos de la interfaz de lista de trabajo deben proporcionar una integración transparente de la página de contenido de la lista de trabajo y las páginas de entrada de la NTier para Enrolamiento de Proveedores. Deben integrar cualquier actividad NTier construida dentro del contenido de la lista de trabajo, asociada con éxito o falla y con el aceptar y rechazar de la lista de trabajo de BPEL. La página de contenido de recepción de licencia es una GUI simplemente consistente.

### 4.22.1 Asociación de Requerimientos

### 4.22.2 Estimados de Diseño

Se estiman 200 horas para la compleción de esta actividad.

## 4.23 API de la Lista de Trabajo

Este componente de diseño es responsable de extender las interfaces de usuario predefinidas en JSP en *web services* que empleen el API de lista de trabajo Oracle Java para que sean invocadas a través de DotNetNuke. Este diseño satisface los requerimientos de la sección 5.

---

<sup>5</sup> En otras palabras este proceso se convierte en un *web service* primitive utilizado por todos los *web services* BPEL actuales y subsecuentes que requieran notificaciones.

#### **4.23.1 Asociación de Requerimientos**

#### **4.23.2 Estimados de Diseño**

Se estiman 100 horas/hombre

#### **4.24 *Web Service* providerEnrollment**

Este componente de diseño constituye el flujo principal del proceso BPEL para enrolamiento de proveedores. Incluye el servicio subordinado enrollB.

#### **4.24.1 Asociación de Requerimientos**

#### **4.24.2 Estimados de Diseño**

Estimado a 20 horas/hombre

#### **4.25 *Web Service* enrollB**

Este *web service* interno es una extensión natural del flujo de enrolamiento de proveedores pero funciona como un punto de entrada común para dataEntryOCR y providerEnrollmentMain.

#### **4.25.1 Asociación de Requerimientos**

#### **4.25.2 Estimado de Diseño**

Estimado a 40 horas

#### **4.26 *Web Service* addServiceLocation?**

Este *Web Service* agrega localidades de servicio al proveedor, incorpora el tipo y especialidad del proveedor, y asocia los programas basados en proveedor a proveedor basados en la base de derechohabiente establecida.

#### **4.26.1 Asociación de Requerimientos**

#### **4.26.2 Estimado de Diseño**

Estimado en 80 horas/hombre.

#### **4.27 *Web Service* checkStatus**

Este *web service* permite que el proveedor verifique el estatus de su aplicación. Requiere consideración del modelo de datos para mantener u obtener el estatus de la aplicación en una nueva tabla del modelo de datos (la misma que soporte al número de guía de la aplicación) o algún tipo de mapeo de las tablas de auditoría de BPEL. Este es un servicio síncrono puro.

#### **4.27.1 Asociación de Requerimientos**

#### **4.27.2 Estimado de Diseño**

Se estima en 20 horas/hombre.

#### **4.28 *Web Service* Display Provider Data**

Este *web service* permite que el proveedor examine sus datos de Medicaid. Requiere una interfaz a las funciones de despliegue de la interfaz NTier para la obtención de datos. Este es un servicio síncrono puro.

#### **4.28.1 Asociación de Requerimientos**

#### **4.28.2 Estimado de Diseño**

Estimado en 40 horas/hombre

## 4.29 Suscripción a las Notificaciones de la Lista de Trabajo

Ver sección 5.4. Las notificaciones de la lista de trabajo simplemente se activan o desactivan con la excepción de que no se elimina la notificación de escalamiento de la lista de trabajo.

### 4.29.1 Asociación de Requerimientos

### 4.29.2 Estimado de Diseño

Se estiman 20 horas/hombre para la generación del proyecto BPEL. Se estiman 10 horas para la GUI que inicia el proceso BPEL.

## 4.30 TABLA DE MAPEO DE REQUERIMIENTOS

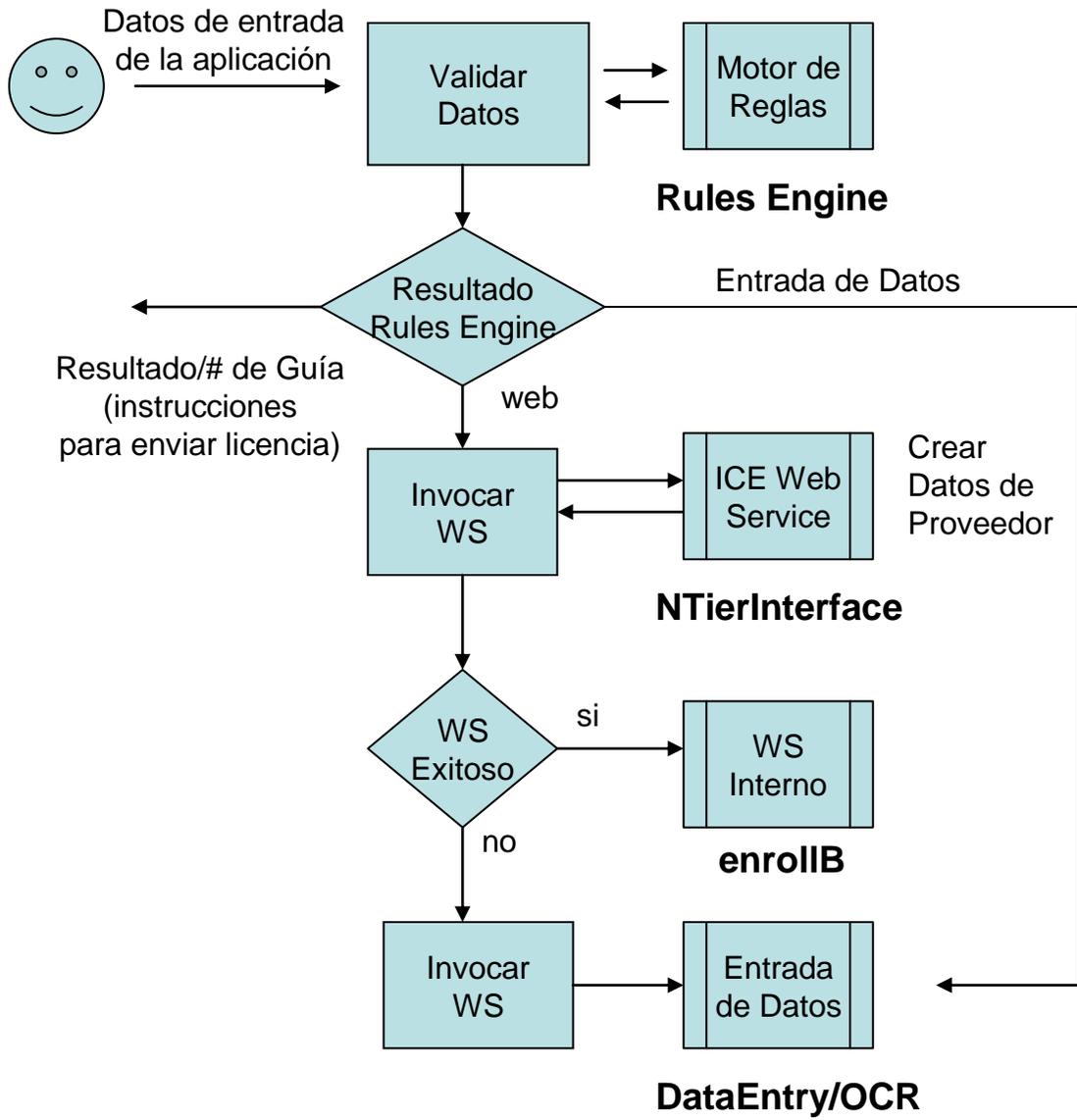
Para las capacidades iniciales de CORE para BPEL, este funciona como un simple marcador de posición. No hay RFPs ni requerimientos asociados con este documento de análisis.

<b>ID requerimiento</b>	<b>Descripción</b>	<b>E/I<sup>6</sup></b>	<b>Componente de Diseño</b>
NA	Construcción del XSD	I	11.1
NA	Modelo de datos del servicio de identidad de la lista de trabajo	I	11.2
NA	Plugin personalizados para el servicio de identidad de la lista de trabajo	I	11.2
NA	Asignación de grupos para el Servicio de Identidad	I	11.2
NA	Asignación de roles para el Servicio de Identidad	I	11.2
NA	<i>Web service</i> de Notificación	I	11.3
NA	Integración con cartas interChange	I	11.3
NA	Notificación interina por correo electrónico	I	11.3
NA	GUI web para Provider enrollment	I	11.4
NA	GUI web para Add service location	I	11.4
NA	GUI web para Check status	I	11.4
NA	Flujo de inicio de entrada de datos BPEL	I	11.4

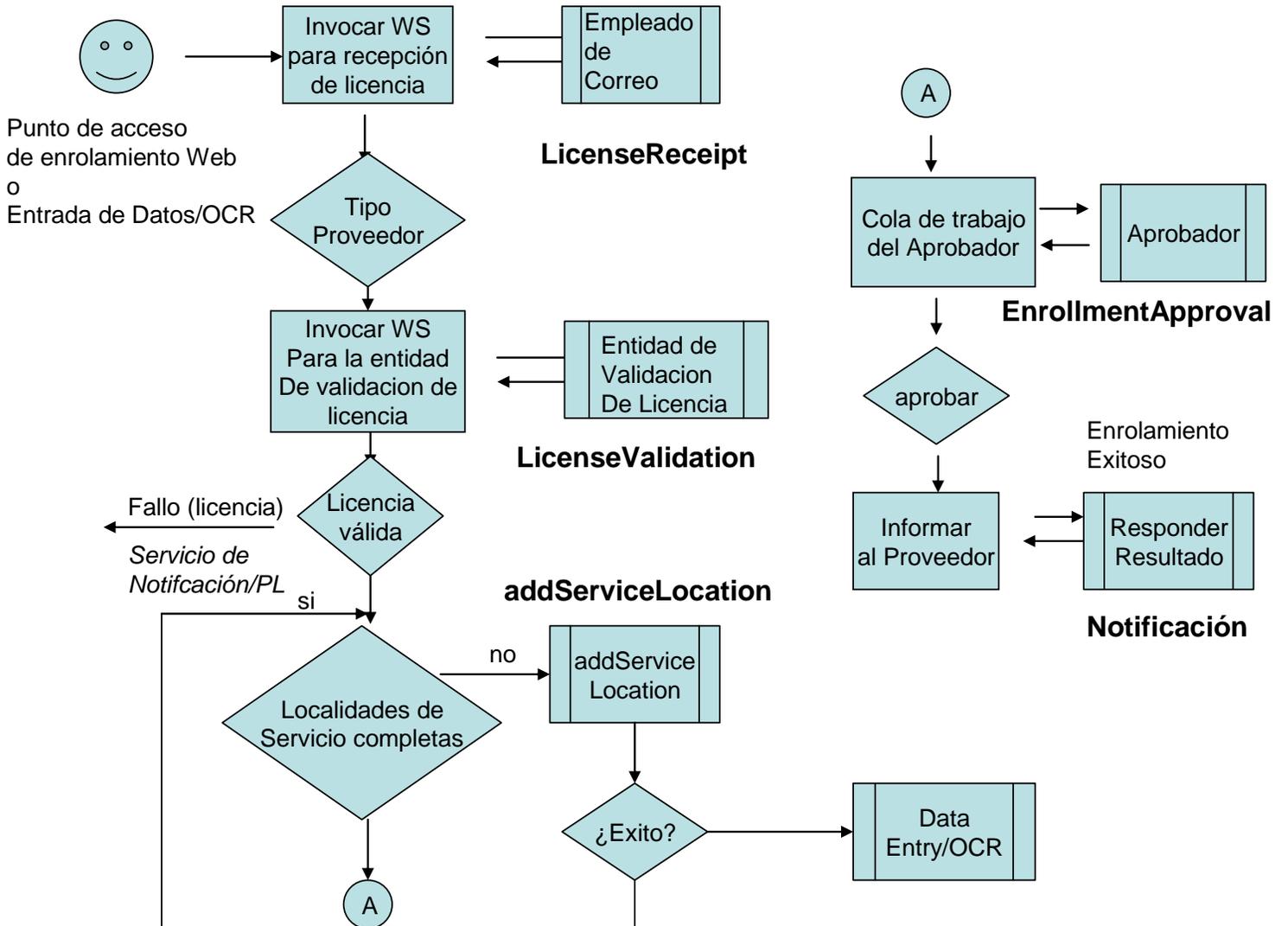
<sup>6</sup> Requerimiento explícito del RFP (E) o derivado implícitamente del RFP (I)

NA	GUI DotNetNuke para la lista de trabajo de entrada de datos	I	11.4
NA	GUI para Enrollment approval	I	11.4
NA	Contenido de recepción de la licencia	I	11.4
NA	GUI de la lista de trabajo DotNetNuke	I	11.4
NA	API de la lista de trabajo	I	11.5
NA	<i>Web service</i> Provider enrollment	I	11.6
NA	<i>Web service</i> interno enrollB	I	11.7
NA	<i>Web service</i> Add service location	I	11.8
NA	<i>Web service</i> Check status	I	11.9
NA	<i>Web service</i> Display provider	I	11.10
NA	Suscripción a notificaciones de la lista de trabajo	I	11.11
NA	Suscripción a la lista de trabajo de la GUI DotNetNuke	I	11.12

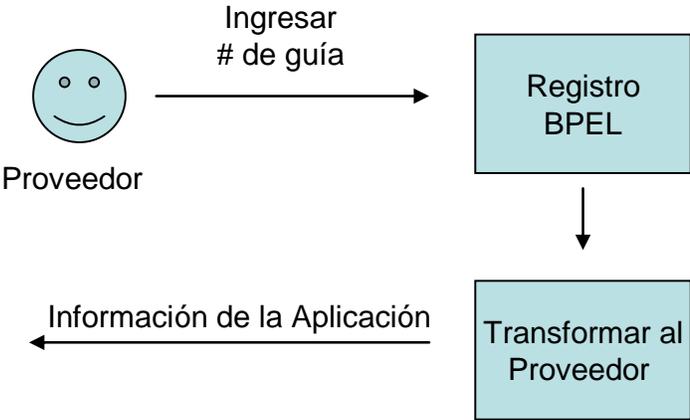
# Apéndice A: Web Service Enrolamiento de Proveedores



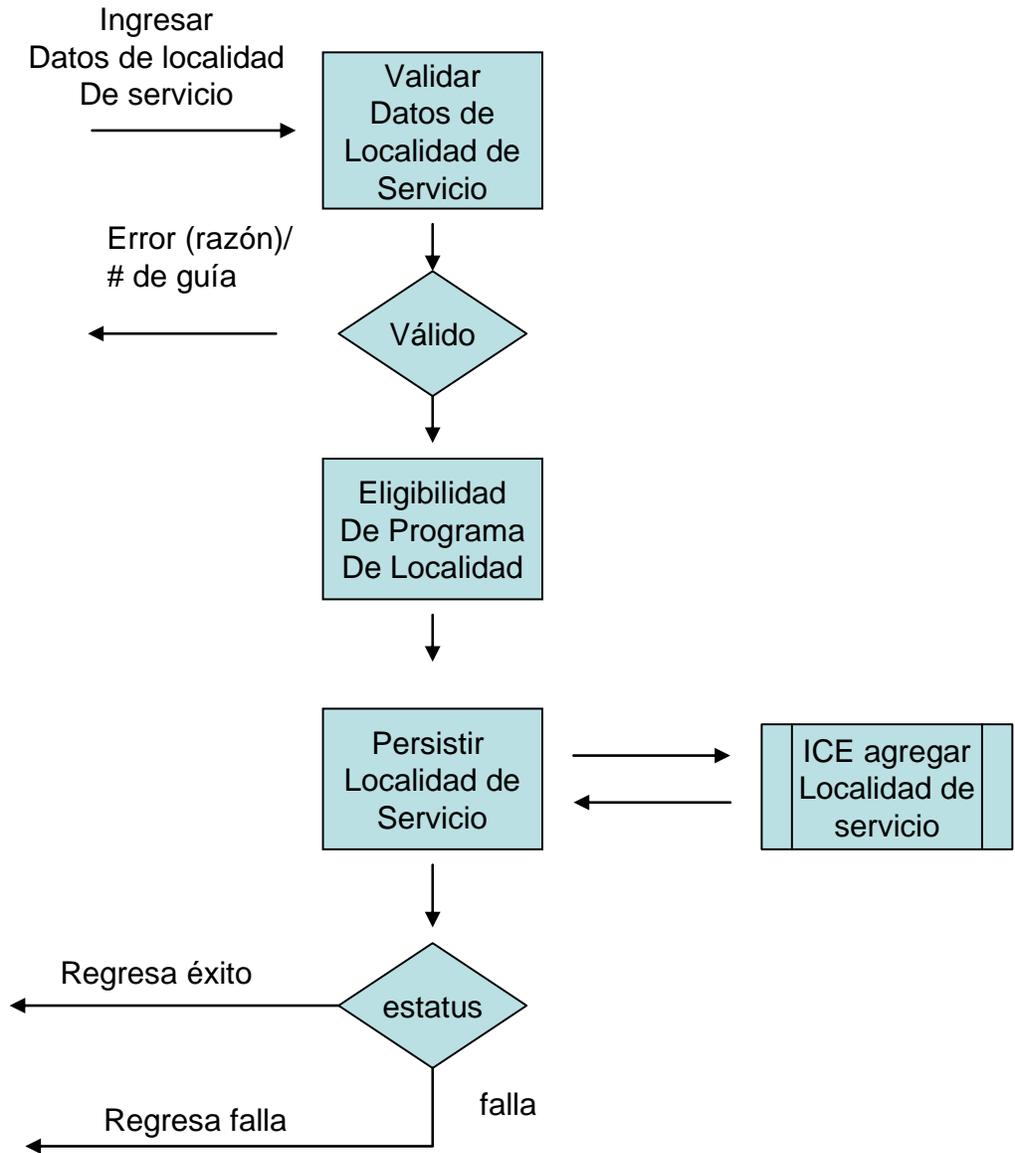
## Apéndice B: Web Service enrollB



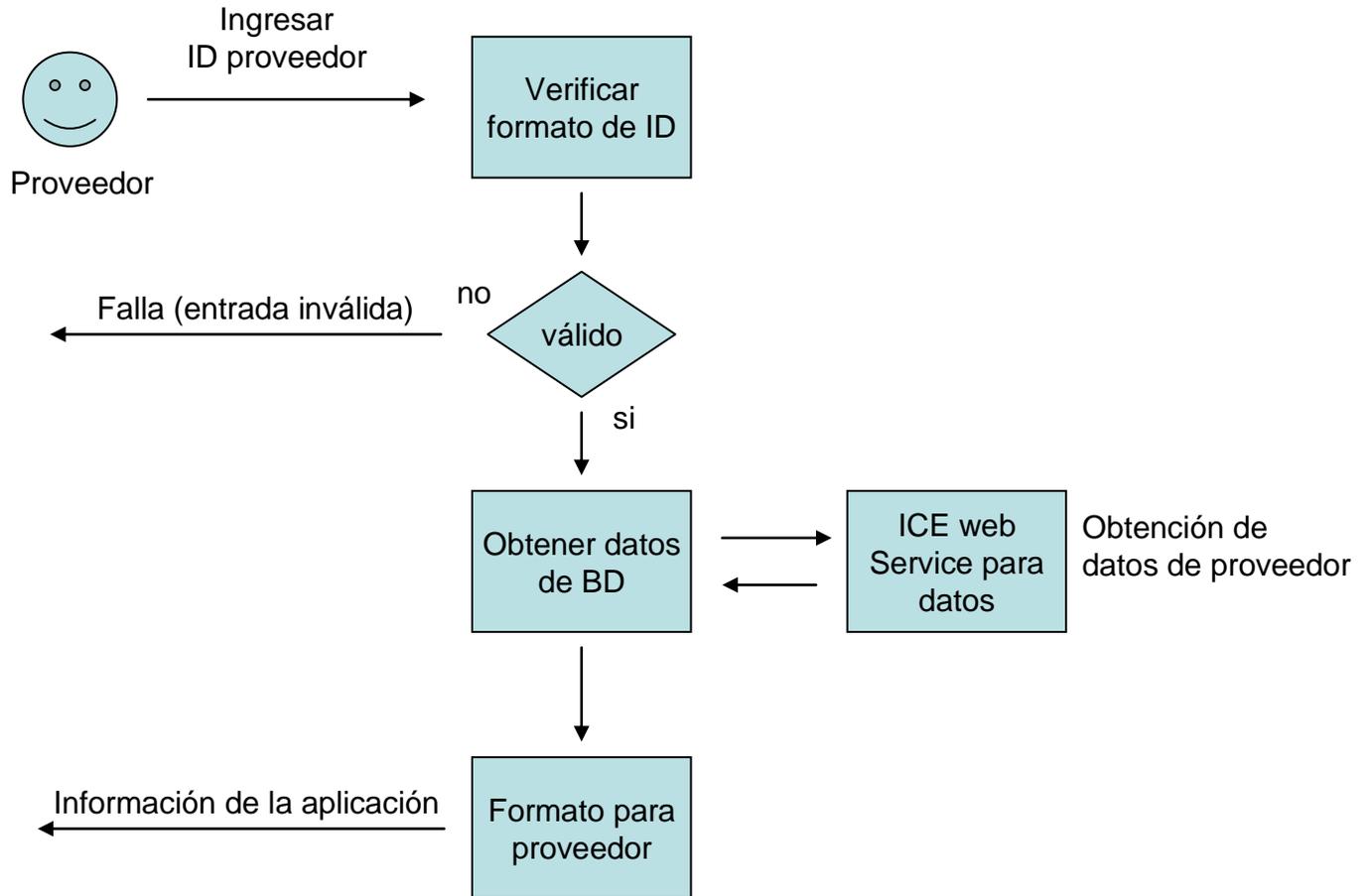
**Apéndice C: Web Service checkEnrollmentStatus (síncrono)**



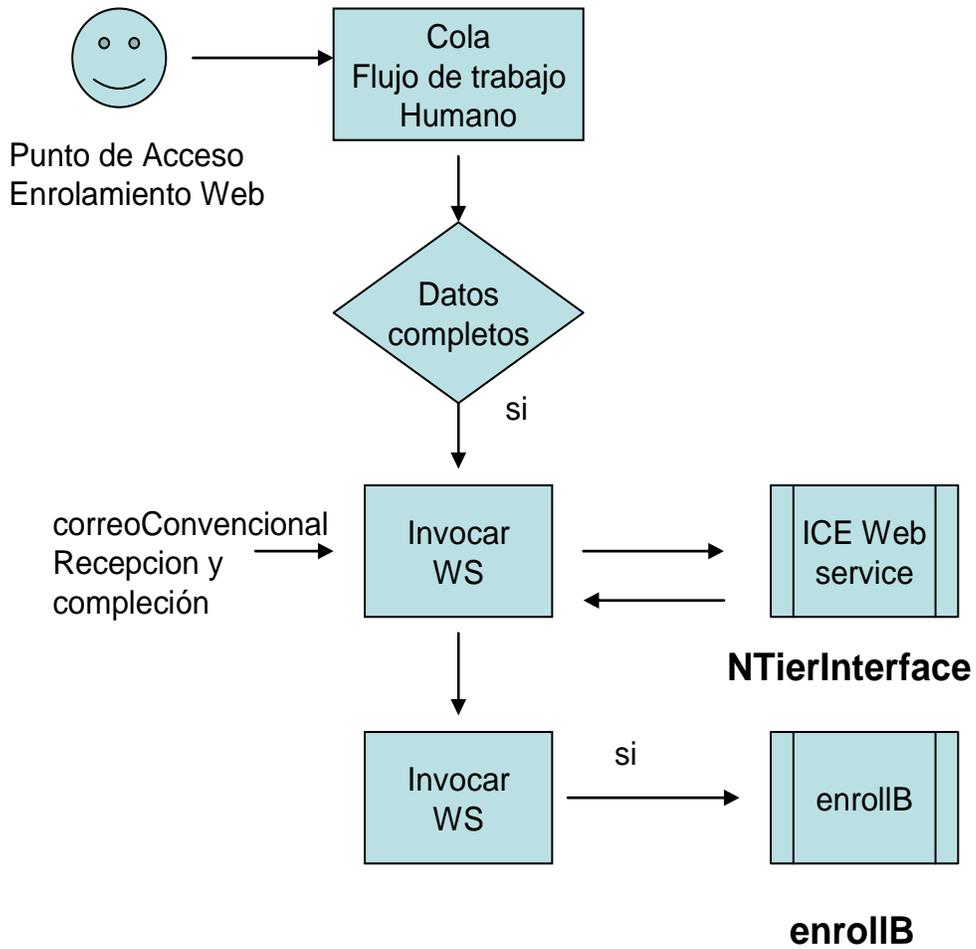
## Apéndice D: Web Service addServiceLocation



## Apéndice E: displayProviderData (síncrono)



## Apéndice F: DataEntry/OCR



## **5. Diseño ProviderEnrollmentMain**

---

### **5.1 DESCRIPCIÓN DEL DISEÑO**

El Proceso BPEL ProviderEnrollmentMain es el punto de entrada principal para la actividad de auto-enrolamiento. Contiene información de proveedor básica como nombre, número de identificación del proveedor, número de licencia, dirección, contactos, etc. La información acerca de las localidades de servicio y las especialidades se incluye como parte de otros servicios separados complementarios.

#### **5.1.1 Alcance del Diseño**

#### **5.1.2 Elemento de Diseño A**

Este proceso BPEL (ProviderEnrollmentMain, de ahora en adelante “PEM”) es una orquestación de la actividad de Enrolamiento de Proveedores. Como se mencionó anteriormente es el punto de entrada principal para la actividad de auto-enrolamiento. Después de que la entrada inicial ha sido ingresada, escribe alguna información a la tabla T\_BPEL\_TRACKING en la base de datos CFMIST1 utilizando el procedimiento almacenado SP\_BPEL\_ADD\_TRACKING para poder hacer un rastreo de la aplicación de enrolamiento. Este genera un número de guía. La información se envía al motor de reglas para validar la información de entrada. Dependiendo de la respuesta del motor de reglas, se invoca una interfaz NTier o la aplicación se rechaza, en cuyo caso PEM termina respondiendo con la salida correspondiente que se obtiene de la tabla T\_BPEL\_RSP\_TXT en CFMIST1 utilizando el procedimiento almacenado SP\_BPEL\_SELECT\_RESPONSE (cual referencia futura a “adquirir una respuesta administrable” puede asumirse como una llamada al procedimiento almacenado que acabamos de mencionar). La interfaz NTier se invoca si el motor de reglas declara que la aplicación puede ser procesada de manera electrónica o que la aplicación necesita ser enviada a Entrada de Datos. Si la interfaz NTier falla, PEM llamara a providerEnrollmentDataEntry (a partir de ahora PEDE) para manejar el flujo de humano requerido, seguido de adquirir la respuesta de T\_BPEL\_RSP\_TXT. De otra manera, PEM continuará y agregará una localidad(es) de servicio. Se pueden agregar hasta un máximo de 5 localidades de servicio, en las cuales PEM llama a addSvcLocNTierInterface para cada iteración. Si al agregar una localidad de servicio hay un error, PEM invoca PEDE, adquiere la respuesta administrable desde T\_BPEL\_RSP\_TXT, y responde de manera síncrona con la salida correspondiente. De manera adicional PEM también realiza esto si el motor de reglas regresa una respuesta “processDataEntry”. Sin embargo, si no hay fallas en las localidades de servicio, PEM invoca providerEnrollmentCommon (de aquí en adelante “PEC”), adquiere la respuesta administrable de T\_BPEL\_RSP\_TXT y responde de manera síncrona con la salida correspondiente. Ver Apéndice A para una representación gráfica del flujo BPEL PEM.

### 5.1.3 Flujo de Trabajo Humano

No hay flujo de trabajo humano directo en PEM. Hay, sin embargo, dos diferentes llamadas a procesos que requieren flujo de trabajo humano: (1) PEC: Después de que la aplicación ha viajado exitosamente a través del flujo PEM, se envía a PEC y (2) PEDE: En caso de que exista algún tipo de complicación con la aplicación de enrolamiento, PEDE es invocado. Sin embargo, estas son llamadas simples a dos procesos diferentes que contienen elementos de flujo de trabajo humano y no elementos de flujo de trabajo humano dentro de PEM.

### 5.1.4 Manejo de Excepciones

Ver Apéndice B para las localidades y descripciones de las excepciones en el flujo PEM.

### 5.1.5 Manejo de Tiempo

\*Todos los tiempos son especulativos

#### **invokeAddTracking**

**Descripción:** Esta es la llamada a la tabla T\_BPEL\_TRACKING en CFMIST1 para insertar la información de rastreo de los proveedores.

**¿Flujo de trabajo humano?:** No

**Tiempo:** Debido a que es solamente una inserción en una base de datos, no se requiere mucho tiempo para llevar esta acción a cabo. Por lo tanto, 5-10 segundos deben ser suficientes para determinar si la acción fue exitosa.

#### **invokeRulesEngine**

**Descripción:** Esta es una llamada al motor de reglas para determinar la elegibilidad básica de una aplicación.

**¿Flujo de trabajo humano?:** No

**Tiempo:** Esta es una llamada síncrona que no debe tomar más de 10 segundos.

#### **invokeDataEntry**

**Descripción:** Llamada a PEDE para mandar la aplicación a un encargado de entrada de datos (flujo de trabajo humano indirecto).

**¿Flujo de trabajo humano?:** No

**Tiempo:** Todo el tiempo ocurre del lado del *partner link*.

#### **invokeNTierInterface**

**Descripción:** Llamada a NTierInterface solo si el motor de reglas no responde con "Reject".

*¿Flujo de trabajo humano?:* No

*Tiempo:* El esquema de tiempo no debe ser mayor a 10 segundos.

### **invokeGetResponse**

*Descripción:* Llamada al procedimiento almacenado SP\_BPEL\_SELECT\_RESPONSE en AIM (CFMIST1) para adquirir la respuesta correspondiente que será presentada al usuario.

*¿Flujo de trabajo humano?:* No

*Tiempo:* Esta es una llamada a una base de datos. No debe requerir más de 5-10 segundos.

### **invokePEC**

*Descripción:* Llamada a providerEnrollmentCommon (PEC) posterior a la recepción de “processElectronically” por parte del motor de reglas, de que NTierInterface regresó “Success”, y addSvcLocNTier no tuvo fallas.

*¿Flujo de trabajo humano?:* No

*Tiempo:* Todo el tiempo ocurre del lado del *partner link*.

### **invokeAddSvcLocNTier**

*Descripción:* Llamada a addSvcLocNTierInterface después de recibir “Success” por parte de NTierInterface.

*¿Flujo de trabajo humano?:* No

*Tiempo:* Llamada síncrona, no se requieren más de 5-10 segundos.

## **5.1.6 Manejadores de Compensación**

Ninguno en este momento.

## **5.1.7 Interfaces NTier**

Hay dos llamadas hacia NTier Interfaces:

- (1) **baseNTierInterface:** Esta interfaz hace toda la persistencia de los datos de la aplicación siempre y cuando el motor de reglas no haya respondido “reject”.
- (2) **addSvcLocNTierInterface:** Esta interfaz toma todos los datos de localidad de servicio y los persiste en la base de datos.

## **5.1.8 Sensores**

Hay dos sensores en este proceso. Uno está localizado en el punto de entrada de este flujo BPEL, en la actividad de procesamiento receiveInput. Este sensor publica el providerName y el providerID que están contenidos en la variable inputVariable y los manda al Monitor de Actividades de Negocio (Business Activity Monitor ó BAM) y a una base de datos que almacene la información para

la generación de reportes. La segunda está localizado en la actividad de proceso replyOutput y publica la variable requestStatus (la cual contiene estatus y razón del estatus) hacia el BAM (aun no implementado) además de a la base de datos que guarda la información para la generación de reportes.

### **5.1.9 Alcance del Proceso BPEL**

En esta sección se incluye la discusión acerca del alcance del flujo de trabajo. La adherencia con las prácticas aceptadas BPEL no requiere mayor discusión.

### **5.1.10 Motor de Reglas**

La llamada al motor de reglas se hace directamente posterior a que se genera un número de guía para la aplicación. Al motor de reglas se le alimenta la información de la petición de Enrolamiento de Proveedores (ProviderEnrollmentRequest), junto con el tipo de petición indicando que es una aplicación *web*.

El motor de reglas analiza los datos y regresa alguna de las siguientes tres respuestas (“processElectronically”, “processDataEntry”, “rejectRequest”) las cuales le dan dirección al flujo BPEL PEM en alguno de los siguientes dos casos: (1) El caso de invocación a NTierInterface, que es la manera correcta de proseguir si el motor de reglas regreso “processElectronically” ó “processDataEntry”, ó (2) “rejection” y de manera consecuente falla en la aplicación.

## **5.2 XSD**

PEM utiliza los siguientes XSDs para funcionar adecuadamente:

**AIM\_SP\_BPEL\_ADD\_TRACKING.xsd** – Ver la sección 5.2.1 para los detalles de este elemento.

**AIM\_SP\_BPEL\_SELECT\_RESPONSE.xsd** – Ver la sección 5.2.1 para los detalles de este elemento.

**providerEnrollment.xsd** - Este XSD define las variables utilizadas por PEM. De hecho, este XSD define las variables usadas, esencialmente, por todo el espacio de trabajo de ProviderEnrollment.

### **5.2.1 Namespace**

<http://www.eds.com/ns/iC.MMIS.WS.Prov.Enrollment>

### **5.2.2 Localidad de los Esquemas**

interChangeCommon.xsd

interChangeRecipient.xsd

interChangeProvider.xsd

providerEnrollment.xsd

## **5.3 WSDL**

PEM requiere los siguientes WSDL para ser funcional:

**SP\_BPEL\_ADD\_TRACKING\_test.wsdl** – Este es el WSDL utilizado para la actividad de invocación invokeAddTracking que inserta la información de rastreo en la correspondiente tabla en la base de datos.

**SP\_BPEL\_SELECT\_RESPONSE\_test.wsdl** – Este es el WSDL utilizado para la actividad de invocación `invokeGetResponse` que toma la respuesta administrable de la correspondiente tabla en la base de datos.

## **5.4 TABLAS DE DATOS**

PEM utiliza las tablas de CFMIST1 bajo el *tablespace* AIM. Estas incluyen T\_BPEL\_TRACKING, T\_BPEL\_RSP\_TXT, y T\_BPEL\_WS\_RESPONSE, entre otras.

### **5.4.1 BPEL**

La tabla T\_BPEL\_TRACKING está en el *tablespace* AIM bajo CFMIST1. Esta tabla se alimenta utilizando el procedimiento almacenado SP\_BPEL\_ADD\_TRACKING. Además, el procedimiento almacenado SP\_BPEL\_SELECT\_RESPONSE utiliza varias tablas del *tablespace* AIM para obtener la respuesta administrable correcta (Ver Apéndice C para los detalles de ambos procedimientos almacenados).

**AIM\_SP\_BPEL\_ADD\_TRACKING.xsd** – Este XSD define el formato para agregar tuplas a la tabla T\_BPEL\_TRACKING utilizando el procedimiento almacenado SP\_BPEL\_ADD\_TRACKING.

**AIM\_SP\_BPEL\_SELECT\_RESPONSE.xsd** – Este XSD define la entrada requerida y la correspondiente salida del procedimiento almacenado SP\_BPEL\_SELECT\_RESPONSE cuando obtiene la respuesta administrable correcta de las tablas localizadas en el *tablespace* AIM en CFMIST1.

### **5.4.2 interChange**

Esta sección habla de las tablas dentro del sistema estándar interChange. Típicamente no se hace referencia a ellas fuera de las interfaces NTier.

## **5.5 SOBRECARGA DE RENDIMIENTO/SISTEMA**

Esto no ha sido determinado. Pruebas posteriores nos dirán los requerimientos para esta sección.

## **5.6 DEPENDENCIAS**

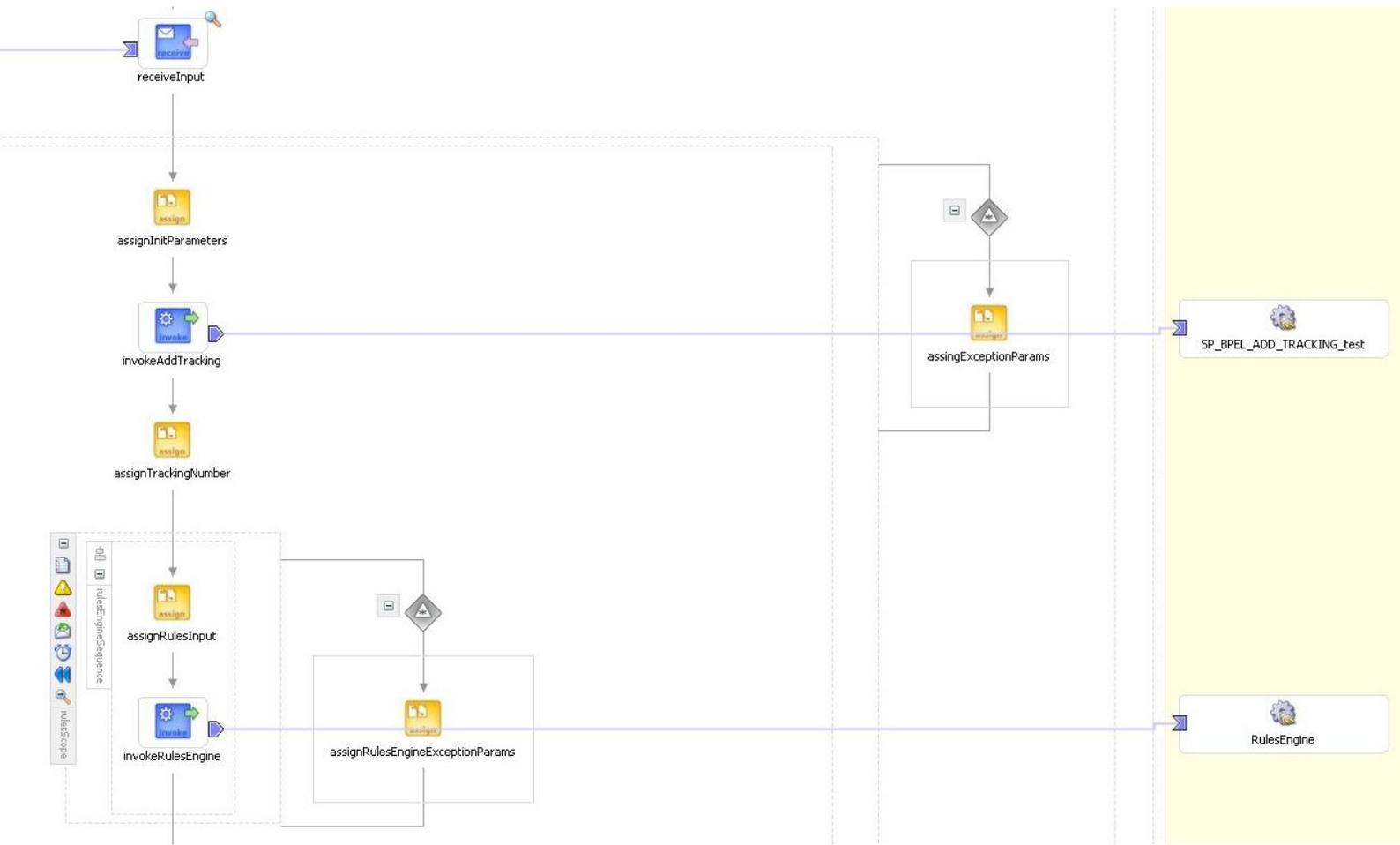
Debido a que PEM es en esencia el orquestador del proceso completo de Enrolamiento de Proveedores, para que PEM trabaje de manera efectiva, todos los demás procesos BPEL deben estar funcionando. Esto incluye PEDE, PEC, NTierInterfaces, y Motor de Reglas. Todas las tablas deben estar funcionando, con la información correcta y ser accesibles, además de que los procedimientos almacenados sean funcionales.

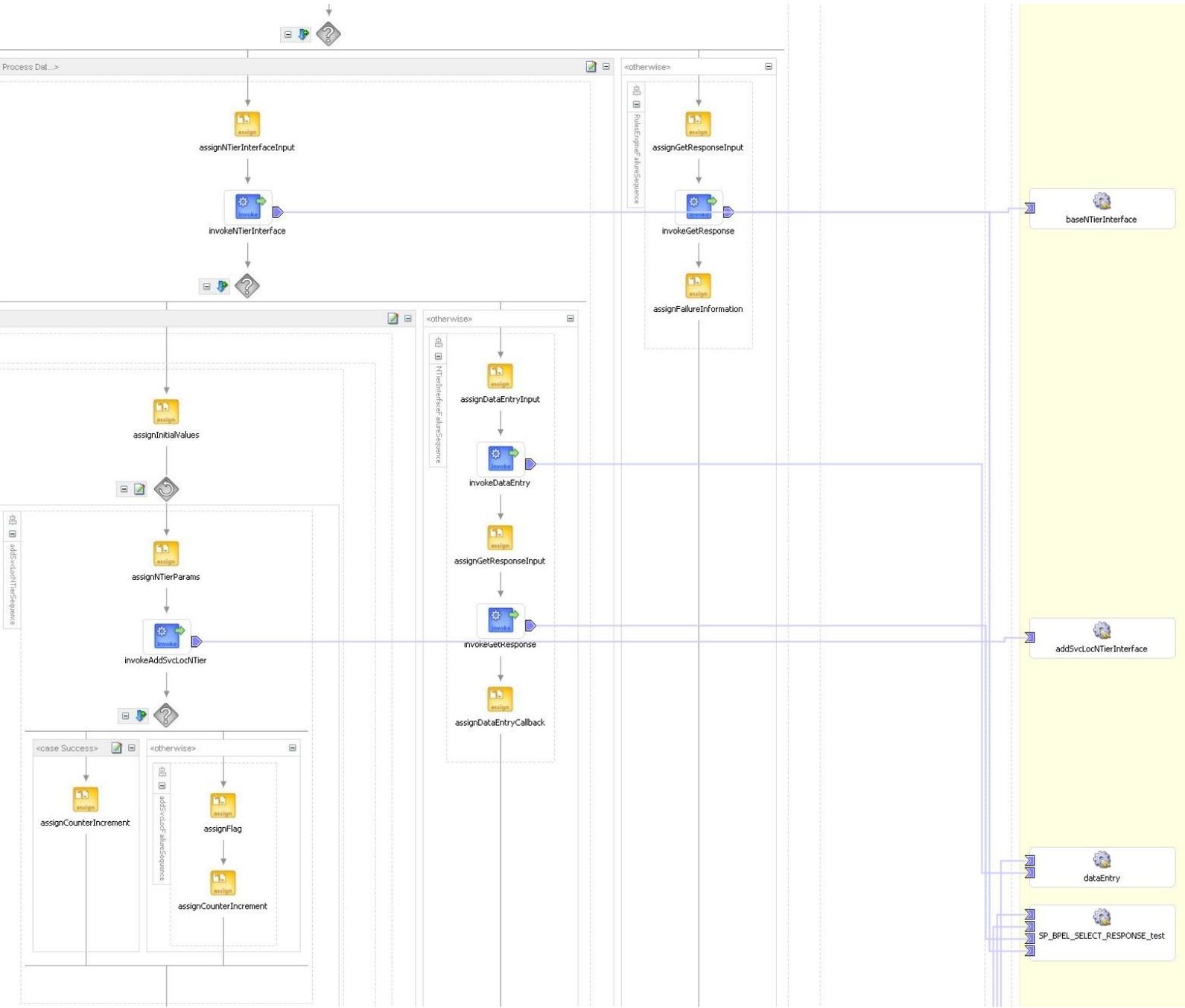
## **5.7 CONSIDERACIONES DE PRUEBA**

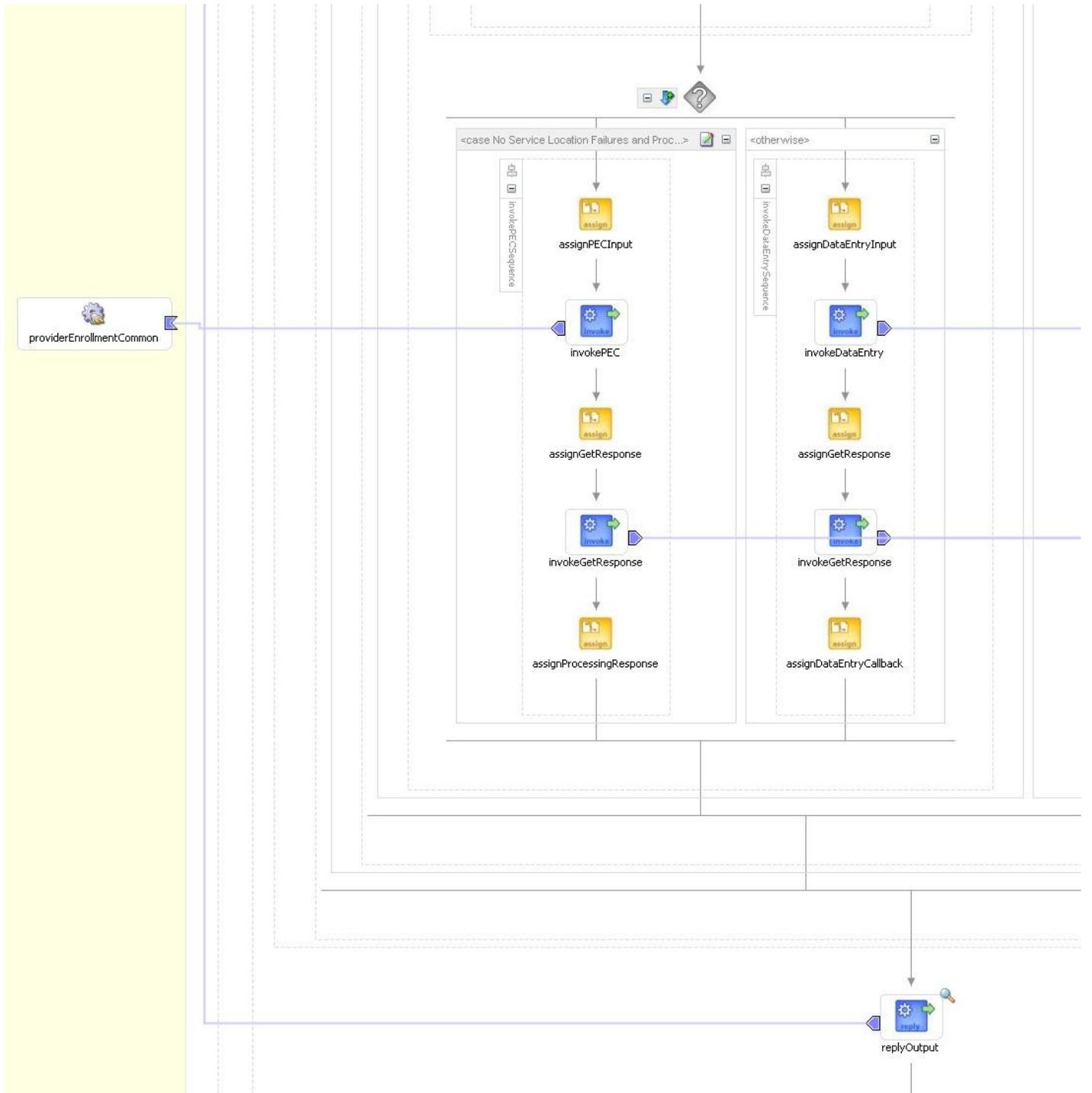
### **5.7.1 Ambiente/Dominio**

La versión más actual de PEM reside actualmente en el dominio ‘test’ del servidor bpeptest.

# Apéndice A: Flujo BPEL ProviderEnrollmentMain

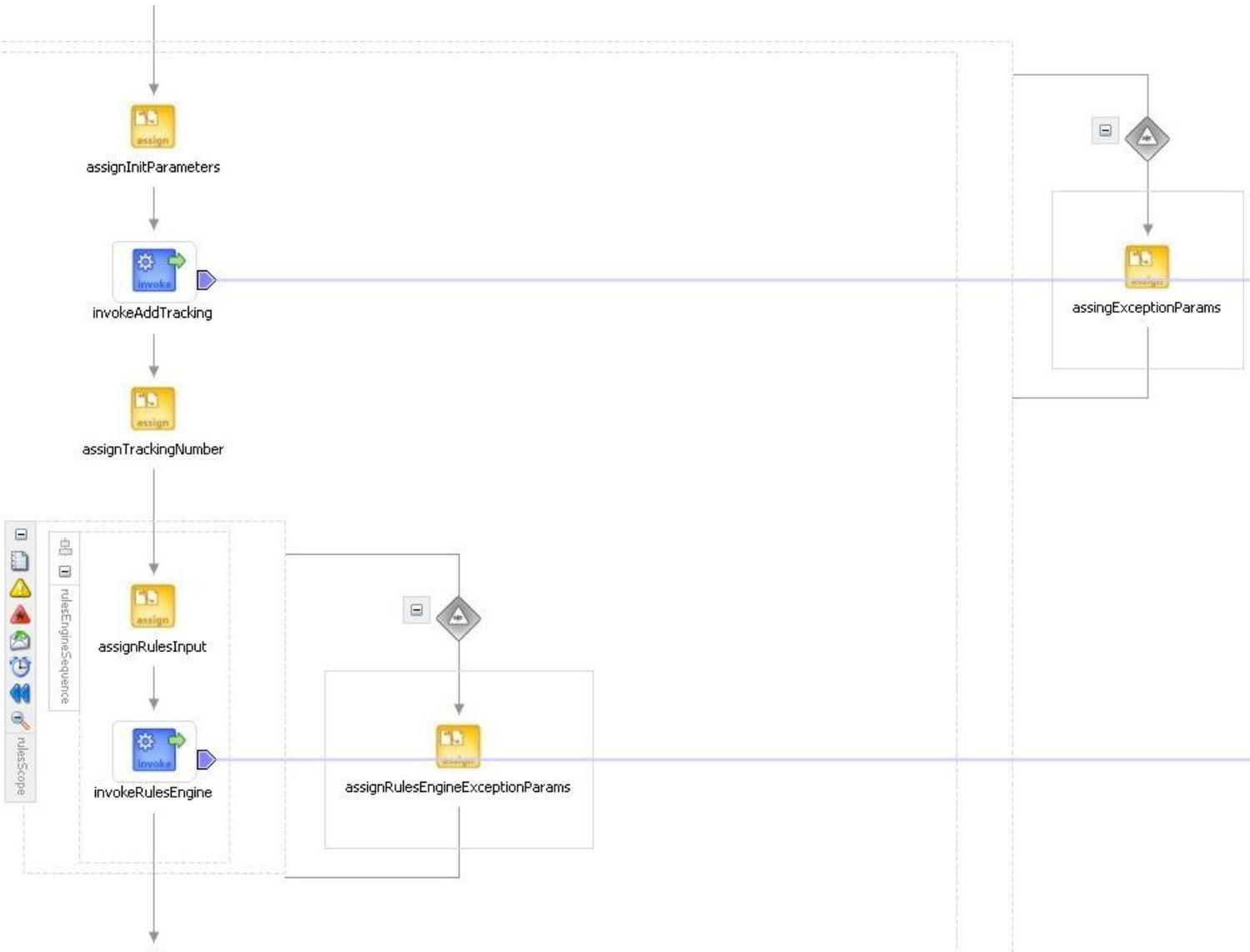






## Apéndice B: Excepciones ProviderEnrollmentMain

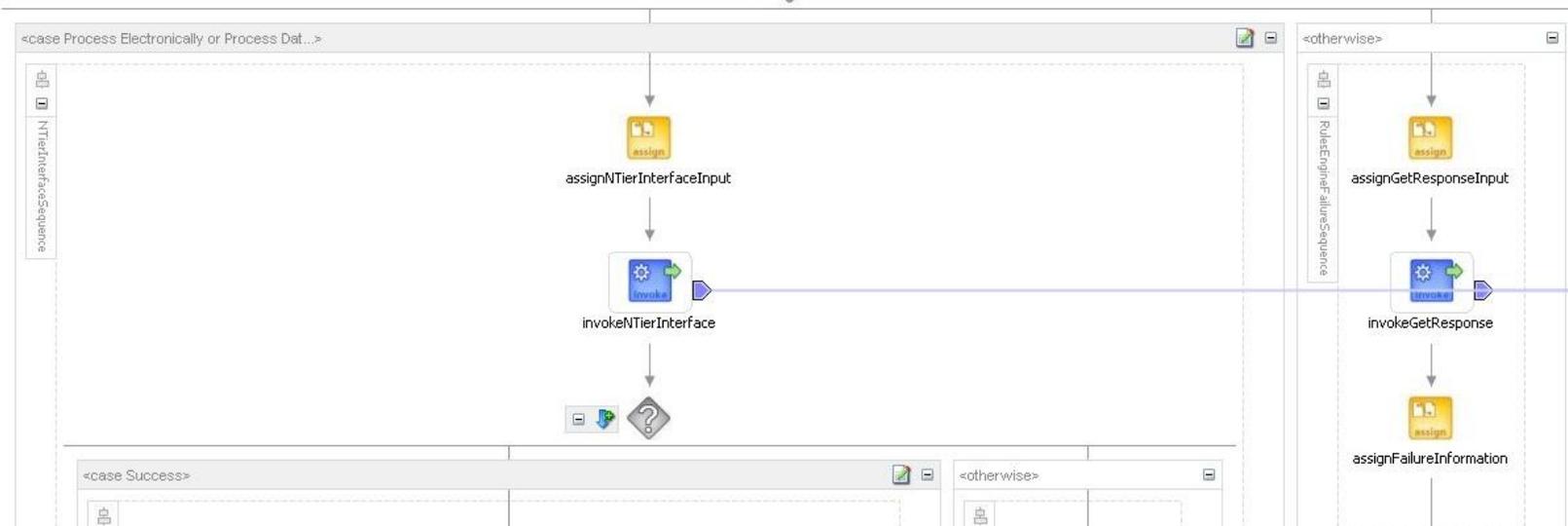
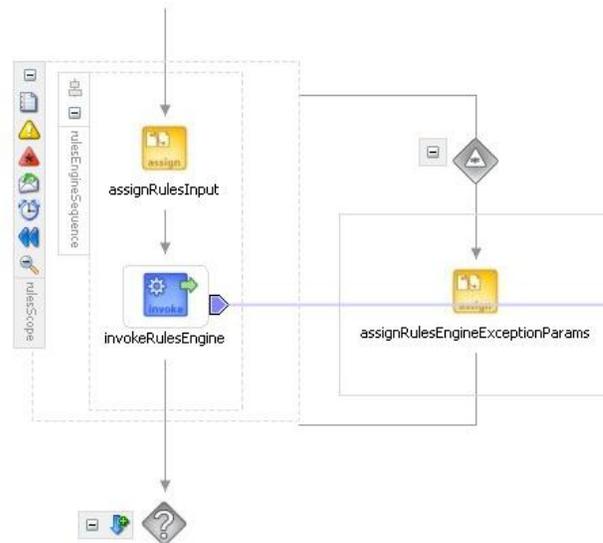
Las siguientes excepciones se basan en darle valor a las variables con información de las excepciones que serán desplegadas una vez que el proceso haya terminado.



### Excepción 1:

Esta rama maneja cualquier excepción general generada en el alcance “Overall”. Esencialmente cualquier excepción en PEM, a menos de que sea manejado por otro manejador de excepciones como los que siguen, será manejado por esta rama.

## Excepción 2:



Esta rama atrapar  cualquier excepci3n generada por el Motor de Reglas y asigna la informaci3n de fallas para desplegar que la invocaci3n al Motor de Reglas ha generado una excepci3n.

## Apéndice C: Detalles de los Procedimientos Almacenados

### Este es el código del procedimiento almacenado SP\_BPEL\_ADD\_TRACKING:

```
CREATE OR REPLACE procedure sp_bpel_add_tracking (provId IN varchar2,
reqType IN varchar2, currentWS IN varchar2, newTrackingNumber OUT
number) as
begin
    insert into t_bpel_tracking (sak_tracking, id_provider, cde_status,
cde_websevice, cde_current_ws, dte_insert, dte_last_updated,
sak_letter)
    values ( seq_bpel_tracking.nextval, provId, 'P', reqType, currentWS,
to_number(to_char(sysdate,'YYYYMMDD')), 0, -1) returning sak_tracking
into newTrackingNumber;
end;
```

### El siguientes es el código del procedimiento almacenado SP\_BPEL\_SELECT\_RESPONSE:

```
CREATE OR REPLACE procedure sp_bpel_select_response( cdeWebService IN
varchar2, cdeWorkflow IN varchar2, cdeResult IN char, cdeActivity IN
varchar2, text OUT varchar2, sakLetter OUT number ) as
begin
    select b.dsc_text,
           a.sak_letter
    into text,
           sakLetter
    from t_bpel_ws_response a,
         t_bpel_rsp_txt b,
         t_bpel_websevice c,
         t_bpel_workflow d
    where
        a.sak_resp_txt = b.sak_resp_txt
    and   a.sak_websevice = c.sak_websevice
    and   a.sak_workflow = d.sak_workflow
    and   c.cde_websevice = cdeWebService
    and   cdeActivity between a.cde_activity_min and
a.cde_activity_max
    and   to_number(to_char(sysdate,'YYYYMMDD')) between a.dte_start
and a.dte_end
    and   a.cde_result = cdeResult
    and   (d.cde_workflow = cdeWorkflow or
           cdeWorkflow = ' ');
end;
```

## **6. Diseño ProviderEnrollmentCommon**

---

### **6.1 DESCRIPCIÓN DEL DISEÑO**

El Proceso BPEL ProviderEnrollmentCommon es responsable de enrolar a un proveedor una vez que ProviderEnrollmentMain o ProviderEnrollmentDataEntry determinan que la aplicación está lista para el enrolamiento final.

#### **6.1.1 Alcance del Diseño**

#### **6.1.2 Elemento de Diseño A**

Este proceso BPEL (ProviderEnrollmentCommon, a partir de ahora “PEC”) es referenciado por los procesos BPEL ProviderEnrollmentMain y ProviderEnrollmentDataEntry. Este proceso recibe, como información de entrada, número de guía, sak (system assigned key) de la aplicación, alguna información de la aplicación (información de notificación como *media* preferida, correo electrónico, número telefónico, etc.) información de recepción de documentación, y fecha. Después de recibir la entrada, verifica para ver si la licencia, forma W9 y la aplicación firmada han sido recibidas. Si no, invoca el proceso BPEL humanWorkflow, solicitando una tarea de licenseReceipt. Si es exitoso, el flujo continua. De otra manera, PEC genera una notificación de falla y el proceso termina. Posteriormente invoca el proceso BPEL licenseValidation para validar la licencia proporcionada por el proveedor. Si este regresa exitosamente, el flujo continua. De otra manera PEC genera una notificación de falla y el proceso termina. Después de esto, PEC invoca el proceso BPEL humanWorkflow para crear una tarea de aprobación de enrolamiento para que alguien pueda aprobar o declinar la aplicación. Si humanWorkflow regresa un aprobación (éxito), PEC va a generar una notificación de éxito y le notifica al proveedor de su aplicación satisfactoria. De otra manera genera una notificación de falla, notifica al proveedor que aplicó de la falla en el enrolamiento y termina el proceso. Ver Apéndice A para una representación gráfica del flujo PEC.

#### **6.2.1 Flujo de Trabajo Humano**

Existen dos llamadas directas a flujos de trabajo humano en PEC. Ambas son procesos BPEL humanWorklow. La primera llama a un flujo de trabajo humano para solicitar la generación de una tarea de recepción de licencia (licenseReceipt) y la segunda llamada a humanWorkflow solicitando la generación de una tarea de aprobación de enrolamiento (enrollmentApproval).

## 6.2.2 Manejo de Tiempo

### **invokeLicenseReceipt**

**Descripción:** Esta es una llamada al proceso BPEL humanWorkflow, solicitando una tarea de recepción de licencia (licenseReceipt), para asegurarse de que la licencia del proveedor ya haya sido recibida para que pueda ser validada y, posteriormente, que la aplicación del proveedor pueda ser aprobada.

**¿Flujo de Trabajo Humano?:** Sí

**Tiempo:** Debido a que esta llamada requiere algún monto de flujos de trabajo humano, el tiempo en esta llamada debe ser tentativamente de 24-48 horas para permitir el elemento de flujo de trabajo humano para esta llamada.

### **invokeNotification**

**Descripción:** Llamada al servicio de notificación (proceso BPEL providerEnrollmentNotification). Hay muchas llamadas a este servicio tanto para los casos de éxito como los casos de error.

**¿Flujo de Trabajo Humano?:** Sí

**Tiempo:** Esta llamada es simplemente una invocación que se hace y se olvida. En cuanto se realiza no tenemos que esperar por la respuesta. Por lo tanto no hay un esquema de tiempo requerido para esta llamada.

### **invLicenseValidation**

**Descripción:** Llamada al proceso de validación de licencia (LicenseValidation) después de que PEC haya determinado que la licencia ha sido recibida satisfactoriamente para ejecutar algunas verificaciones básicas en la información de la licencia para determinar si es válida/legítima.

**¿Flujo de Trabajo Humano?:** Sí

**Tiempo:** El tiempo debe rondar entre los 5-10 segundos.

### **invokeEnrollmentApproval**

**Descripción:** Llamada al proceso humanWorkflow, solicitando una tarea de aprobación de enrolamiento (enrollmentApproval), la cual va a asignar una tarea (flujo de trabajo humano) para que el trabajador humano pueda procesar la aplicación del proveedor.

**¿Flujo de Trabajo Humano?:** Sí

**Tiempo:** Esta llamada requiere un flujo de trabajo humano. Debido a esto el tiempo debe encontrarse entre 24-48 horas para darle un tiempo amplio al humano para obtener esta tarea y aprobarla o denegarla.

## 6.2.3 Manejadores de Compensación

En este momento, no existe persistencia de datos y cualquier falla va a generar simplemente un mensaje de error y posteriormente terminar el flujo.

#### **6.2.4 Interfaces NTier**

Ninguna llamada a NTierInterface está presente en PEC.

#### **6.2.5 Sensores**

No hay sensores actualmente presentes en PEC.

#### **6.2.6 Alcance del Proceso BPEL**

La discusión del alcance del flujo de trabajo se ingresa aquí como vaya siendo necesario. La adherencia con las prácticas aceptadas BPEL no requiere mayor discusión.

#### **6.2.7 Motor de Reglas**

No hay llamadas al motor de reglas en PEC.

### **6.3 XSD**

PEC utiliza los siguientes XSDs para funcionar correctamente:

- **providerEnrollment.xsd** - Este XSD define las variables utilizadas por PEC.
- **interChangeCommon.xsd** – Este XSD define muchos tipos para las variables de información de proveedor de PEC.
- **interChangeProvider.xsd** – Este XSD define las restricciones para los tipos utilizados para la información de proveedores.
- **interChangeRecipient.xsd**

### **6.4 Namespace**

<http://www.eds.com/ns/iC.MMIS.WS.Prov.Enrollment>

### **6.5 Localidad de los Esquemas**

interChangeCommon.xsd

providerEnrollment.xsd

interChangeProvider.xsd

### **6.6 WSDL**

PEC no requiere WSDLs adicionales para funcionar además de su propio WSDL.

### **6.7 SOBRECARGA DE RENDIMIENTO/SISTEMA**

Esto no ha sido determinado. Pruebas posteriores nos dictarán los requerimientos para esta sección.

### **6.8 DEPENDENCIAS**

PEC depende de los siguientes *web services* para funciona adecuadamente: humanWorkflow y licenseValidation. Además utiliza el *web service* Notification pero no es componente crítico para el proceso ya que las llamadas a éste son únicamente con propósitos de notificación y no afectan de manera adversa la

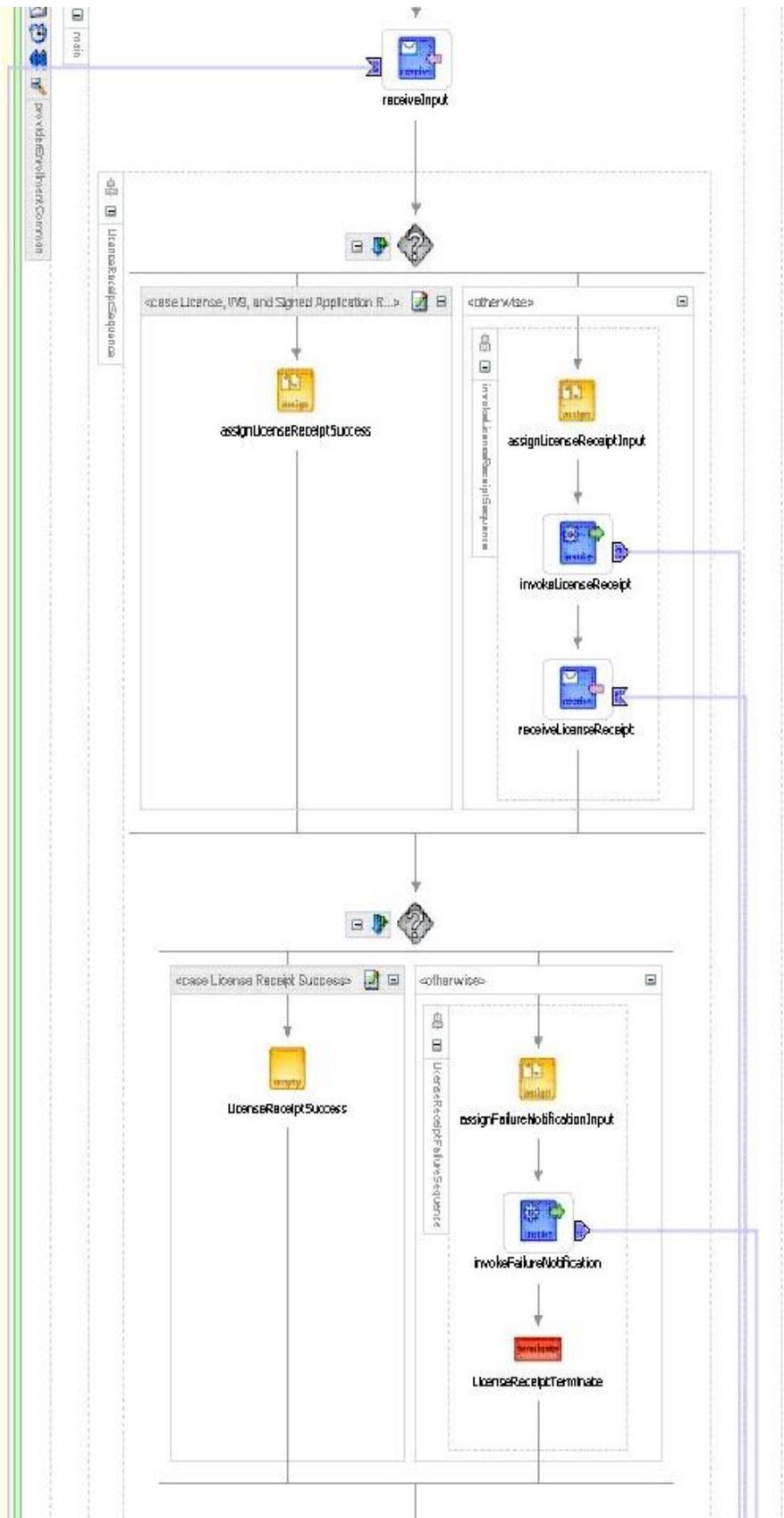
ejecución del proceso en caso de que este llegue a fallar.

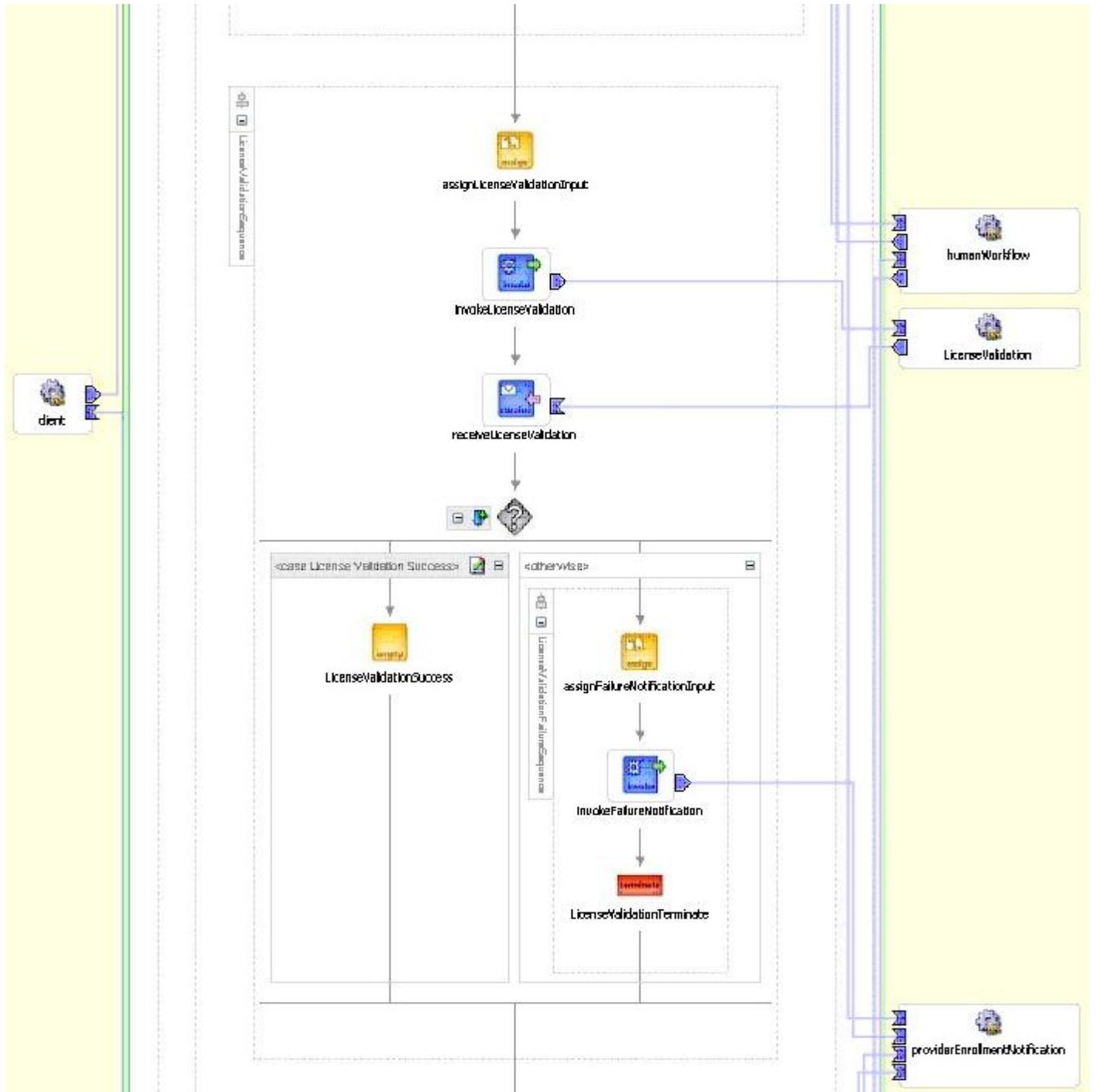
## **6.9 CONSIDERACIONES DE PRUEBA**

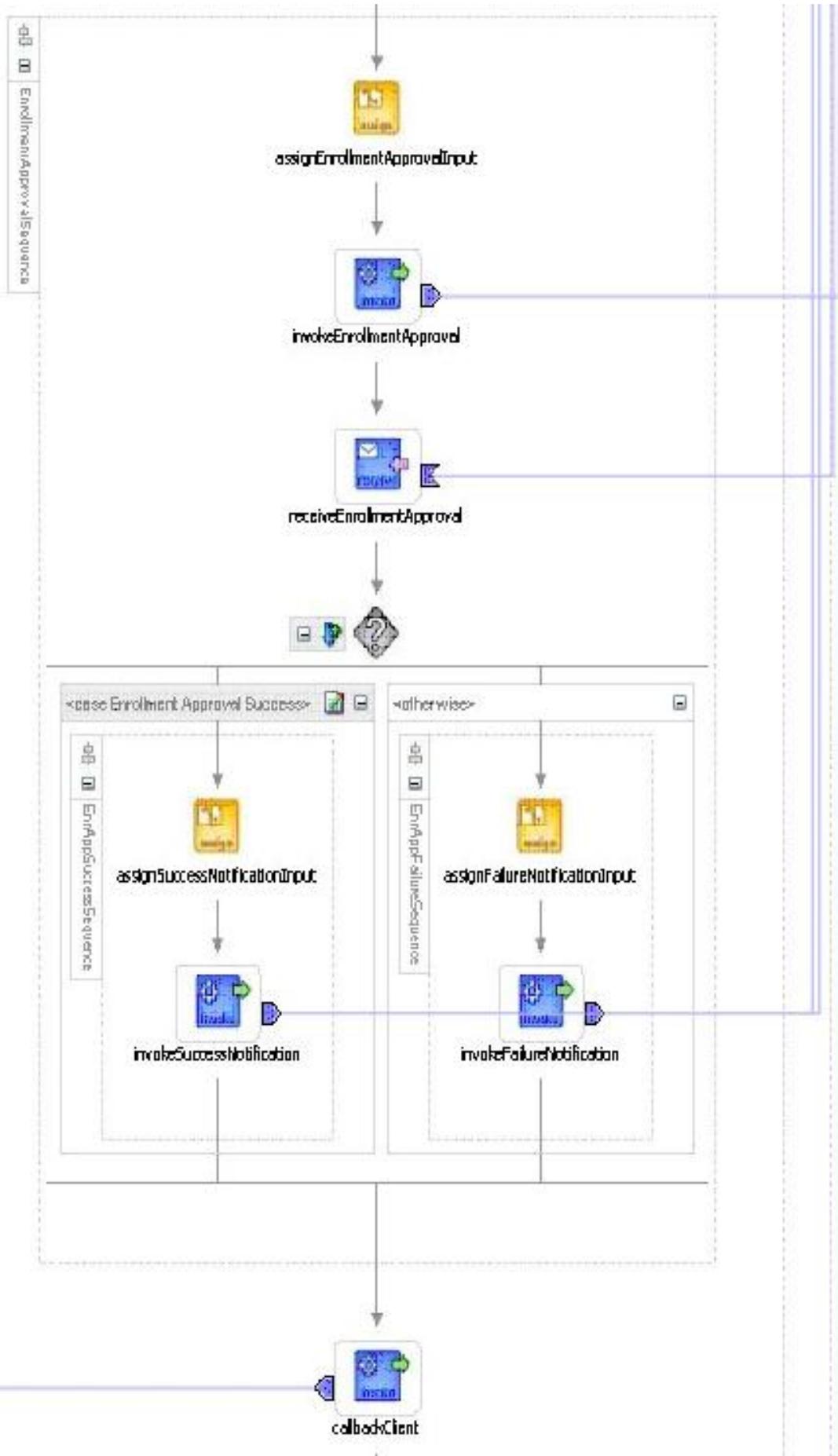
### **6.9.1 Ambiente/Dominio**

La última versión de PEC se encuentra actualmente en el dominio 'test' en bpeptest.

# Apéndice A: Flujo BPEL ProviderEnrollmentCommon







## **7. Diseño Web Service Verificar Estatus**

---

### **7.1 DESCRIPCIÓN DEL DISEÑO**

Un *web service* síncrono que permite a los proveedores de servicios médicos verificar el estatus de su aplicación como proveedor de Medicaid. El usuario proporciona un número de guía (que se les proporciona cuando enviaron una petición de enrolamiento), posteriormente el *web service* busca la entrada correspondiente y devuelve el estatus de la aplicación. Si el número de guía no es válido (es decir, no fue encontrado), o si hay problemas de autenticación, el *web service* devuelve un mensaje de error. Una representación gráfica del flujo de datos se puede ver en el Apéndice A. La implementación BPEL de este servicio complementa otra, más ineficiente y costosa, de sistemas antiguos donde los proveedores recibían la notificación del estatus de su aplicación por correo o por teléfono.

#### **7.1.1 Alcance del Diseño**

#### **7.1.2 Web Service Verificar Estatus**

El servicio que va a llevar a cabo el proceso de negocio de verificación de estatus, el cual permite al proveedor conocer el estatus de su aplicación proporcionando el número de guía. El proveedor visita una forma *web* en el Internet la cual inicialmente solicita alguna forma de identificación (a través de un servicio de identidad) y un número de guía. Después de que ocurre la autenticación, se le informa al proveedor del estatus de su aplicación, asumiendo que proporcionó un número de guía válido. Otra información de contacto puede ser también desplegada para que el proveedor pueda hablar con un representante acerca de algún problema que se le pudiera presentar.

#### **7.1.3 Flujo de Trabajo Humano**

No se requieren elements de flujo de trabajo humano; el proceso es completamente síncrono.

#### **7.1.4 Manejo de Excepciones**

La implementación BPEL de este *web service* debe considerar excepciones que involucren ingreso de números de guía erróneos e inhabilidad para comunicarse correctamente con la base de datos.

#### **7.1.5 Diseño Específico**

El service utiliza el campo de estatus de la tabla *bpel CUBE\_INSTANCE* para determinar en que fase se encuentra el proceso correspondiente de la aplicación (esto es Aplicación Denegada, Etapa de Verificación de Información, etc.). El campo de estatus en esta tabla simplemente almacena el alcance de la actividad actual dentro de esta instancia (identificado por el *conversionId* / número de guía) del proceso *bpel*. Esta información necesita después ser traducida a un enunciado acerca del proceso de la aplicación que el usuario pueda entender. La parte difícil de este módulo es obtener el estatus del proceso *bpel* correcto *ProviderEnrollmentMain*. *ProviderEnrollmentMain* debe proporcionar el *conversationId* de la instancia del *web service*, en cuyo caso el *conversationId* equivale al número de guía,<sup>48</sup>

necesita haber un mapeo local (una tabla) de los conversationIds actuales de las instancias de procesos a los números de guía proporcionados por el usuario. Así, el servicio de verificación de estatus consulta al servidor orabpel para determinar el alcance de los procesos identificados por el conversationId y obtener el alcance la instancia que se está ejecutando.

## **7.2 XSD**

Se requiere un XSD simple con campos para el número de guía y el estatus de la aplicación. (checkStatus.xsd)

## **7.3 WSDL**

Se requiere un WSDL que defina el adaptador de la base de datos para consultar el servidor orabpel acerca del estatus de la instancia ProviderEnrollmentMain identificada por el número de guía.

## **7.4 TABLAS DE DATOS**

El *web service* va a acceder la tabla de orabpel CUBE\_INSTANCE para obtener el alcance de la actividad actual dentro de la instancia de la invocación al *web service*. El nombre de alcance se traduce posteriormente en un enunciado informativo acerca del estatus de la aplicación para que el usuario lo pueda leer. Adicionalmente, otra tabla (tal vez inclusive PNR\_ENROLLMENT\_TRACKING) puede ser necesario para hacer un mapeo de las conversationIds específicas de la instancia con los números de guía proporcionados por el usuario. Esta tabla tendría que ser creada por el proceso ProviderEnrollmentMain y accesada por el servicio de verificación de estatus.

## **7.5 SOBRECARGA DE RENDIMIENTO/SISTEMA**

En este momento se desconoce la habilidad del servidor para mandar un alto volumen de peticiones de clientes concurrentes. Se requieren pruebas de estrés para determinar el rendimiento del sistema.

## **7.6 DEPENDENCIAS**

### **7.6.1 Web Services BPEL auxiliares**

Este modulo requiere la conducta correcta de los demás *web services* para una compleción correcta.

### **7.6.2 Servicio de Identidad**

Autentifica e identifica al cliente que envía la petición. El *web service* de verificación de estatus depende de este módulo para la seguridad y privacidad de la información del proveedor. El mayor problema de seguridad en este proceso es la identificación y autenticación del proveedor. Para cumplir con HIPPA, nadie debe poder ver el estatus de una aplicación más que el proveedor que envió las formas para convertirse en un proveedor de Medicaid. Idealmente, un servicio de identidad se utilizaría para asegurarse de que la información del proveedor es segura y que solo usuarios autorizados pueden checar su propio estatus.

### **7.6.3 Enrolamiento de Proveedores**

Permite a los proveedores enrolarse como proveedores de Medicaid y les proporciona un número de guía para su aplicación. El *web service* de verificación de estatus depende en el servicio de enrolamiento de proveedores para agregar satisfactoriamente al proveedor en la base de datos de seguimiento de enrolamiento para proporcionar el número de guía correcto al usuario al completar el servicio. Para más información se puede revisar el documento de diseño de Enrolamiento de Proveedores.

## **7.7 CONSIDERACIONES DE PRUEBA**

El sistema debe ser probado con casos de prueba unitarios que exploten la conducta del sistema cuando se manejan números de guía válidos e inválidos. Fuera de estos dos dominios, puede ser necesario hacer pruebas de estrés en el servidor, para determinar el máximo número de peticiones que el servidor puede manejar al mismo tiempo y poder proporcionar resultados precisos.

### **7.7.1 Herramientas**

Las capacidades de prueba en Oracle BPEL deben ser suficientes para probar el *web service* de verificación de estatus.

### **7.7.2 Ambiente/Dominio**

Las localidades de los *partner links* del servicio de Identidad y del acceso a Bases de Datos deben ser conocidas.

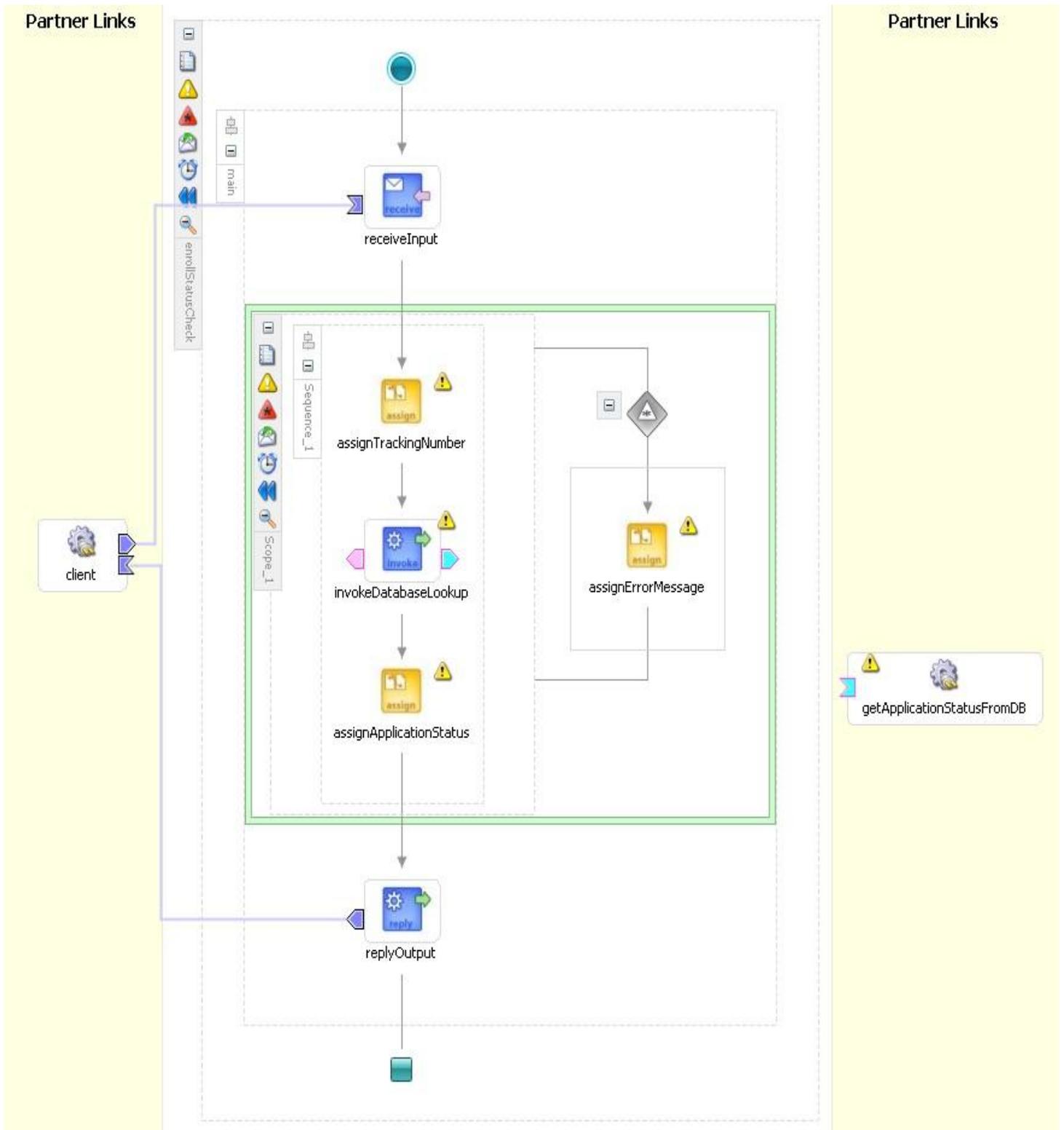
## **7.8 ENTREGABLES**

*checkStatus.xsd*

*enrollStatusCheck.bpel*

*dbGetApplicationStatus.wsdl*

# Apéndice A: Flujo BPEL Verificación de Estatus



## **8. Diseño Agregar Localidad de Servicio**

---

### **8.1 CONVENCIONES DEL DOCUMENTO**

Todo el texto en *itálicas* se refiere al nombre descriptivo, de acuerdo a las prácticas aceptadas BPEL, de un proceso BPEL asociado con el segmento de Enrolamiento de Proveedores del sistema interChange Enhancements.

### **8.2 DESCRIPCIÓN DEL DISEÑO**

Este *web service* agrega una localidad de servicio a un proveedor, incorpora el tipo de proveedor y la especialidad, y asocia el proveedor a los programas de proveedores basados en una base de derechohabientes. Específicamente, una localidad de servicio se refiere a un conjunto de información acerca de un proveedor, como el tipo y su especialidad. Como ejemplo podemos tener a un proveedor que sea un médico general en una oficina. Sin embargo, ese mismo doctor puede ser un cirujano pediátrico en un hospital. De manera que hay dos localidades de servicio diferentes para el mismo médico debido a que proporciona dos diferentes servicios en dos localidades diferentes.

### **8.3 Alcance del Diseño**

El propósito de este diseño es satisfacer los requerimientos asociados con el *web service addServiceLocation*.

### **8.4 Web Service Add Service Location**

El *web service addServiceLocation* involucra invocar el *web service* como una operación independiente para darle soporte a un proveedor ya existente en el sistema. Desde este punto el proceso comienza, de manera que una localidad de servicio pueda ser agregada a una base existente de proveedor ya sea por el proveedor mismo (al actualizar su perfil) o internamente por un empleado de correo.

#### **8.4.1 Web Service AddSvcLocCommon**

El *web service AddSvcLocCommon* se utiliza para segregar las llamadas asíncronas a *web services*. De esta manera *addServiceLocation* puede mandar una respuesta administrable inmediata al usuario sin tener que esperar por la respuesta a la llamada asíncrona desde los *web services* invocados como *humanWorkflow* y *notification*. *AddSvcLocCommon* engloba la entrada de datos, la aplicación de enrolamiento y llamadas a *web services* de notificación.

#### **8.4.2 Web Service addSvcLocEnrollApp**

El *web service AddSvcLocEnrollApp* es un servicio auxiliar para mantener la estructura como está definida en la sección 3.2.1. Este servicio engloba las llamadas a la aplicación de enrolamiento y a notificación.

#### **8.4.3 Flujo de Trabajo Humano**

No hay tareas de flujo de trabajo humano asociadas con `addServiceLocation` debido a que todas las tareas de flujo de trabajo humano están diversificadas en los siguientes *web services* que son invocados por `addServiceLocation`. Estos *web services* son `addSvcLocCommon` y `addSvcLocEnrollApp`.

#### **8.4.4 Interfaz Gráfica de usuario Add Service Location**

Para el *web service* `add service location`, la interfaz gráfica de usuario debe contener todos los campos necesarios asociados con la capacidad de agregar una localidad de servicio. La interfaz incluye también campos previamente poblados por el sistema con información base de los proveedores. Estas son pantallas de .NET ensambladas de acuerdo a los XSDs definidos. La interfaz de usuario solo es visible al agregar una localidad de servicio para un proveedor ya existente.

#### **8.4.5 Fallas en la Localidad de Servicio**

En el evento de que al intentar agregar una localidad de servicio se presente un error, se vuelve imperativo que cada localidad de servicio fallida marcada como una localidad de servicio fallida no sea persistida en la base de datos. Estas localidades de servicio marcadas deben ser almacenadas y enviadas a una lista de trabajo para que el empleado de correo actúe sobre ellas.

#### **8.4.6 Manejo de Excepciones**

El diseño del flujo de trabajo debe considerar fallas al leer o escribir en las tablas de la base de datos asociadas con la información base del proveedor y en concordancia con los parámetros de tiempo del flujo de trabajo. Cuando un error así se presenta, se debe devolver un parámetro de falla al flujo de trabajo y los eventos/mensajes subsecuentes deben ser devueltos al proceso que invocó.

#### **8.4.7 Manejo de Tiempo**

Cada llamada debe considerar el tiempo de procesamiento y definir parámetros de tiempo apropiados para asegurarse de que el flujo de trabajo se ejecute hasta su compleción. Para `addServiceLocation`, será necesario que haya parámetros de tiempo para asegurarnos de que si el usuario no ingresa la información dentro de un tiempo establecido, el flujo de proceso BPEL `addServiceLocation` va a continuar ejecutándose hasta su compleción, no afectando de esta manera a los procesos que lo invocan. Esto por supuesto solo aplica para instancias donde el flujo de trabajo humano esté involucrado. Todos los flujos de trabajo humano se manejan con un tiempo de 10 segundos.

#### **8.4.8 Manejadores de Compensación**

No hay manejadores de compensación asociados con `addServiceLocation`.

#### **8.4.9 Interfaces NTier**

Este servicio es una interfaz .NET que invoca la funcionalidad de la capa de negocios al agregar a un proveedor en el sistema. En relación con el *web service* `addServiceLocation`, es necesario almacenar todos los datos de las localidades de servicio en la base de datos a través de la capa de negocios NTier.

#### **8.4.10 Sensores**

El *web service* `addServiceLocation` requiere de monitoreo en la entrada y salida del proceso.

#### **8.4.11 Alcance del Proceso BPEL**

La discusión del alcance del flujo de trabajo se ingresa aquí como va siendo necesario. La adherencia a las prácticas aceptadas BPEL no requiere mayor discusión.

#### **8.4.12 Motor de Reglas**

Con respecto al motor de reglas, solo cuando se trata de un proveedor existente será necesario hacer referencia al motor de reglas para validar los datos de la localidad de servicio.

#### **8.4.13 Entradas del Proceso**

El Proceso AddServiceLocation obtiene datos ingresados por el proveedor a través de una forma. En el Apéndice D se puede encontrar una tabla con las variables específicas y su tipo de dato. Para más detalle y la construcción de las variables en relación al flujo BPEL, ver Sección 4 y el Apéndice A. A continuación se muestra una descripción de las variables.

idType – De tipo cadena (string) y puede tomar los siguientes valores: “tax”, “medicare”, “dea”, “nabp”, y “npi”. Esta variable se utiliza para denotar el tipo de id relacionado con el proveedor en cuestión.

id – De tipo “idDef” que es de tipo cadena (string) con una longitud máxima de 13 caracteres.

svcLoc – svcLoc se utiliza como un nombre descriptivo para la localidad de servicio

county – Condado en donde se localiza esta localidad de servicio

city - Ciudad en la cual se localiza esta localidad de servicio

street - Calle en donde se localiza esta localidad de servicio

zipbase – Primeros cinco dígitos del código postal en donde se localiza esta localidad de servicio

zipExtention – Sigüientes cuatro dígitos del código postal en donde se localiza esta localidad de servicio

areaCode – Código de área en donde se localiza esta localidad de servicio

digits – Primeros tres dígitos del número telefónico en donde se localiza esta localidad de servicio

exchange – Últimos 4 dígitos del número de teléfono en donde se localiza esta localidad de servicio

extension – Número de extensión en donde se localiza esta localidad de servicio

provType – El tipo de proveedor

provSpec – La especialidad del proveedor

targetRecipientProgram – Programa de Medicaid a donde está destinado este proveedor

language - Idioma que el proveedor habla

languageStartDate – Fecha en que el proveedor comenzó a hablar este idioma

languageStopDate – Fecha en que el proveedor dejó de hablar este idioma

billingIndicator -

handicapAccess – Valor verdadero/falso utilizado para determinar si la localidad de servicio tiene accesos para minusválidos

latitude – Latitud de las coordenadas para esta localidad de servicio

longitude – Longitud de las coordenadas para esta localidad de servicio

orgCode -

pubPvtInd -

WithHoldFICA -

HIPAA837Capable -

## 8.5 XSD

El *web service* addServiceLocation es interfaz de muchos otros *web services*. Debido a esto es necesario proporcionar diferentes construcciones de petición/respuesta para cada *web service*. Una descripción de cada uno de los tipos de entrada se proporciona en el Apéndice B con una descripción de su propósito en la sección 3.2.11. Las representaciones actuales de código se proporcionan en el Apéndice A.

## 8.6 Namespace

El *Namespace* para este proceso BPEL es el siguiente:

- <http://www.eds.com/ns/iC.MMIS.WS.Prov.Enrollment>

## 8.7 Localidad del Esquema

Todos los esquemas se encuentran de manera local

## 8.8 WSDL

No se requiere otro WSDL además de addServiceLocation.wsdl

## 8.9 TABLAS DE DATOS

Debido a que un flujo de trabajo humano es utilizado en uno de los escenarios del *web service* addServiceLocation, será necesario acceder información del analista desde el servidor LDAP para proporcionar acciones en tareas para los empleados de correo y las notificaciones para los proveedores.

### 8.9.1 BPEL

No hay tablas de datos asociadas con el *tablespace* bpel

### 8.9.2 interChange

Las tablas de datos dentro del estándar de interChange incluyen aquellas asociadas con el servidor LDAP utilizadas para addServiceLocation.

## 8.10 SOBRECARGA DE RENDIMIENTO/SISTEMA

El rendimiento depende del usuario para que este ingrese la información de una manera expedita para continuar con el flujo de proceso BPEL.

### 8.11 DEPENDENCIAS

Las dependencias asociadas con la compleción de addServiceLocation incluyen, compleción del servicio de notificación, motor de reglas, aprobación de enrolamiento, interfaces NTier y dataEntryOCR.

### 8.12 ENTREGABLES

El *web service* addServiceLocation depende de dependencias externas como el motor de reglas, el servicio de notificación, configuración LDAP, y el desarrollo de las clases de iCE.

### 8.13 TABLA DE MAPEO DE REQUERIMIENTOS

Esta tabla captura todos los requerimientos asociados con el diseño y hace referencia a la subsección que cubre estos requerimientos en la sección 2 de la documentación de Análisis de Enrolamiento de Proveedores. Los documentos de diseño y pruebas pueden mantener asociación con estos elementos.

<b>ID Requerimiento</b>	<b>Descripción</b>	<b>Elemento de Diseño</b>
N/A	<i>Web Service</i> Add Service Location	11.8
N/A	GUI Web Add Service Location	11.4
N/A	<i>Web Service</i> Notificación	11.3
N/A	API de la lista de trabajo	11.5
N/A	<i>Web service</i> interno enrollB	11.7
N/A	Suscripción a las notificaciones de la lista de trabajo	11.11
N/A	GUI .NetNuke de suscripción a la lista de trabajo	11.12
N/A	GUI .NetNuke de la lista de trabajo	11.12

## Apéndice A: Construcción del mensaje Add Service Location

Esta es la construcción xml del mensaje addServiceLocationRequest y el mensaje addServiceLocationResponse.

### Add Service Location Request

```
<element name="addServiceLocationProcessRequest">
  <complexType>
    <sequence>
      <element name="idType" type="string"/>
      <element name="id" type="string"/>
      <element name="svcLoc" type="string"/>
      <element name="county" type="string"/>
      <element name="address" type="string"/>
      <complexType>
        <sequence>
          <element name="street" type="string"/>
          <element name="city" type="string"/>
          <element name="state" type="string"/>
          <element name="zipcode" type="string"/>
          <complexType>
            <sequence>
              <element name="ZipBase" type="string"/>
              <element name="ZipExtension" type="string"/>
            </sequence>
          </complexType>
        </sequence>
      </complexType>
    </sequence>
  </complexType>
  <element name="serviceLocationPhone" type="string"/>
  <complexType>
    <sequence>
      <element name="areaCode" type="string"/>
      <element name="exchange" type="string"/>
      <element name="digits" type="string"/>
    </sequence>
  </complexType>
</element>
```

```

        <element name="extension" type="string"/>
    </sequence>
</complexType>
<element name="provType" type="string"/>
<element name="provSpec" type="string"/>
<element name="targetRecipientProgram" type="string"/>
<element name="languageSpoken" type="string"/>
    <complexType>
        <sequence>
            <element name="language" type="string"/>
                <element name="languageStartDate" type="string"/>
                <element name="languageStopDate" type="string"/>
            </sequence>
        </complexType>
<element name="billingIndicator" type="string"/>
<element name="handicapAccess" type="string"/>
<element name="latitude" type="string"/>
<element name="longitude" type="string"/>
<element name="orgCode" type="string"/>
<element name="pubPvtInd" type="string"/>
<element name="withholdFICA" type="string"/>
<element name="HIPAA837Capable" type="string"/>
</sequence>
</complexType>
</element>

```

### **Add Service Location Response**

```

<xsd:element name="addServiceLocationResponse"
    type="iC.MMIS.WS.Prov.Enrollment:addServiceLocationResponseType"/>

```

```

<xsd:complexType name="addServiceLocationResponseType">
    <xsd:sequence>

```

```

    <xsd:element name="requestStatus"
      type="iC.MMIS.WS.Common:requestStatusType"/>
    <xsd:element name="stateContact"
      type="iC.MMIS.WS.Common:contactType"/>
  </xsd:sequence>
</xsd:complexType>

```

### Add Service Location Common Request

```

<element name="addSvcLocCommonRequest"
  type="iC.MMIS.WS.Prov.Enrollment:addSvcLocCommonRequestType"
  "/>

```

```

<complexType name="addSvcLocCommonRequestType">
  <sequence>
    <element name="addServiceLocationRequest"
      type="iC.MMIS.WS.Prov.Enrollment:addServiceLocationRequestType"
      "/>
    <element name="trackingNumber"
      type="iC.MMIS.WS.Common:trackingNumberType"/>
  </sequence>
</complexType>

```

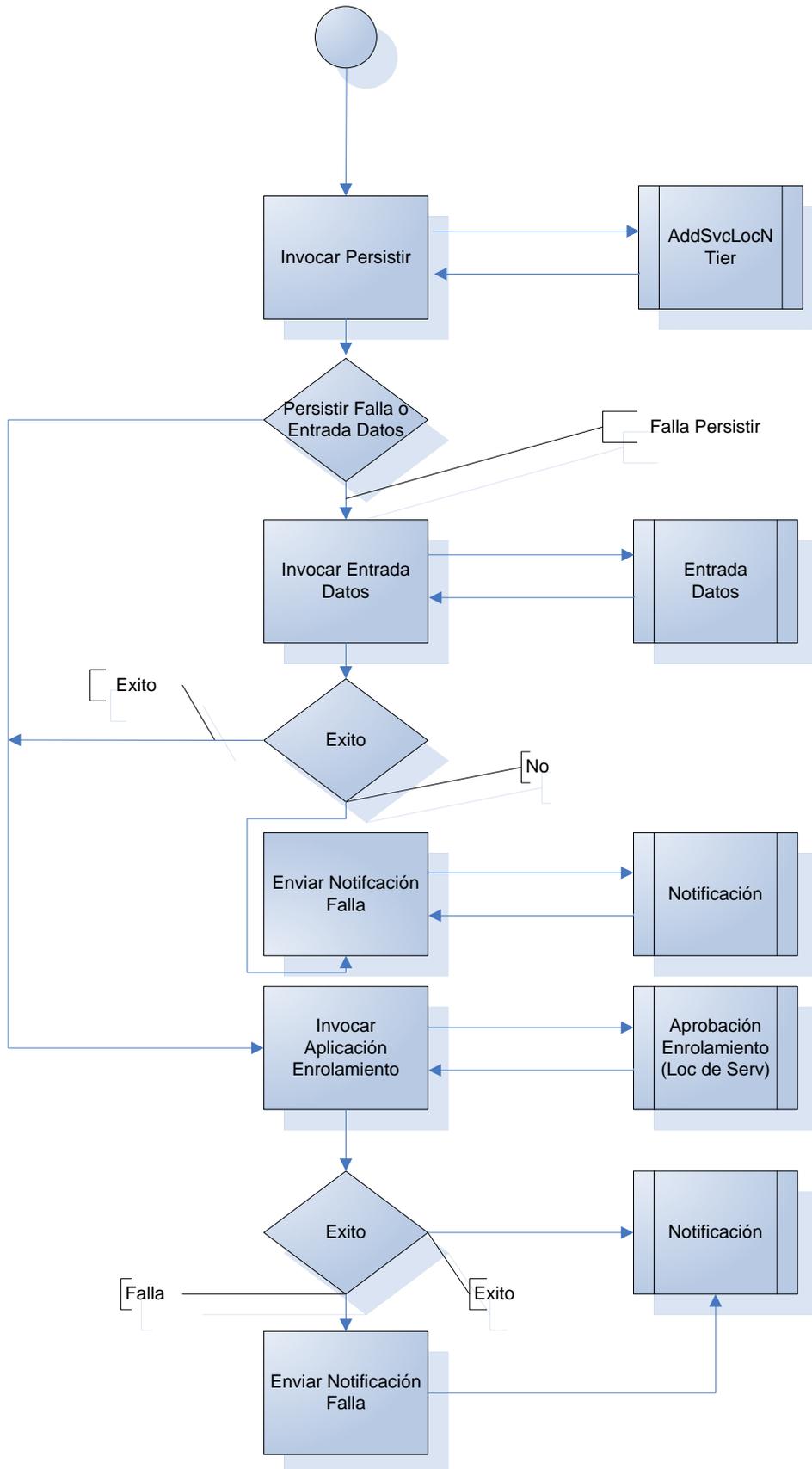
\* Nótese que la construcción del mensaje addSvcLocEnrollApp es la misma que la construcción de addSvcLocCommon

### Apéndice B: Mapeo de entradas/variables de la forma Add Service Location

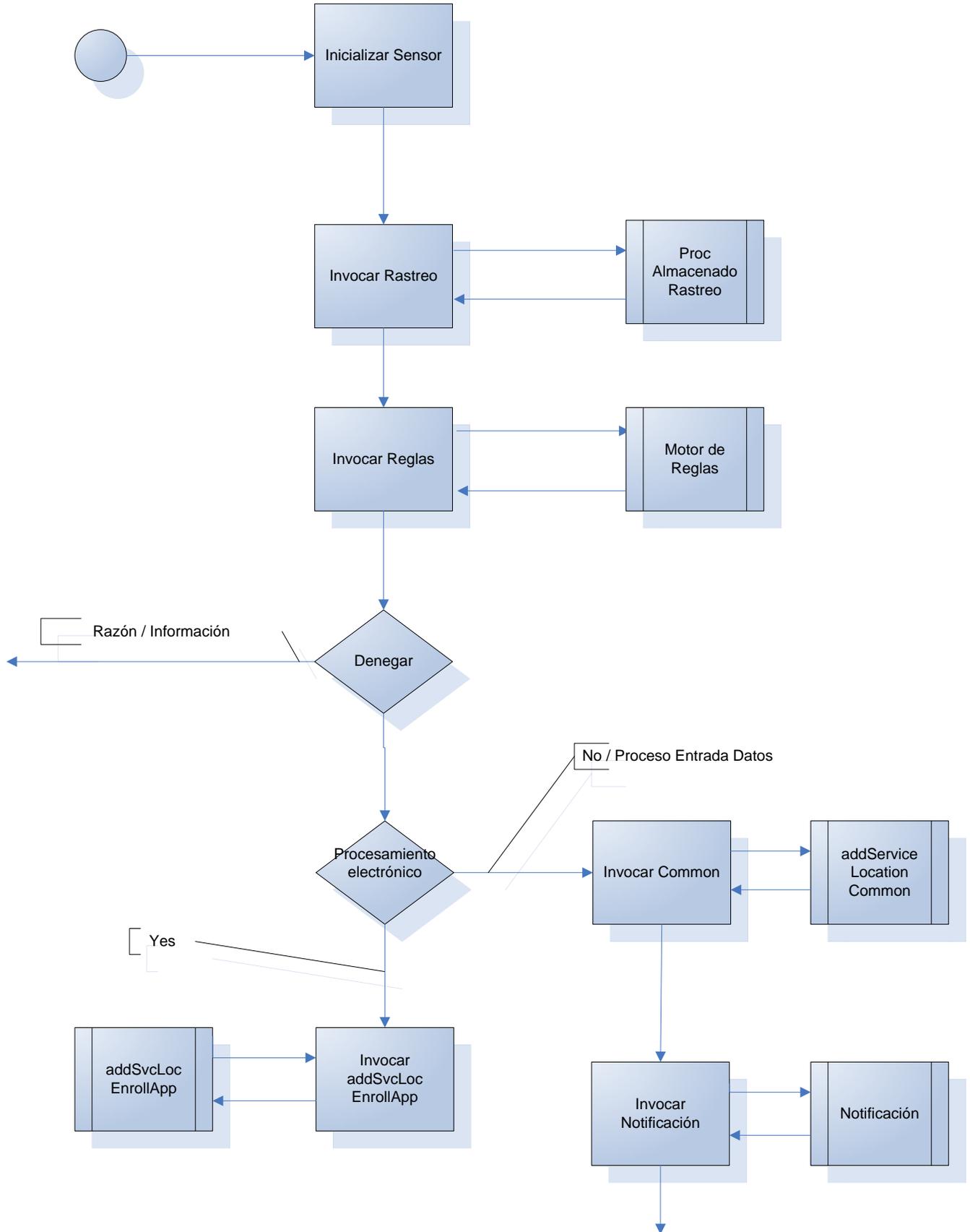
Variable	Tipo de Dato	Etiqueta	Valor por omisión
idType	String	Provider ID Type	Blank
Id	Numeric	Provider ID	Blank
svcLoc	String	Service Location	Blank(drop down)
County	String	County	Blank

City	String	City	Blank
street	String	Street	Blank
state	String	State	Blank
zipbase	String	Zip	Blank
zipExtention	String	Zip (concatenated)	Blank
areaCode	String	Phone(concatenated )	Blank
digits	String	Phone(concatenated )	Blank
exchange	String	Phone(concatenated )	Blank
extension	String	Extension	Blank
provType	String	Provider Type	Blank
provSpec	String	Specialties	Blank(Drop Down)
targetRecipientProgram	N/A	N/A	N/A
Language	String	Language	Blank(drop down)
languageStartDate	Date	Effective Date	Blank
languageStopDate	Date	End Date	Blank
billingIndicator	Boolean	Billing Indicator	Yes
handicapAccess	Boolean	Handicapped Accessible?	Blank(drop down)
Latitude	latitudeType (String)	Latitude	TBD
Longitude	longitudeType (String)	Longitude	TBD
orgCode	Boolean	Organizational Code	
pubPvtInd	String	Public/Private Indicator	Private
WithHoldFICA	Boolean	Withhold FICA	No
HIPAA837Capable	Date	837 Cert. Date	Blank

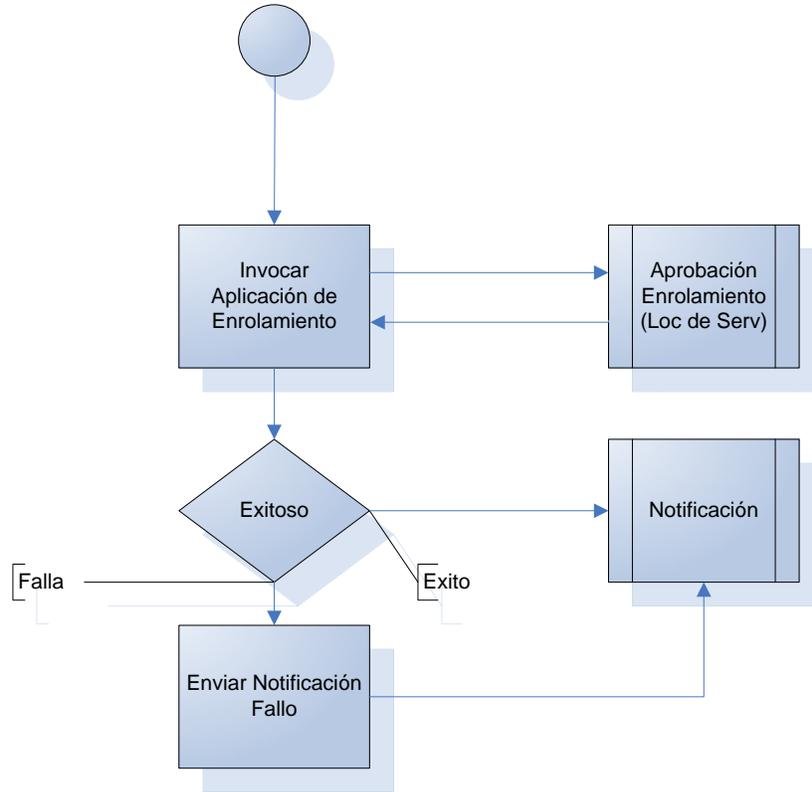
## Apéndice C: Diagrama de flujo de datos Add Service Location Common



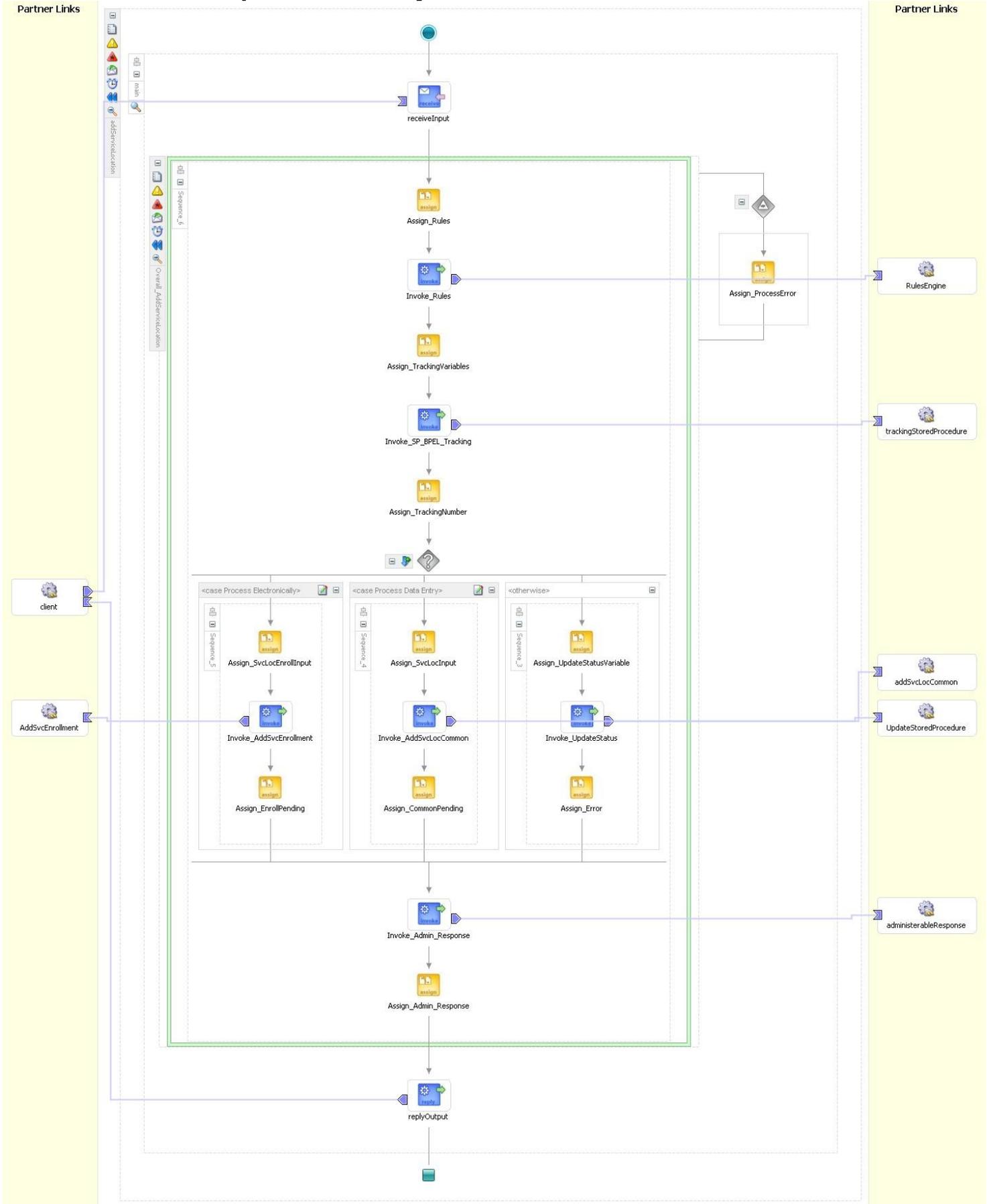
## Apéndice D: Diagrama de flujo de datos Add Service Location



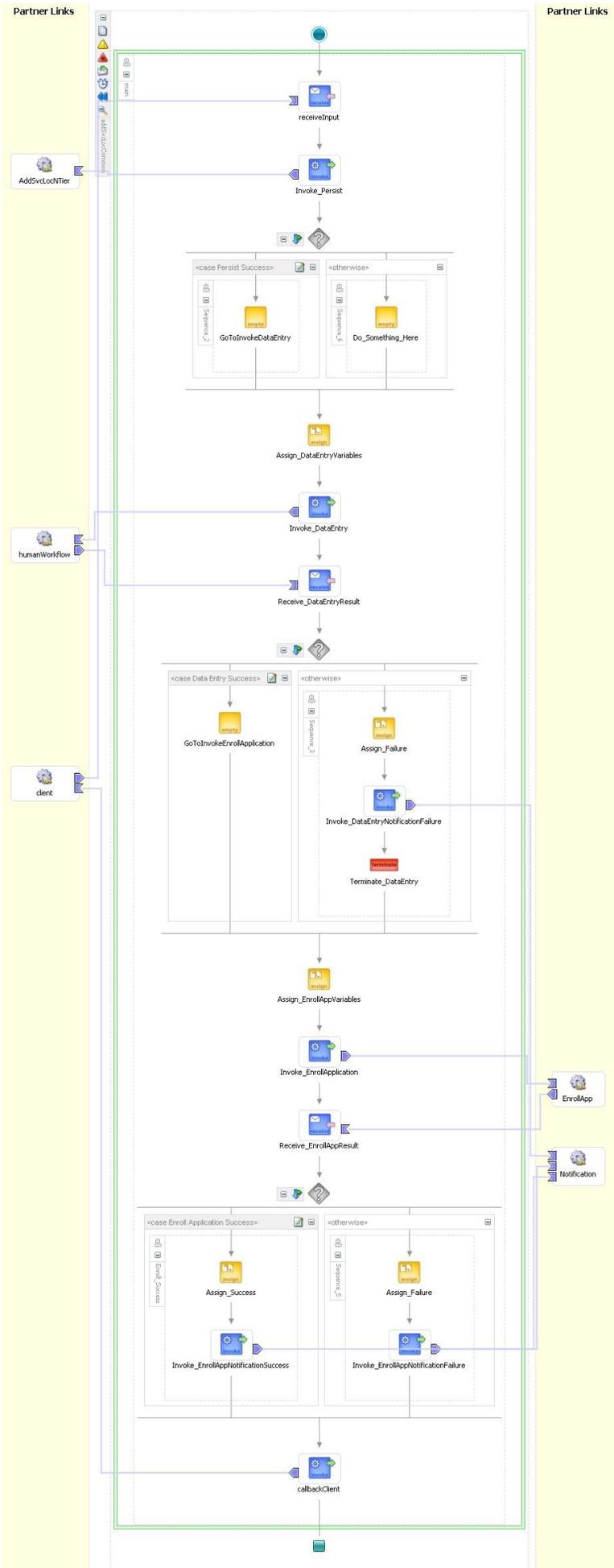
## Apéndice E: Diagrama de Flujo de Datos Add Service Location Enroll App



# Apéndice F: Flujo BPEL Add Service Location

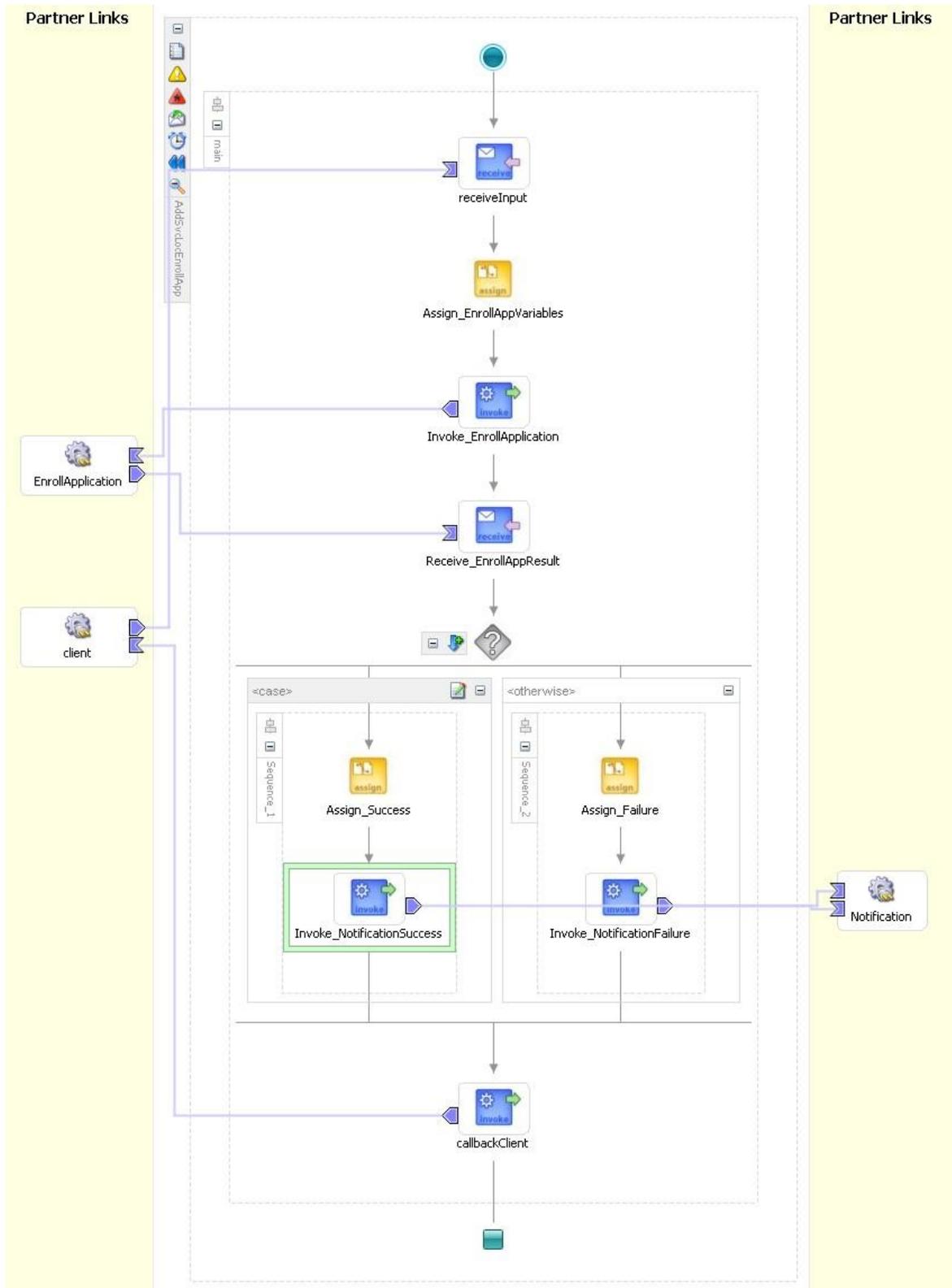


# Apéndice BPEL Add Location



# G: Flujo Service Common

## Apéndice H: Flujo BPEL Add Service Location Enroll App



Página dejada en blanco intencionalmente.

## 9. Diseño de XSD para Enrolamiento de Proveedores

### 9.1 DESCRIPCIÓN DEL DISEÑO

Este diseño proporciona todas las definiciones de XSD para el proceso de enrolamiento de proveedores. Debido a que los otros diseños de enrolamiento de proveedores dependen de este, éste se presenta primero. Se anticipa que estas definiciones puedan requerir revisiones frecuentes conforme vaya madurando los diseños dependientes.

#### 9.1.1 Alcance del Diseño

El alcance se limita al *web service* de enrolamiento de proveedores y cualquier *web service* auxiliar que apoye el enrolamiento de proveedores. Los elementos de los XSDs están encapsulados por área funcional.

#### 9.1.2 interChangeCommon.xsd

Esta definición de esquema es común para los *web services* de interChange. Su *namespace* está definido como iC.MMIS.WS.Common

### 9.2 XSD

Los XSDs actuales para Enrolamiento de Proveedores se anexan en los apéndices para propósitos de revisión. La actualización de versiones se mantiene en el siguiente directorio compartido (al igual que en Visual Source Safe):

<\\usplddssc002\Public\BPEL\XSD>

### 9.3 Namespace

La siguiente table viene de las prácticas aceptadas de BPEL

Namespace	Área
iC.MMIS.WS	interChange web service general
iC.MMIS.WS.Common	interChange web service Common
iC.MMIS.WS.Prov	interChange web service Provider
iC.MMIS.WS.Recipient	interChange web service Recipient
iC.MMIS.WS.Prov.Enrollment	interChange web service Provider Enrollment
iC.MMIS.WS.Notification	interchange web service Notification

### 9.4 Localidad de los Esquemas

Los esquemas asociados con las actividades BPEL se estipulan como locales (solo una referencia del nombre del archivo en el elemento de `schemaLocation`). La centralización de los XSDs en el sistema de archivos y

la distribución a través de técnicas de administración de código fuente resuelven los problemas de versiones y múltiples copias para JDev y el BPEL PM<sup>7</sup>.

## 9.5 CONSIDERACIONES DE PRUEBAS

Para ser efectivo este diseño se debe probar de una manera estratificada para ajustar las limitaciones de JDev. La capa de pruebas es más sofisticada mientras más hacia abajo se procede:

- Formato (editor Emacs)
- Validación de sintaxis XML (JDev)
- Validación XML (JDev)
- Compilación del *Web Service* (JDev)
- Asistente de Asignación de Variables (JDev)
- Procesamiento en tiempo de ejecución (BPEL PM)

Las pruebas asociadas con los últimos dos elementos se asumen parcialmente por diseños dependientes.

## 9.6 Herramientas

Ver arriba.

## 9.7 Ambiente/Dominio

Los dominios específicos para pruebas se crean a través de un sub-directorio del repositorio central XML (XSD) en el servidor Solaris. (Ver localidad del esquema arriba).

## 9.8 TABLA DE MAPEO DE REQUERIMIENTOS

<b>ID requerimiento</b>	<b>Descripción</b>	<b>Elemento de Diseño</b>
N/A	XSDs Enrolamiento de Proveedores	11.1

<sup>7</sup> Este es un cambio significativo de la estructura identificada por las Prácticas Generalmente Aceptadas de BPEL. Los requerimientos de soporte de versiones hicieron necesario este cambio.

## Apéndice A: interChangeCommon.xsd

```
<?xml version="1.0"
encoding="windows-1252"??>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:iC.MMIS.WS.Common="http://www.eds.com/ns/iC.MM
IS.WS.Common"

targetNamespace="http://www.eds.com/ns/iC.MMIS.WS.Co
mmon"
  elementFormDefault="qualified"
  >

  <xsd:annotation>
    <xsd:documentation>
      This document defines interChange common
structures.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:simpleType name="nameTypeDef">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="personal"/>
      <xsd:enumeration value="organization"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:element name="personName"
    type="iC.MMIS.WS.Common:personNameType"/>

  <xsd:complexType name="personNameType">
    <xsd:sequence>

      <xsd:element name="firstName"
        type="iC.MMIS.WS.Common:firstNameType">
      </xsd:element>

      <xsd:element name="middleName"
        type="iC.MMIS.WS.Common:middleNameType"
        minOccurs="0">
      </xsd:element>

      <xsd:element name="lastName"
        type="iC.MMIS.WS.Common:lastNameType">
      </xsd:element>
```

```

    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="firstNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="20"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="middleNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="20"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="lastNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="20"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="organizationNameType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="50"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="usPhoneNumber"
  type="iC.MMIS.WS.Common:usPhoneNumberType"/>

<xsd:complexType name="usPhoneNumberType">
  <xsd:sequence>
    <xsd:element name="areaCode"
      type="iC.MMIS.WS.Common:areaCodeType"/>
    <xsd:element name="exchange"
      type="iC.MMIS.WS.Common:exchangeType"/>
    <xsd:element name="digits"
      type="iC.MMIS.WS.Common:digitsType"/>
    <xsd:element name="extension"
      type="iC.MMIS.WS.Common:extensionType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="areaCodeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d"/>
  </xsd:restriction>

```

```

</xsd:simpleType>

<xsd:simpleType name="exchangeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="digitsType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d\d"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="extensionType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d\d"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="contact"
  type="iC.MMIS.WS.Common:contactType"/>

<xsd:complexType name="contactType">
  <xsd:sequence>
    <xsd:element name="title"
      type="iC.MMIS.WS.Common:titleType"/>
    <xsd:element name="name"
      type="iC.MMIS.WS.Common:personNameType"/>
    <xsd:element name="phoneNumber"

      type="iC.MMIS.WS.Common:usPhoneNumberType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="titleType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="10"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="address"
  type="iC.MMIS.WS.Common:addressType"/>

<xsd:complexType name="addressType">
  <xsd:sequence>
    <xsd:element name="street"
      type="iC.MMIS.WS.Common:streetType"

```

```

        maxOccurs="2"/>

        <xsd:element name="city"
            type="iC.MMIS.WS.Common:cityType"/>

        <xsd:element name="state"
            type="iC.MMIS.WS.Common:stateType"/>

        <xsd:element name="zipcode"
            type="iC.MMIS.WS.Common:zipCodeType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="countyType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="[A-Z][A-Z]"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="streetType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="20"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="cityType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="10"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="zipCodeType">
    <xsd:sequence>
        <xsd:element name="zipBase"
            type="iC.MMIS.WS.Common:zipBaseType"/>

        <xsd:element name="zipExtension"
            type="iC.MMIS.WS.Common:zipExtensionType"
            minOccurs="0"/>

    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="zipBaseType">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\d\d\d\d"/>
    </xsd:restriction>
</xsd:simpleType>

```

```
<xsd:simpleType name="zipExtensionType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d\d"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name="stateType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="AK"/>
    <xsd:enumeration value="AL"/>
    <xsd:enumeration value="AR"/>
    <xsd:enumeration value="AZ"/>
    <xsd:enumeration value="CA"/>
    <xsd:enumeration value="CO"/>
    <xsd:enumeration value="CT"/>
    <xsd:enumeration value="DC"/>
    <xsd:enumeration value="DE"/>
    <xsd:enumeration value="FL"/>
    <xsd:enumeration value="GA"/>
    <xsd:enumeration value="HI"/>
    <xsd:enumeration value="IA"/>
    <xsd:enumeration value="ID"/>
    <xsd:enumeration value="IL"/>
    <xsd:enumeration value="IN"/>
    <xsd:enumeration value="KS"/>
    <xsd:enumeration value="KY"/>
    <xsd:enumeration value="LA"/>
    <xsd:enumeration value="MA"/>
    <xsd:enumeration value="MD"/>
    <xsd:enumeration value="ME"/>
    <xsd:enumeration value="MI"/>
    <xsd:enumeration value="MN"/>
    <xsd:enumeration value="MO"/>
    <xsd:enumeration value="MS"/>
    <xsd:enumeration value="MT"/>
    <xsd:enumeration value="NC"/>
    <xsd:enumeration value="ND"/>
    <xsd:enumeration value="NE"/>
    <xsd:enumeration value="NH"/>
    <xsd:enumeration value="NJ"/>
    <xsd:enumeration value="NM"/>
    <xsd:enumeration value="NV"/>
    <xsd:enumeration value="NY"/>
    <xsd:enumeration value="OH"/>
    <xsd:enumeration value="OK"/>
    <xsd:enumeration value="OR"/>
    <xsd:enumeration value="PA"/>
```

```

        <xsd:enumeration value="RI"/>
        <xsd:enumeration value="SC"/>
        <xsd:enumeration value="SD"/>
        <xsd:enumeration value="TN"/>
        <xsd:enumeration value="TX"/>
        <xsd:enumeration value="UT"/>
        <xsd:enumeration value="VA"/>
        <xsd:enumeration value="VT"/>
        <xsd:enumeration value="WA"/>
        <xsd:enumeration value="WI"/>
        <xsd:enumeration value="WV"/>
        <xsd:enumeration value="WY"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="sakType">
    <xsd:restriction base="xsd:positiveInteger"/>
</xsd:simpleType>

<xsd:element name="requestStatus"
    type="iC.MMIS.WS.Common:requestStatusType"/>

<xsd:complexType name="requestStatusType">
    <xsd:sequence>
        <xsd:element name="status"
            type="iC.MMIS.WS.Common:statusType"/>
        <xsd:element name="failureReason"
            type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="statusType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="success"/>
        <xsd:enumeration value="fail"/>
        <xsd:enumeration value="pending"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType
    name="applicationInfoType">
    <xsd:sequence>
        <xsd:element name="applicationSource"

type="iC.MMIS.WS.Common:applicationSourceType"/>
        <xsd:element name="responseMedia"

type="iC.MMIS.WS.Common:notificationTypeDef"/> 75

```

```

        <xsd:element name="respondToEmailAddress"
            type="xsd:string"/>
        <xsd:element name="contactPhone"

type="iC.MMIS.WS.Common:usPhoneNumberType"/>
        <xsd:element name="faxNumber"

type="iC.MMIS.WS.Common:usPhoneNumberType"/>
        <xsd:element name="demo"
            type="xsd:boolean"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="applicationSourceType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="web"/>
        <xsd:enumeration value="mail"/>
        <xsd:enumeration value="fax"/>
        <xsd:enumeration value="handDelivery"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="notificationTypeDef">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="email"/>
        <xsd:enumeration value="fax"/>
        <xsd:enumeration value="phone"/>
        <xsd:enumeration value="snailMail"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType
name="electronicFundsAccountType">
    <xsd:sequence>
        <xsd:element name="routingNumber"

type="iC.MMIS.WS.Common:routingNumberType"/>
        <xsd:element name="accountNumber"

type="iC.MMIS.WS.Common:accountNumberType"/>
        <xsd:element name="bankAccountType"

type="iC.MMIS.WS.Common:bankAccountTypeDef"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="routingNumberType">

```

```

    <xsd:restriction base="xsd:string">
      <xsd:pattern
value="\d\d\d\d\d\d\d\d\d\d\d\d\d\d"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="accountNumberType">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="\d\d\d\d\d\d\d\d\d\d"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="bankAccountTypeDef">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="checking"/>
      <xsd:enumeration value="savings"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="dateType">
    <xsd:restriction base="xsd:date"/>
  </xsd:simpleType>

  <xsd:element name="language"
    type="iC.MMIS.WS.Common:languageType"/>

  <xsd:simpleType name="languageType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Arabic"/>
      <xsd:enumeration value="Chinese"/>
      <xsd:enumeration value="Czech"/>
      <xsd:enumeration value="English"/>
      <xsd:enumeration value="French"/>
      <xsd:enumeration value="Japanese"/>
      <xsd:enumeration value="Portugese"/>
      <xsd:enumeration value="Spanish"/>
      <xsd:enumeration value="Thai"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>

```

## Apéndice B: interChangeProvider.xsd

```
<?xml version="1.0"
encoding="windows-1252"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:iC.MMIS.WS.Prov="http://www.eds.com/ns/iC.MMIS
.WS.Prov"

xmlns:iC.MMIS.WS.Common="http://www.eds.com/ns/iC.MM
IS.WS.Common"

xmlns:iC.MMIS.WS.Recip="http://www.eds.com/ns/iC.MMI
S.WS.Recip"

targetNamespace="http://www.eds.com/ns/iC.MMIS.WS.Pr
ov"
  elementFormDefault="qualified"
  >

  <xsd:annotation>
    <xsd:documentation>
      Contains the definitions related to the
provider subsystem.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Comm
on"
    schemaLocation="interChangeCommon.xsd"/>

  <xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Reci
p"
    schemaLocation="interChangeRecipient.xsd"/>

  <xsd:element name="id"
    type="iC.MMIS.WS.Prov:idDef"/>

  <xsd:simpleType name="idDef">
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="13"/>
    </xsd:restriction>
  </xsd:simpleType>
```

```

<xsd:simpleType name="idTypeDef">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="tax"/>
    <xsd:enumeration value="medicare"/>
    <xsd:enumeration value="dea"/>
    <xsd:enumeration value="nabp"/>
    <xsd:enumeration value="npi"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:element name="svcLoc"
  type="iC.MMIS.WS.Prov:svcLocType"/>

<xsd:simpleType name="svcLocType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[A-Z]"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="provTypeDef">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="provSpecType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d\d\d"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="provProgramType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="5"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="handicapAccessType">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>

<xsd:simpleType name="billingIndicatorType">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>

<xsd:simpleType name="latitudeType">
  <xsd:restriction base="xsd:string"/>

```

```

</xsd:simpleType>

<xsd:simpleType name="longitudeType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:simpleType name="orgCodeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[123]"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="pubPvtIndType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[BP]"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="withholdFICAType">
  <xsd:restriction base="xsd:boolean"/>
</xsd:simpleType>

<xsd:complexType name="providerIdType">
  <xsd:sequence>
    <xsd:element name="idType"
      type="iC.MMIS.WS.Prov:idTypeDef"/>

    <xsd:element name="id"
      type="iC.MMIS.WS.Prov:idDef"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="licenseNumberType">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="10"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="licenseTypeDef">
  <xsd:restriction base="xsd:string">
    <xsd:maxLength value="2"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="providerNameType">
  <xsd:sequence>
    <xsd:element name="nameType"

```

```

        type="iC.MMIS.WS.Common:nameTypeDef"/>
    <xsd:element name="personName"
        type="iC.MMIS.WS.Common:personNameType"/>
    <xsd:element name="organizationName"
        type="iC.MMIS.WS.Common:organizationNameType"/>
</xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="licenseDef">
    <xsd:sequence>
        <xsd:element name="licenseNumber"
            type="iC.MMIS.WS.Prov:licenseNumberType"/>

        <xsd:element name="licenseType"
            type="iC.MMIS.WS.Prov:licenseTypeDef"/>

        <xsd:element name="state"
            type="iC.MMIS.WS.Common:stateType"/>

        <xsd:element name="certificationDate"
            type="iC.MMIS.WS.Common:dateType"/>

        <xsd:element name="certificationEndDate"
            type="iC.MMIS.WS.Common:dateType"
            minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="providerInfoType">
    <xsd:sequence>
        <xsd:element name="providerName"
            type="iC.MMIS.WS.Prov:providerNameType"/>
        <xsd:element name="providerId"
            type="iC.MMIS.WS.Prov:providerIdType"/>
        <xsd:element name="providerAddress"
            type="iC.MMIS.WS.Common:addressType"/>
        <xsd:element name="providerPhone"
            type="iC.MMIS.WS.Common:usPhoneNumberType"/>
        <xsd:element name="providerContact"
            type="iC.MMIS.WS.Common:contactType"/>
        <xsd:element name="providerLicense"
            type="iC.MMIS.WS.Prov:licenseDef"/>
        <xsd:element name="faxNumber"
            type="iC.MMIS.WS.Common:usPhoneNumberType"/>
    </xsd:sequence>
</xsd:complexType>

```

```

type="iC.MMIS.WS.Common:usPhoneNumberType"/> 81

```

```
</xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="serviceLocationType">
  <xsd:sequence>
    <xsd:element name="svcLoc"
      type="iC.MMIS.WS.Prov:svcLocType"/>
    <xsd:element name="county"
      type="iC.MMIS.WS.Common:countyType"/>
    <xsd:element name="address"
      type="iC.MMIS.WS.Common:addressType"/>
    <xsd:element name="serviceLocationPhone"
      type="iC.MMIS.WS.Common:usPhoneNumberType"/>
    <xsd:element name="provType"
      type="iC.MMIS.WS.Prov:provTypeDef"/>
    <xsd:element name="provSpec"
      type="iC.MMIS.WS.Prov:provSpecType"
      maxOccurs="5"/>
    <xsd:element name="targetRecipientProgram"
      type="iC.MMIS.WS.Recip:recipientProgramType"
      maxOccurs="5"/>
    <xsd:element name="languageSpoken"
      type="iC.MMIS.WS.Prov:languageSpokenType"
      maxOccurs="5"/>
    <xsd:element name="billingIndicator"
      type="iC.MMIS.WS.Prov:billingIndicatorType"/>
    <xsd:element name="handicapAccess"
      type="iC.MMIS.WS.Prov:handicapAccessType"/>
    <xsd:element name="latitude"
      type="iC.MMIS.WS.Prov:latitudeType"/>
    <xsd:element name="longitude"
      type="iC.MMIS.WS.Prov:longitudeType"/>
    <xsd:element name="orgCode"
      type="iC.MMIS.WS.Prov:orgCodeType"/>
    <xsd:element name="pubPvtInd"
      type="iC.MMIS.WS.Prov:pubPvtIndType"/>
    <xsd:element name="withholdFICA"
      type="iC.MMIS.WS.Prov:withholdFICAType"/>
    <xsd:element name="HIPAA837Capable"
      type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="providerCoreDataType">
```

```

    <xsd:sequence>
      <xsd:element name="providerInfo"
        type="iC.MMIS.WS.Prov:providerInfoType"/>

      <xsd:element
name="providerEnrollmentApplicationInfo"

        type="iC.MMIS.WS.Common:applicationInfoType"/>

      <xsd:element name="eftAccount"

        type="iC.MMIS.WS.Common:electronicFundsAccountTy
pe"/>

      <xsd:element name="groupMember"
        type="iC.MMIS.WS.Prov:groupMemberType"/>

      <xsd:element name="certification"
        type="iC.MMIS.WS.Prov:certificationType"
        maxOccurs="5"/>

      <xsd:element name="serviceLocation"
        type="iC.MMIS.WS.Prov:serviceLocationType"
        maxOccurs="5"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:complexType name="medicalGroupType">
  <xsd:annotation>
    <xsd:documentation>
      This lists subordinate members of a group.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="groupMember"
      type="iC.MMIS.WS.Prov:groupMemberType"/>
  </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="groupMemberType">
  <xsd:sequence>
    <xsd:element name="providerId"
      type="iC.MMIS.WS.Prov:providerIdType"/>
    <xsd:element name="serviceLocation"
      type="iC.MMIS.WS.Prov:svcLocType"/>
    <xsd:element name="providerName"
      type="iC.MMIS.WS.Prov:providerNameType"/>
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:element name="endDate"
            type="iC.MMIS.WS.Common:dateType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="certificationType">
    <xsd:sequence>
        <xsd:element name="certTitle"
            type="iC.MMIS.WS.Prov:certTitleType"/>
        <xsd:element name="certDateStart"
            type="iC.MMIS.WS.Common:dateType"/>
        <xsd:element name="certDateEnd"
            type="iC.MMIS.WS.Common:dateType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="certTitleType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="20"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="languageSpokenType">
    <xsd:sequence>
        <xsd:element name="language"
            type="iC.MMIS.WS.Common:languageType"/>
        <xsd:element name="languageStartDate"
            type="iC.MMIS.WS.Common:dateType"/>
        <xsd:element name="languageStopDate"
            type="iC.MMIS.WS.Common:dateType"/>
    </xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

## Apéndice C: interChangeProviderEnrollment.xsd

```
<?xml version="1.0"
encoding="windows-1252"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:iC.MMIS.WS.Prov.Enrollment="http://www.eds.com
/ns/iC.MMIS.WS.Prov.Enrollment"

xmlns:iC.MMIS.WS.Prov="http://www.eds.com/ns/iC.MMIS
.WS.Prov"

xmlns:iC.MMIS.WS.Recip="http://www.eds.com/ns/iC.MMI
S.WS.Recip"

xmlns:iC.MMIS.WS.Common="http://www.eds.com/ns/iC.MM
IS.WS.Common"

targetNamespace="http://www.eds.com/ns/iC.MMIS.WS.Pr
ov.Enrollment"
  elementFormDefault="qualified"
  >
  <xsd:annotation>
    <xsd:documentation>
      Contains the definitions necessary for
provider enrollment main
interface. Include this under wsd1 message
part (payload element).

      Imports provider and common definitions.

      Schema location: In the development
environment there doesn't exist
the defined EDS namespace (www.eds.com/ns/iC).
There are a few
different options available in the interim.
The centralized
version is located at
$ORABPEL_HOME/system/xmllib. The local copy
is under the project directory (requiring
multiple copies, one for
each project).

      Centralized version:

schemaLocation="http://localhost:9700/orabpel/xmllib
```

```

/providerEnrollment.xsd"/>
    Localized Version:
schemaLocation="public_html/providerEnrollment.xsd"/
>

    </xsd:documentation>
</xsd:annotation>

<xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Prov
"
    schemaLocation="interChangeProvider.xsd"/>

<xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Reci
p"
    schemaLocation="interChangeRecipient.xsd"/>

<xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Comm
on"
    schemaLocation="interChangeCommon.xsd"/>

    <xsd:element name="providerEnrollmentRulesRequest"
        type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRulesRequestType"/>

    <xsd:element
name="providerEnrollmentRulesResponse"
        type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRulesResponseType"/>
    <xsd:element name="internalBServiceRequest"
        type="iC.MMIS.WS.Prov.Enrollment:internalBService
eRequestType"/>

    <xsd:element name="dataEntryOCRRequest"
        type="iC.MMIS.WS.Prov.Enrollment:dataEntryOCRReq
uestType"/>

    <xsd:element name="dataEntryOCRResponse"
        type="iC.MMIS.WS.Prov.Enrollment:dataEntryOCRRes
ponseType"/>

    <xsd:element name="providerEnrollmentRequest"
        type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType" />

```

```

<xsd:element name="providerEnrollmentResponse"
  type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entResponseType"/>

<xsd:element name="NTierInterfaceRequest"
  type="iC.MMIS.WS.Prov.Enrollment:NTierInterfaceR
equestType"/>

<xsd:element name="NTierInterfaceResponse"
  type="iC.MMIS.WS.Prov.Enrollment:NTierInterfaceR
esponseType"/>

<xsd:element name="licenseReceiptRequest"
  type="iC.MMIS.WS.Prov.Enrollment:licenseReceiptR
equestType"/>

<xsd:element name="licenseReceiptResponse"
  type="iC.MMIS.WS.Prov.Enrollment:licenseReceiptR
esponseType"/>

<xsd:element name="licenseValidationRequest"
  type="iC.MMIS.WS.Prov.Enrollment:licenseValidati
onRequestType"/>

<xsd:element name="licenseValidationResponse"
  type="iC.MMIS.WS.Prov.Enrollment:licenseValidati
onResponseType"/>

<xsd:element name="notificationRequest"
  type="iC.MMIS.WS.Prov.Enrollment:notificationReq
uestType"/>

<xsd:element name="notificationResponse"
  type="iC.MMIS.WS.Common:requestStatusType"/>

<xsd:element name="enrollmentApprovalRequest"
  type="iC.MMIS.WS.Prov.Enrollment:enrollmentAppro
valRequestType"/>

<xsd:element name="enrollmentApprovalResponse"
  type="iC.MMIS.WS.Prov.Enrollment:enrollmentAppro
valResponseType"/>

<xsd:complexType name="dataEntryOCRRequestType">
  <xsd:sequence>
    <xsd:element name="trackingNumber"

type="iC.MMIS.WS.Prov.Enrollment:trackingNumber87

```

```

type"/>
    <xsd:element name="providerEnrollmentRequest"
        type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
    <xsd:element name="webReceiptTimeStamp"
        type="xsd:dateTime"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="dataEntryOCRResponseType">
    <xsd:sequence>
        <xsd:element name="providerEnrollmentRequest"
            type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
        <xsd:element name="interChangeSakProv"
            type="iC.MMIS.WS.Common:sakType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="providerEnrollmentRulesRequestType">
    <xsd:sequence>
        <xsd:element name="trackingNumber"
            type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="providerEnrollmentType"
            type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entTypeDef"/>
        <xsd:element name="providerEnrollmentRequest"
            type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="providerEnrollmentTypeDef">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="web"/>
        <xsd:enumeration value="mail"/>
        <xsd:enumeration value="fax"/>
    </xsd:restriction>
</xsd:simpleType>

```

```

    <xsd:complexType
name="providerEnrollmentRulesResponseType">
    <xsd:sequence>
        <xsd:element name="action"

type="iC.MMIS.WS.Prov.Enrollment:rulesActionType
"/>
        <xsd:element name="rulesResponse"
type="xsd:string"/>
        <xsd:element name="providerEnrollmentRequest"

type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="rulesActionType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration
value="processElectronically"/>
        <xsd:enumeration value="processDataEntry"/>
        <xsd:enumeration value="rejectRequest"/>
    </xsd:restriction>
</xsd:simpleType>

<xsd:complexType
name="internalBServiceRequestType">
    <xsd:sequence>
        <xsd:element name="trackingNumber"

type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="providerEnrollmentRequest"

type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
        <xsd:element name="interChangeSakProv"
type="iC.MMIS.WS.Common:sakType"/>
        <xsd:element name="licenseReceived"
type="xsd:boolean"/>
        <xsd:element name="licenseInfo"

type="iC.MMIS.WS.Prov.Enrollment:licenseReceiptR
esponseType"/>
        <xsd:element name="webReceiptTimeStamp"
type="xsd:dateTime"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType
name="providerEnrollmentRequestType">
  <xsd:sequence>

    <xsd:element name="providerCoreData"

      type="iC.MMIS.WS.Prov:providerCoreDataType"/>

  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NTierInterfaceRequestType">
  <xsd:sequence>
    <xsd:element name="providerEnrollmentRequest"

      type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="NTierInterfaceResponseType">
  <xsd:sequence>
    <xsd:element name="interChangeSakProv"
      type="iC.MMIS.WS.Common:sakType"/>
    <xsd:element name="requestStatus"

      type="iC.MMIS.WS.Common:requestStatusType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="providerEnrollmentApplicationInfo">
  <xsd:sequence>
    <xsd:element name="applicationSource"

      type="iC.MMIS.WS.Common:applicationSourceType"/>
    <xsd:element name="responseMedia"

      type="iC.MMIS.WS.Common:notificationTypeDef"/>
    <xsd:element name="respondToEmailAddress"
      type="xsd:string"/>
    <xsd:element name="demo"
      type="xsd:boolean"/>
  </xsd:sequence>
</xsd:complexType>

```

```

    <xsd:complexType
name="enrollmentApprovalRequestType">
    <xsd:sequence>
        <xsd:element name="trackingNumber"

            type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="providerEnrollmentRequest"

            type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
        <xsd:element name="licenseReceiptData"

            type="iC.MMIS.WS.Prov.Enrollment:licenseReceiptR
esponseType"/>
        <xsd:element name="licenseValidationData"

            type="iC.MMIS.WS.Prov.Enrollment:licenseValidati
onResponseType"/>
        <xsd:element name="interChangeSakProv"

            type="iC.MMIS.WS.Common:sakType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="enrollmentApprovalResponseType">
    <xsd:sequence>
        <xsd:element name="approverId"

            type="iC.MMIS.WS.Common:contactType"/>
        <xsd:element name="requestStatus"

            type="iC.MMIS.WS.Common:requestStatusType"/>
        <xsd:element name="statusTimestamp"

            type="xsd:dateTime"/>
        <xsd:element name="supplementalInfo"

            type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="notificationRequestType">
    <xsd:sequence>
        <xsd:element name="trackingNumber"

            type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="notificationType"

            type="iC.MMIS.WS.Common:notificationTypeDef"/>

```

```

        <xsd:element name="requestStatus"
type="iC.MMIS.WS.Common:requestStatusType"/>
        <xsd:element name="timeStamp"
            type="xsd:dateTime"/>
        <xsd:element name="approver"
            type="iC.MMIS.WS.Common:contactType"/>
        <xsd:element name="emailAddress"
            type="xsd:string"/>
        <xsd:element name="providerContact"
            type="iC.MMIS.WS.Common:contactType"/>
        <xsd:element name="physicalAddress"
            type="iC.MMIS.WS.Common:addressType"/>
        <xsd:element name="faxNumber"

type="iC.MMIS.WS.Common:usPhoneNumberType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="licenseValidationRequestType">
    <xsd:sequence>
        <xsd:element name="trackingNumber"

type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="license"
            type="iC.MMIS.WS.Prov:licenseDef"/>
        <xsd:element name="licenseTracking"

type="iC.MMIS.WS.Prov.Enrollment:licenseReceiptR
esponseType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="licenseValidationResponseType">
    <xsd:sequence>
        <xsd:element name="licenseValidationStatus"

type="iC.MMIS.WS.Prov.Enrollment:licenseValidati
onStatusType"/>
        <xsd:element name="validationTimeStamp"
            type="xsd:dateTime"/>
        <xsd:element name="validator"
            type="iC.MMIS.WS.Common:contactType"/>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:simpleType
name="licenseValidationStatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="valid"/>
    <xsd:enumeration value="expired"/>
    <xsd:enumeration value="suspicious"/>
    <xsd:enumeration value="failure"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="licenseValidationIdType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:complexType name="licenseReceiptRequestType">
  <xsd:sequence>
    <xsd:element name="trackingNumber"

type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
    <xsd:element name="providerEnrollmentRequest"

type="iC.MMIS.WS.Prov.Enrollment:providerEnrollm
entRequestType"/>
    <xsd:element name="interChangeSakProv"
type="iC.MMIS.WS.Common:sakType"/>
    <xsd:element name="webReceiptTimeStamp"
type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType
name="licenseReceiptResponseType">
  <xsd:sequence>
    <xsd:element name="clerkId"

type="iC.MMIS.WS.Prov.Enrollment:clerkIdType"/>
    <xsd:element name="requestStatus"

type="iC.MMIS.WS.Common:requestStatusType"/>
    <xsd:element name="licenseReceiptTimestamp"
type="xsd:dateTime"/>
    <xsd:element name="licenseImageKey"

type="iC.MMIS.WS.Prov.Enrollment:licenseImageKey
Type"/>
    <xsd:element name="supplementalTrackingInfo" 93

```

```

        type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="clerkIdType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:simpleType name="licenseImageKeyType">
    <xsd:restriction base="xsd:string"/>
</xsd:simpleType>

<xsd:complexType
name="providerEnrollmentResponseType">
    <xsd:sequence>
        <xsd:element name="requestStatus"

            type="iC.MMIS.WS.Common:requestStatusType"/>

        <xsd:element name="trackingNumber"

            type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
        <xsd:element name="licenseInstructions"
            type="xsd:string">
            <xsd:annotation>
                <xsd:documentation>
                    Contains instructions to the requestor
on where to mail the
                    provider's license and to provide the
tracking number to
                    expedidite handling.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="trackingNumberType">
    <xsd:restriction base="xsd:integer"/>
</xsd:simpleType>

<xsd:element name="inputProviderName"
    type="iC.MMIS.WS.Prov:providerNameType"/>

<xsd:element name="license"
    type="iC.MMIS.WS.Prov:licenseDef"/>

```

```
<xsd:element name="queryEnrollmentStatusRequest"
  type="iC.MMIS.WS.Prov.Enrollment:queryEnrollment
StatusRequestType"/>
```

```
<xsd:element name="queryEnrollmentStatusResponse"
  type="iC.MMIS.WS.Prov.Enrollment:queryEnrollment
StatusResponseType"/>
```

```
<xsd:complexType
name="queryEnrollmentStatusRequestType">
  <xsd:annotation>
    <xsd:documentation>
      The query request is intended to offer
either input options of
      tracking number or name. Both are not
required to perform the
      query.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="trackingNumber"
      type="iC.MMIS.WS.Prov.Enrollment:trackingNumberT
ype"/>
    <xsd:element name="enrollmentName"
      type="iC.MMIS.WS.Prov:providerNameType"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType
name="queryEnrollmentStatusResponseType">
  <xsd:sequence>
    <xsd:element name="enrollmentStatus"
      type="xsd:string"/>
    <xsd:element name="supplementalInfo"
      type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:element name="addServiceLocationRequest"
  type="iC.MMIS.WS.Prov.Enrollment:addServiceLocat
ionRequestType"/>
```

```
<xsd:complexType
name="addServiceLocationRequestType">
  <xsd:sequence>
    <xsd:element name="providerId"
```

```

        type="iC.MMIS.WS.Prov:providerIdType"/>
        <xsd:element name="serviceLocation"

        type="iC.MMIS.WS.Prov:serviceLocationType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="addServiceLocationResponse"
    type="iC.MMIS.WS.Prov.Enrollment:addServiceLocat
ionResponseType"/>

<xsd:complexType
name="addServiceLocationResponseType">
    <xsd:sequence>
        <xsd:element name="requestStatus"

        type="iC.MMIS.WS.Common:requestStatusType"/>
        <xsd:element name="stateContact"
            type="iC.MMIS.WS.Common:contactType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="queryProviderInfoRequest"
    type="iC.MMIS.WS.Prov.Enrollment:queryProviderIn
foRequestType"/>

<xsd:complexType
name="queryProviderInfoRequestType">
    <xsd:sequence>
        <xsd:element name="providerId"
            type="iC.MMIS.WS.Prov:providerIdType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:element name="queryProviderInfoResponse"
    type="iC.MMIS.WS.Prov.Enrollment:queryProviderIn
foResponseType"/>

<xsd:complexType
name="queryProviderInfoResponseType">
    <xsd:sequence>
        <xsd:element name="providerCoreData"
            type="iC.MMIS.WS.Prov:providerCoreData"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## Apéndice D: interChangeRecipient.xsd

```
<?xml version="1.0"
encoding="windows-1252"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:iC.MMIS.WS.Recip="http://www.eds.com/ns/iC.MMI
S.WS.Recip"

targetNamespace="http://www.eds.com/ns/iC.MMIS.WS.Re
cip"
  elementFormDefault="qualified"
  >

  <xsd:annotation>
    <xsd:documentation>
      Contains the definitions related to the
recipient subsystem.

      Currently these definitions are quite limited
as the entries are only
      associated with the provider enrollment
process.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:simpleType name="recipientProgramType">
    <xsd:annotation>
      <xsd:documentation>
        This enumerated type identifies recipient
programs.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="TXIX"/>
      <xsd:enumeration value="W-IHA"/>
      <xsd:enumeration value="W-IHC"/>
      <xsd:enumeration value="SLA"/>
      <xsd:enumeration value="NET"/>
      <xsd:enumeration value="CCP"/>
      <xsd:enumeration value="SLMB"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

## Apéndice E: Notification.xsd

```
<?xml version="1.0"
encoding="windows-1252"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:iC.MMIS.WS.Common="http://www.eds.com/ns/iC.MM
IS.WS.Common"

xmlns:iC.MMIS.WS.Notification="http://www.eds.com/ns
/iC.MMIS.WS.Notification"

targetNamespace="http://www.eds.com/ns/iC.MMIS.WS.No
tification"
  elementFormDefault="qualified"
  >

  <xsd:annotation>
    <xsd:documentation>
      This document defines interChange notification
structures.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:import
    namespace="http://www.eds.com/ns/iC.MMIS.WS.Comm
on"
    schemaLocation="interChangeCommon.xsd"/>

  <xsd:element name="notificationRequest"
    type="iC.MMIS.WS.Notification:notificationReques
tType"/>

  <xsd:element name="notificationResponse"
    type="iC.MMIS.WS.Common:requestStatusType"/>

  <xsd:complexType name="notificationRequestType">
    <xsd:sequence>
      <xsd:element name="webServiceType"

type="iC.MMIS.WS.Notification:webServiceTypeDef"
/>
      <xsd:element name="trackingNumber"

type="iC.MMIS.WS.Common:trackingNumberType"/>
    <xsd:element name="notificationType"
```

```

    type="iC.MMIS.WS.Common:notificationTypeDef"/>
    <xsd:element name="requestStatus"

type="iC.MMIS.WS.Common:requestStatusType"/>
    <xsd:element name="timeStamp"
        type="xsd:dateTime"/>
    <xsd:element name="approver"
        type="iC.MMIS.WS.Common:contactType"/>
    <xsd:element name="emailAddress"
        type="xsd:string"/>
    <xsd:element name="providerContact"
        type="iC.MMIS.WS.Common:contactType"/>
    <xsd:element name="physicalAddress"
        type="iC.MMIS.WS.Common:addressType"/>
    <xsd:element name="faxNumber"

type="iC.MMIS.WS.Common:usPhoneNumberType"/>
</xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="webServiceTypeDef">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="providerEnrollment"/>
        <xsd:enumeration value="addServiceLocation"/>
        <xsd:enumeration value="priorAuthorization"/>
    </xsd:restriction>
</xsd:simpleType>

</xsd:schema>

```

## 10. Diseño de *Web Service* de Notificación

---

### 10.1 DESCRIPCIÓN DEL DISEÑO

El proceso de notificación recibe una entrada del usuario, en la cual el usuario especifica el método de notificación que debe utilizarse para contactarlo. Los métodos que se tiene planeado implementar para la notificación son correo electrónico y correo convencional.

Nota: Cuando la notificación se invoca requiere de la siguiente información

trackingNumber, NotificationType, webServiceType,  
providerContact(firstName& lastName), emailAddress, status)

#### 10.1.1 Alcance del Diseño

#### 10.1.2 Elemento de Diseño

El proceso de Notificación contiene dos partes: la generación de cartas, y la generación de correo electrónico. La generación de cartas utiliza la capacidad actual de interChange para generarlas. La generación de cartas requiere del uso de un procedimiento almacenado, el cual llena las tablas requeridas. Una vez que las tablas han sido llenadas, la aplicación actual cuyo *script* se llama “SYSJDLTR” genera cartas para todas las peticiones que han sido escritas en la tabla “T\_LTR\_RQST\_TEMPLATE”. BPEL está pre-configurado para enviar notificaciones usando el servicio inalámbrico de Oracle. El servicio de notificación soporta el envío de correo electrónico a través de SMTP así como la recepción desde cuentas de correo basadas en IMAP o POP. Debido a lo sensible de la información y los requerimientos de HIPAA, si se proporciona un correo electrónico con la aplicación podemos enviar una respuesta interina de éxito o fracaso y hacer referencia al número de guía, tipo de petición, referencia a la carta y cuando puede ser esperada, así como un número de contacto en caso de que la carta no sea recibida en un tiempo adecuado. Tanto la generación de cartas como las notificaciones por correo electrónico requieren que el sistema identifique qué contenido se va a utilizar como información para el correo electrónico o la carta. Un número de identificación único llamado “sak\_letter” será asignado a las cartas que son generadas. De momento el almacenamiento de las cartas y correos electrónicos se hace a través de tablas en la base de datos. Esto permite de una manera sencilla la actualización, borrado o agregación de nuevas cartas y correos electrónicos. Cada respuesta de correo electrónico tiene formato HTML y contiene variables que se identifican comenzando con el símbolo “@”, el cual se reemplaza con los datos del proveedor durante el proceso de notificación.

#### 10.1.3 Flujo de Trabajo Humano

Ninguno identificado por el momento

#### 10.1.4 Manejo de Excepciones

Si el procedimiento almacenado falla entonces tendremos un código diferente de “0” que permita reconocer el tipo de falla (0 significa éxito).

## 10.2 XSD

Cada *Web Service* contiene un valor en código duro para identificar el tipo de *web service* (*webServiceType*). El *xsd* del servicio de notificación (*notification.xsd*) se ha modificado para soportar los siguientes tipos *providerEnrollment*, *addServiceLocation* y *priorAuthorization*.

## 10.3 Namespace

<http://www.eds.com/ns/iC.MMIS.WS.Prov.Enrollment>

## 10.4 Locación de los Esquemas

- [interChangeCommon.xsd](#)
- [interChangeRecipient.xsd](#)
- [interChangeProvider.xsd](#)

## 10.5 WSDL

El proceso de notificación hace uso de los siguientes archivos WSDL:

- *NotificationService\_test\_.wsdl*
- *selectLetterID\_test\_.wsdl*
- *selectResponse\_test\_.wsdl*
- *updateSak\_test\_.wsdl*
- *callInsertLetter\_test\_.wsdl*

## 10.6 TABLAS DE DATOS

### 10.6.1 BPEL

Para propósitos de pruebas hemos creados la misma estructura de tablas que el sistema actual está utilizando.

### 10.6.2 interChange

Las tablas que estamos utilizando se encuentran en la base de datos *cfmist1* que es parte de *interChange*.

## 10.7 DEPENDENCIAS

El diseño depende del acceso a un servidor SMTP para poder enviar correos electrónicos. El sistema *interChange* que está hospedado en un servidor UNIX cuenta con el demonio *sendmail* para el envío de correos.

## 10.8 CONSIDERACIONES DE PRUEBA

### 10.8.1 Herramientas

Estamos utilizando el procedimiento almacenado localizado en el Apéndice A para probar si es posible llenar las tablas requeridas sin el uso de la Utilidad de Generación de Cartas. Las pruebas se realizan por el momento a través de SQL PLUS.

### 10.8.2 Ambiente/Dominio

La versión más actual del servicio de Notificación se encuentra momentaneamente en el dominio “test” de *bpeltest*.

## Apéndice A: Procedimiento almacenado

```
-- Procedimiento para insertar un campo en la tabla de cartas
CREATE OR REPLACE PROCEDURE p_bpel_insert_letter
  (template_id IN CHAR, to_id IN CHAR, dsc_extra_data VARCHAR2)
IS
  sak_ltr_rqst  NUMBER(9);
  sak_ltr       NUMBER(9);
  sak_ltr_tmplt NUMBER(9);
  sak_query     NUMBER(9);
  gen_mode      CHAR(1);
  dte_parm1    NUMBER(8);

BEGIN

  -- Obtener fecha de ciclo de t_system_parms
  SELECT date_parm_1
  INTO   dte_parm1
  FROM   t_system_parms
  WHERE  nam_program = 'AIMCYCLE';

  -- Obtener siguiente sak_letter_request de tabla t_system_keys
  SELECT sak
  INTO   sak_ltr_rqst
  FROM   t_system_keys
  WHERE  sak_name = 'SAK_LETTER_REQUEST'
  FOR UPDATE;

  UPDATE t_system_keys
  SET    sak = sak + 1
  WHERE  sak_name = 'SAK_LETTER_REQUEST';

  COMMIT; -- Necesario para realizar FOR UPDATE

  -- Obtener sak_letter y sak_letter_template
  SELECT ltr.sak_letter,
         ltrtmp.sak_letter_template,
         ltrtmp.sak_query
  INTO   sak_ltr,
         sak_ltr_tmplt,
         sak_query
  FROM   t_letter ltr,
         t_letter_template ltrtmp
  WHERE  ltr.id_letter = template_id
  AND    ltr.sak_letter = ltrtmp.sak_letter;

  INSERT INTO t_letter_request
  (sak_letter_request, id_clerk, dte_request)
  VALUES (sak_ltr_rqst, 'SYS', to_char(sysdate, 'YYYYMMDD'));

  gen_mode := 'B';

  -- Insertar ltr_rqst template
  INSERT INTO t_ltr_rqst_templat
  (sak_letter,
   sak_letter_template,
   sak_letter_request,
   id_people,
   dte_generate,
```

```
        ind_generate_mode,  
        dsc_extra_data)  
VALUES (sak_ltr,  
        sak_ltr_tmplt,  
        sak_ltr_rqst,  
        to_id,  
        dte_parm1,  
        gen_mode,  
        dsc_extra_data);  
COMMIT;  
  
END p_bpel_insert_letter;
```

# 11. Diseño para desarrollo de *Web Services* de Lista de Trabajo

---

## 11.1 DESCRIPCIÓN DEL DISEÑO

Este documento describe los *web services* de lista de trabajo para enrolamiento de proveedores. Actualmente estamos enfrenando una falta de servicios de autenticación, los cuales requieren que el usuario ingrese a la lista de trabajo. Versiones futuras del servidor BPEL podrían no necesitar autenticación de credenciales, para hacer esto vamos a utilizar adaptadores LDAP. Actualmente la lista de trabajo requiere autenticación de usuario utilizando el API de la clase, una vez que el usuario ha sido autenticado, utilizamos las llamadas al API para mostrar las tareas de usuario actuales, obtener los roles del usuario y hacer disponibles los roles como acciones para cada tarea, donde una tarea puede ser alguna de las siguientes: Accept, Reject, Suspend, Renew y Escalate. Actualmente estamos utilizando autenticación de usuario independiente a través de llamadas al API, pero en el futuro utilizaremos los *web services* LDAP en la capa media como un paso para tener un solo ingreso en el sistema. Favor de referirse al Alcance de Diseño, sección 2.1 debajo.

Tendremos un archivo JAR con las importaciones necesarias para la lista de trabajo. A continuación se muestra la estructura de dichos paquetes:

- **WorklistAPI.JAR**
  - oracle.tip.pc.api.worklist.\*
  - oracle.tip.pc.api.worklist.IWorklistTask
  - oracle.tip.pc.services.hw.worklist.WorklistService
  - com.oracle.services.bpel.\*

Llamadas a API utilizadas con los imports previos:

- Authentication: authenticateUser(User Name, Password)
- Suspend: suspendTask(Worklist Context, Task ID)
- Escalate: escalateTask(Worklist Context, Task ID)
- Renew: renewTask(Worklist Context, Task ID, Duration Days)
- Reject: customTaskOperation(Worklist Context, Task ID, "REJECT")
- Accept: customTaskOperation(Worklist Context, Task ID, "ACCEPT")
- Task Information: getWorklistTaskDetails(Worklist Context, Task ID)

- **Objetos/Clases:**

La fase de reconstrucción de la nueva lista de trabajo al ser transicionada desde el prototipo requiere el desarrollo de las siguientes clases:

- **Clase Work-list:** Después de hacer un análisis de los resultados en la salida del *web service* BPEL, esta clase crea una instancia de las siguientes clases:
  - **Clase Tasks:** Esta clase almacena todas las tareas y sus respectivas propiedades. Esto permite un mejor acceso y organización de los datos.
  - **Clase Tasks View:** Esta clase hace uso de objetos visuales para desplegar las tareas y sus propiedades, permitiendo el acceso eficiente a la información a través de cualquier GUI y la comunicación con otros objetos.
  - **Clase User Information:** Esta clase almacena toda la información del usuario, la cual se requiere para enviarle comandos al *web service*, también se utiliza para propósitos de visualización.

Ver el Apéndice D para más información acerca de estas clases.

La reconstrucción también hará uso del objeto “Entity Maintenance Panel” (Panel de mantenimiento de entidades) el cual se proporciona con la instalación de DotNetNuke/SODA. Este objeto sigue el formato visual utilizado en el sistema ICE, por lo que necesitamos utilizar el mismo.

- **Embedded Java**

Para cada acción hay un componente *embedded java* el cual maneja dicha acción, éste requiere el id de la tarea y autenticación del usuario. Estos componentes se activan a través de un componente *switch* en el proceso BPEL el cual requiere un comando y autenticación del usuario por del lado del cliente.

## 11.2 ALCANCE DEL DISEÑO

El flujo de trabajo humano actual requiere que el usuario se firme cada vez que quiera ver la lista de trabajo. La autenticación del usuario en una computadora aislada se realiza cada vez que el usuario se firma. La autenticación también se realiza cada vez que el usuario envía una acción: Listar tareas actuales, aceptar una tarea, rechazar, renovar, suspender y escalar una tarea.

El utilizar *web services* LDAP en la capa media nos permite una firma única. Lo que significa que una vez que el usuario se ha firmado en su estación de trabajo, la autenticación solo ocurre una vez. Ya no será necesario autenticar las acciones enviadas por el usuario.

## 11.3 WEB SERVICE DEL PROCESO BPEL

Este *web service* inicia las operaciones de la lista de trabajo, el proceso comienza al recibir los datos de entrada, el siguiente paso es verificar las credenciales de autenticación<sup>8</sup>, una vez que las credenciales han sido autenticadas la operación se puede llevar a cabo en una sentencia *switch*. Las diferentes operaciones proporcionadas por este *web service* son:

- Listar Tareas (List Tasks)
- Aceptar (Accept)
- Rechazar (Reject)
- Renovar (Renew)
- Suspend (Suspend)
- Escalate (Escalar)

Estas operaciones actualizan la lista de trabajo y le envían la salida adecuada a quien las inició.

### 11.3.1 FLUJO DE TRABAJO HUMANO

- Este *Web Service* se creó con el propósito de proporcionar un *front-end* para el API de lista de trabajo proporcionado por el servidor BPEL. Además de la GUI .NET que explota el WSDL, se lleva a cabo una actividad de flujo de trabajo humano a través de una GUI *ad hoc*
- Las notificaciones se proporcionan para hacerle saber al asignado en cuanto reciba una tarea que requiere su acción
- No se han identificado restricciones de tiempo hasta esta versión del documento

### 11.3.2 MANEJO DE EXCEPCIONES

Actualmente se manejan dos excepciones verificadas

- *WorklistServiceException*. Esta excepción se atrapa si el ID de la tarea no existe
- *PCException*. Esta excepción se atrapa si el comando solicitado no puede ser aplicado a la tarea

### 11.3.3 SENSORES

Cada una de las acciones disponibles para el usuario dependiendo de sus roles tiene su sensor. Este sensor es una simple asignación que genera una marca de tiempo para cada acción.

## 11.4 GUI ASP.NET / C#

Ver Apéndice B

---

<sup>8</sup> Futuras versiones de BPEL pueden no necesitar autenticación de credenciales

## 11.5 WSDL

El archivo WSDL asociado con el proceso protótipo de la lista de trabajo BPEL contiene las siguientes entradas: nombre de usuario, contraseña, comando, y fecha de renovación. Ver Apéndice D.

## 11.6 DEPENDENCIAS

Ver Apéndice A

## 11.7 CONSIDERACIONES DE PRUEBA

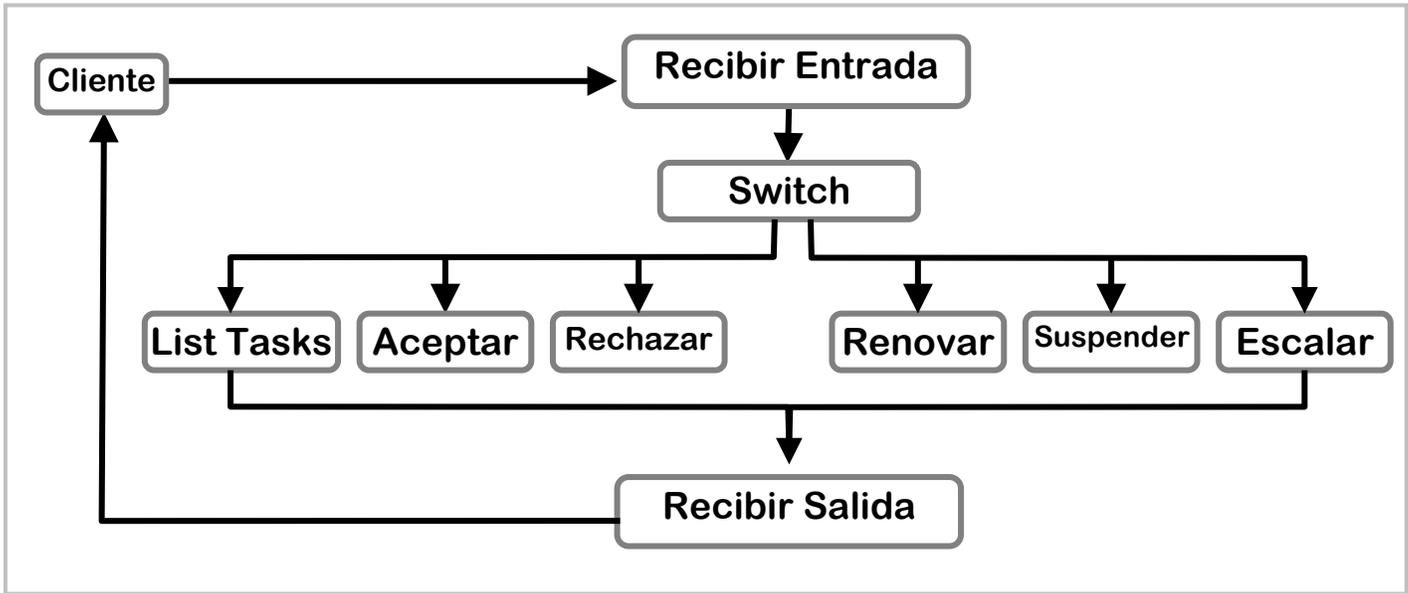
Se deben aplicar algunas pruebas para asegurarse de que el *web service* de la lista de trabajo BPEL tenga una interfaz correcta con el portal DotNetNuke y la GUI diseñada en .NET. La funcionalidad de la GUI debe ser también probada para estar seguro de que las acciones se lleven a cabo adecuadamente. Una manera de hacerlo es a través de una verificación visual de la salida de la funcionalidad a través de las siguientes acciones: Listar tareas, actualizar tareas, editar tareas, aceptar tareas, rechazar tareas, suspender tareas, escalar tareas, renovar tareas, autenticaciones de usuario y roles.

## 11.8 ENTREGABLES

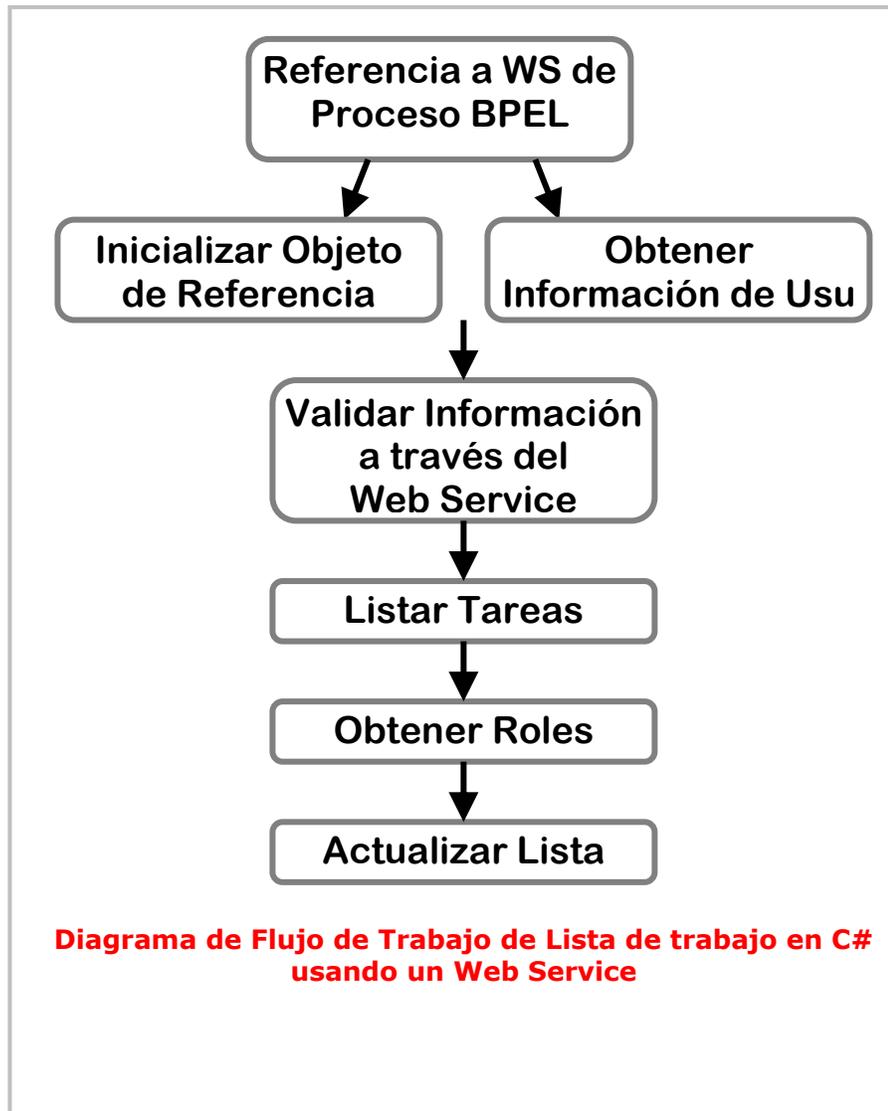
La siguiente es una lista de los archivos utilizados (la lista de archivos es sujeta a cambios):

- NewWorkList.bpel
- NewWorkList.wsdl
- NewWorkList.jpr
- bpel.xml
- build.xml
- graphics.xml
- WebApplication1.csproj
- WebApplication1.sln
- WebForm1.aspx
- WebForm1.aspx.cs
- WebForm2.aspx
- WebForm2.aspx.cs

## Apéndice A: Diagramas de Flujo de Trabajo

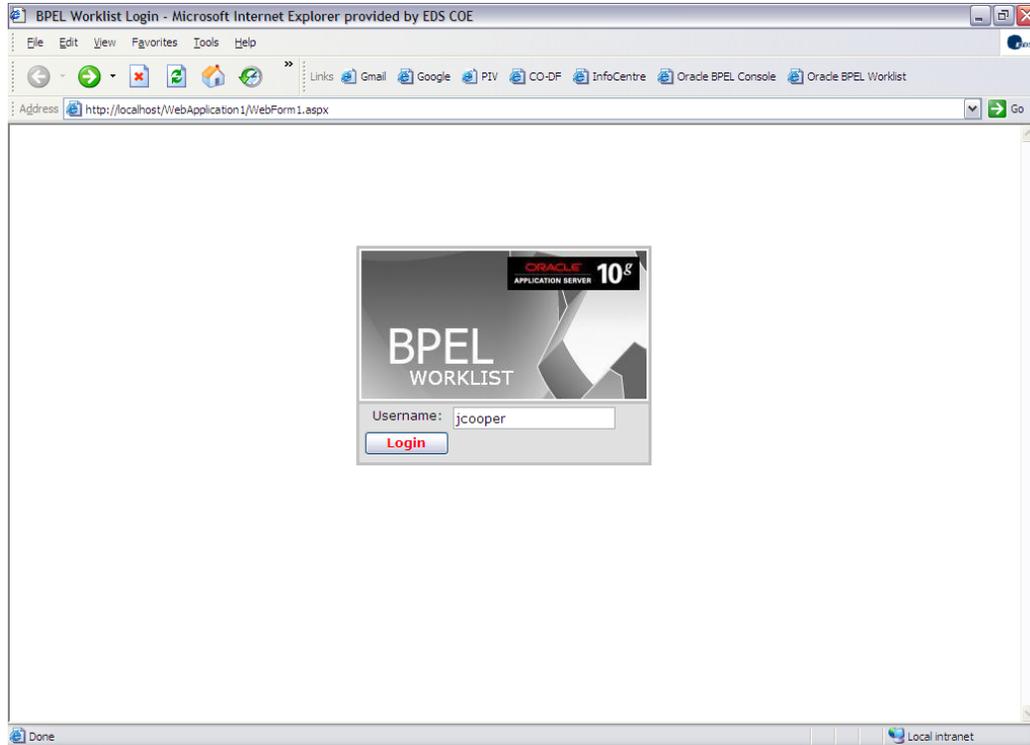


**Diagrama de Web Service del proceso BPEL Lista de Trabajo**

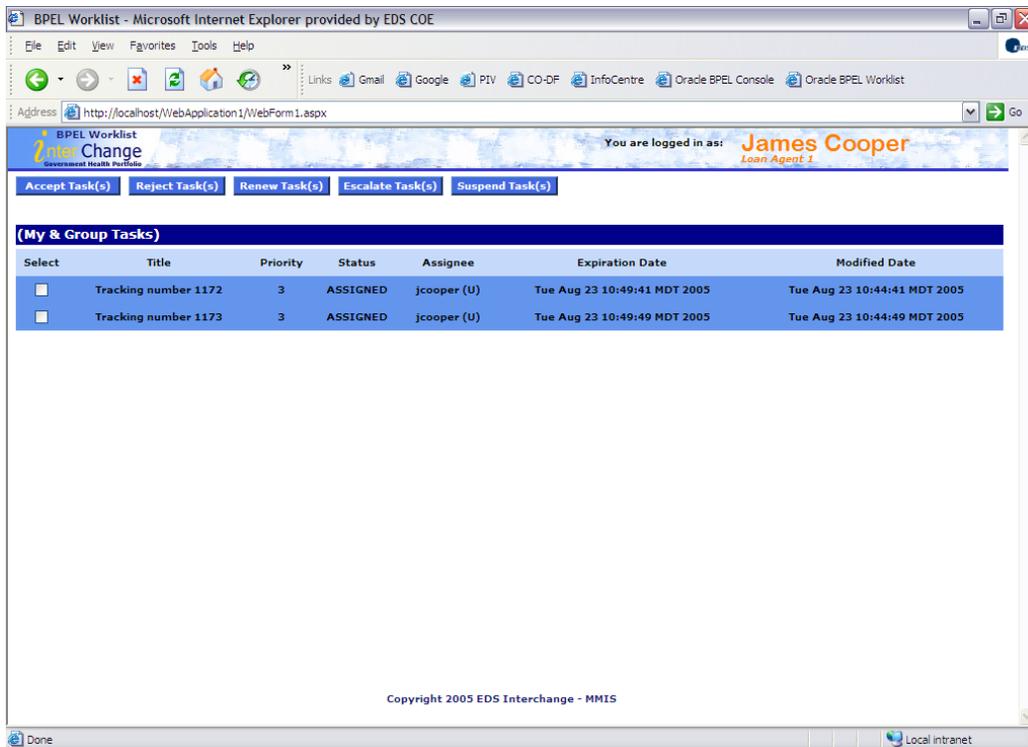


**Diagrama de Flujo de Trabajo de Lista de trabajo en C# usando un Web Service**

## Apéndice B: Diseño actual de la GUI

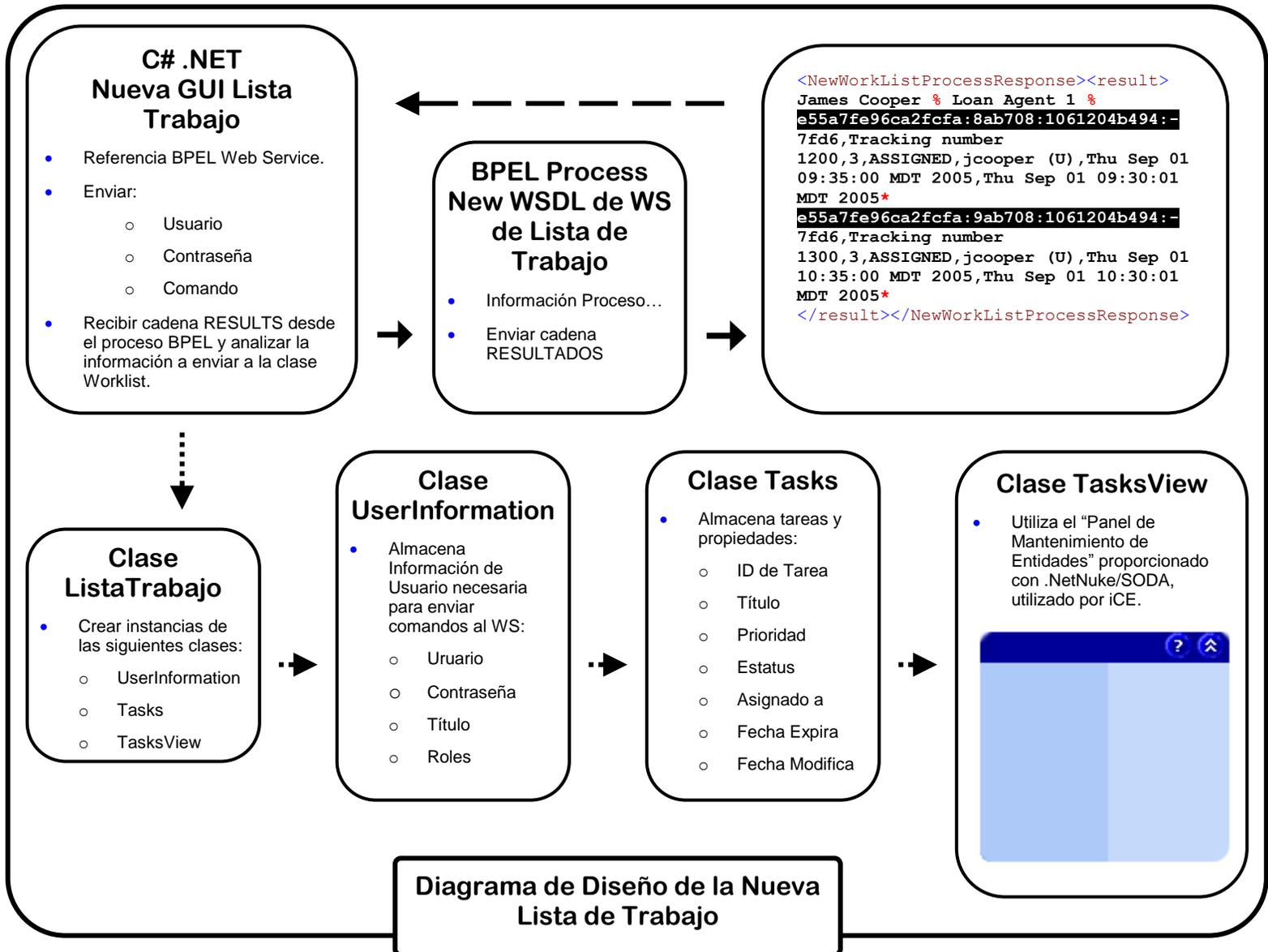


**Pantalla 1:** Pantalla de login, no requiere password.



**Pantalla 2:** Lista de tareas, lista las tareas actuales y proporciona acceso a las acciones disponibles dependiendo del rol

## Apéndice C: Diagrama de Clases



## Apéndice D: Archivo WSDL

```
<definitions name="BPELWorkList"
  targetNamespace="http://xmlns.oracle.com/BPELWorkList"
  xmlns=http://schemas.xmlsoap.org/wsdl/
xmlns:ns1="WorklistXSD"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-
link/"
  xmlns:client="http://xmlns.oracle.com/BPELWorkList"
  >
  <types>
    <schema attributeFormDefault="qualified"
elementFormDefault="qualified"
  targetNamespace="http://xmlns.oracle.com/BPELWorkList"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ns1="WorklistXSD"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-
link/"
  xmlns:client="http://xmlns.oracle.com/BPELWorkList">
      <element name="BPELWorkListProcessRequest">
        <complexType>
          <sequence>
            <element name="input" type="string"/>
          </sequence>
        </complexType>
      </element>
      <element name="BPELWorkListProcessResponse">
        <complexType>
          <sequence>
            <element name="result" type="string"/>
          </sequence>
        </complexType>
      </element>
    </schema>
    <schema xmlns=http://www.w3.org/2001/XMLSchema
xmlns:ns1="WorklistXSD"
  xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/part
ner-link/"
  xmlns:client="http://xmlns.oracle.com/BPELWorkList">
      <import namespace="WorklistXSD"
schemaLocation="WorklistXSD.xsd"/>
    </schema>
  </types>
  <message name="BPELWorkListRequestMessage">
    <part name="payload" element="ns1:WorkListXSD_Request"/>
  </message>
```

```
<message name="BPELWorkListResponseMessage">
  <part name="payload" element="ns1:WorkListXSD_Response"/>
</message>
<portType name="BPELWorkList">
  <operation name="process">
    <input message="client:BPELWorkListRequestMessage"/>
    <output message="client:BPELWorkListResponseMessage"/>
  </operation>
</portType>
<plnk:partnerLinkType name="BPELWorkList">
  <plnk:role name="BPELWorkListProvider">
    <plnk:portType name="client:BPELWorkList"/>
  </plnk:role>
</plnk:partnerLinkType>
</definitions>
```

## Apéndice E: Archivo XSD WorkListXSD.XSD

```
<?xml version="1.0" encoding="windows-1252"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://www.example.org"
            targetNamespace="WorklistXSD"
            elementFormDefault="qualified" >

  <xsd:element name="WorkListXSD_Request">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="UserName" type="xsd:string"/>
        <xsd:element name="Command" type="xsd:string"/>
        <xsd:element name="TaskID" type="xsd:string"/>
        <xsd:element name="DurationDays" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="WorkListXSD_Response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="UserName" type="xsd:string"/>
        <xsd:element name="UserTitle" type="xsd:string"/>
        <xsd:element name="UserRoles" type="xsd:string"/>
        <xsd:element name="Errors" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="WorkListXSD_Task">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="TaskID" type="xsd:string"/>
        <xsd:element name="Title" type="xsd:string"/>
        <xsd:element name="Priority" type="xsd:string"/>
        <xsd:element name="Status" type="xsd:string"/>
        <xsd:element name="Assignee" type="xsd:string"/>
        <xsd:element name="ExpirationDate" type="xsd:string"/>
        <xsd:element name="ModifiedDate" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

## **12. Diseño Interfaces NTier**

---

### **12.1 DESCRIPCIÓN DEL DISEÑO**

Estas interfaces son una colección de *web services* BPEL síncronos los cuales se comunican con un *web service* .NET para invocar las funciones de agregar y actualizar información de Proveedores de la capa de negocios NTier dentro del sistema de Medicaid. La capa de negocios NTier consiste básicamente en una representación de modelo de objetos de la información de proveedores y el *web service* relacionado .NET proporciona métodos para la construcción de estos objetos de proveedor, la mutación de los campos de datos de proveedores y acceso a cualquier información relevante proveniente de *web services* externos (en este caso, el flujo BPEL). Otra cosa que la capa NTier lleva a cabo además del control de la representación como modelo de objetos de alto nivel es la persistencia de los datos de proveedores en la base de datos correspondiente. Así, todos los módulos de *web services* BPEL dentro del sistema de Enrolamiento de Proveedores utilizarían una interfaz NTier si necesitan comunicarse con la base de datos de enrolamiento de proveedores.

Algunos ejemplo de los flujos BPEL que utilizan estas interfaces son los *web services* Provider Enrollment Main (11.6), Add Service Location (11.8), y Display Provider Data (11.10). Abajo se encuentra una descripción detallada de cómo cada uno de estos servicios interactúa con el modelo de objetos NTier. Aunque es posible que cada uno de estos servicios se comunique con el *web service* .NET directamente, la adición de estas interfaces NTier reduce el acoplamiento con la capa de negocios NTier y, en adición, aísla las capas de cambios potenciales en las capas NTier o del flujo BPEL. De manera similar, los servicios BPEL pueden utilizar la base de datos directamente usando TopLink, pero esta *API* ha probado ser ineficiente y propensa a errores para esta aplicación. En conjunto, esta estrategia genera una arquitectura que es más robusta y modularizada que lo que sería con los diseños alternativos. En el Apéndice C tenemos una representación gráfica del diseño.

#### **12.1.1 Interfaz NTier Provider Enrollment Main**

El servicio Provider Enrollment Main invoca una interfaz NTier para generar y añadir un proveedor en el sistema de Medicaid. La interfaz se comunica con la representación del modelo de objetos NTier a través del *web service* .NET para instanciar el correspondiente objeto de proveedores y persistir la información en la base de datos.

#### **12.2.1 Flujo de Trabajo Humano**

No se requieren elementos de trabajo humano; el proceso es completamente síncrono.

### **12.2.2 Manejo de Excepciones**

Al menos, el *web service* debe manejar la incapacidad para conectar con el servicio .NET. Debe también verificar los errores en la entrada de datos desde la forma *web* para no realizar invocaciones innecesarias.

### **12.2.3 Esquemas de Tiempo**

No se han establecido esquemas de Tiempo

### **12.2.4 Manejadores para Compensación**

Los datos no son persistidos directamente por este *web service*, aunque este proceso no invoca el servicio .NET que lleva a cabo esta tarea, de manera que los manejadores de compensación no son necesarios.

## **12.3 Add Service Location NTier Interface**

El servicio para Agregar una Localidad de Servicio utiliza una interfaz NTier para actualizar la información de un proveedor que ya existe en el sistema. La interfaz se comunica con el servicio web .NET para localizar al proveedor dentro de la base de datos y anexar una nueva localidad de servicio.

### **12.3.2 Flujo de Trabajo Humano**

Los elementos de flujo de trabajo humano no son requeridos, el proceso es completamente síncrono.

### **12.3.3 Manejo de Excepciones**

Al menos, el *web service* debe manejar la inhabilidad para comunicarse con el servicio .NET. La interfaz debe también cubrir errores en la entrada de datos de la página web para no hacer una llamada innecesaria. El proceso debe manejar el caso donde el proveedor no está en el sistema, o donde la información del proveedor no puede ser localizada.

### **12.3.4 Esquemas de Tiempo**

No se han establecido esquemas de Tiempo

### **12.3.5 Manejadores de Compensación**

No se persisten datos con este *web service*, aunque el proceso invoca un servicio .NET el cual lleva a cabo esta tarea, de manera que los manejadores de compensación no son requeridos.

## **12.4 XSD**

Cada interfaz NTier tendrá un XSD similar al XSD requerido por su correspondiente *web service* BPEL. Esto tiene sentido dado que las interfaces NTier deben capturar la misma información que se obtiene por el proceso BPEL y propagarla hacia el *web service* .NET para la eventual sincronización con la base de datos.

## **12.5 SOBRECARGA DE RENDIMIENTO/SISTEMA**

En este momento se desconoce la capacidad del servidor para manejar un volumen alto de peticiones concurrentes por parte del cliente. Se requieren pruebas de estrés para determinar el rendimiento del sistema.

## **12.6 DEPENDENCIAS**

### **12.6.1 Web Services BPEL Auxiliares**

Este modulo requiere la conducta correcta de los siguientes *web services* para su compleción correcta.

### **12.6.2 Web Service Enrolamiento NTier**

Un *web service* basado en .NET que actua como una interfaz para persistir eventualmente la información del proveedor en la base de datos de enrolamiento e instancia un proveedor en la representación del modelo de objetos. La NTierInterface debe ser capaz de comunicarse satisfactoriamente con este servicio para poder agregar la petición del proveedor para enrolamiento dentro de la base de datos.

### **12.6.3 Provider Enrollment**

Permite a los proveedores enrolarse como proveedores de Medicaid y les proporciona un número de guía de su aplicación. La NTierInterface depende del servicio de enrolamiento de proveedores para recolectar satisfactoriamente y pasar la información de proveedores de manera que el registro se agregue a la base de datos de rastreo de enrolamiento. Para más información verificar el documento de diseño de Enrolamiento de Proveedores.

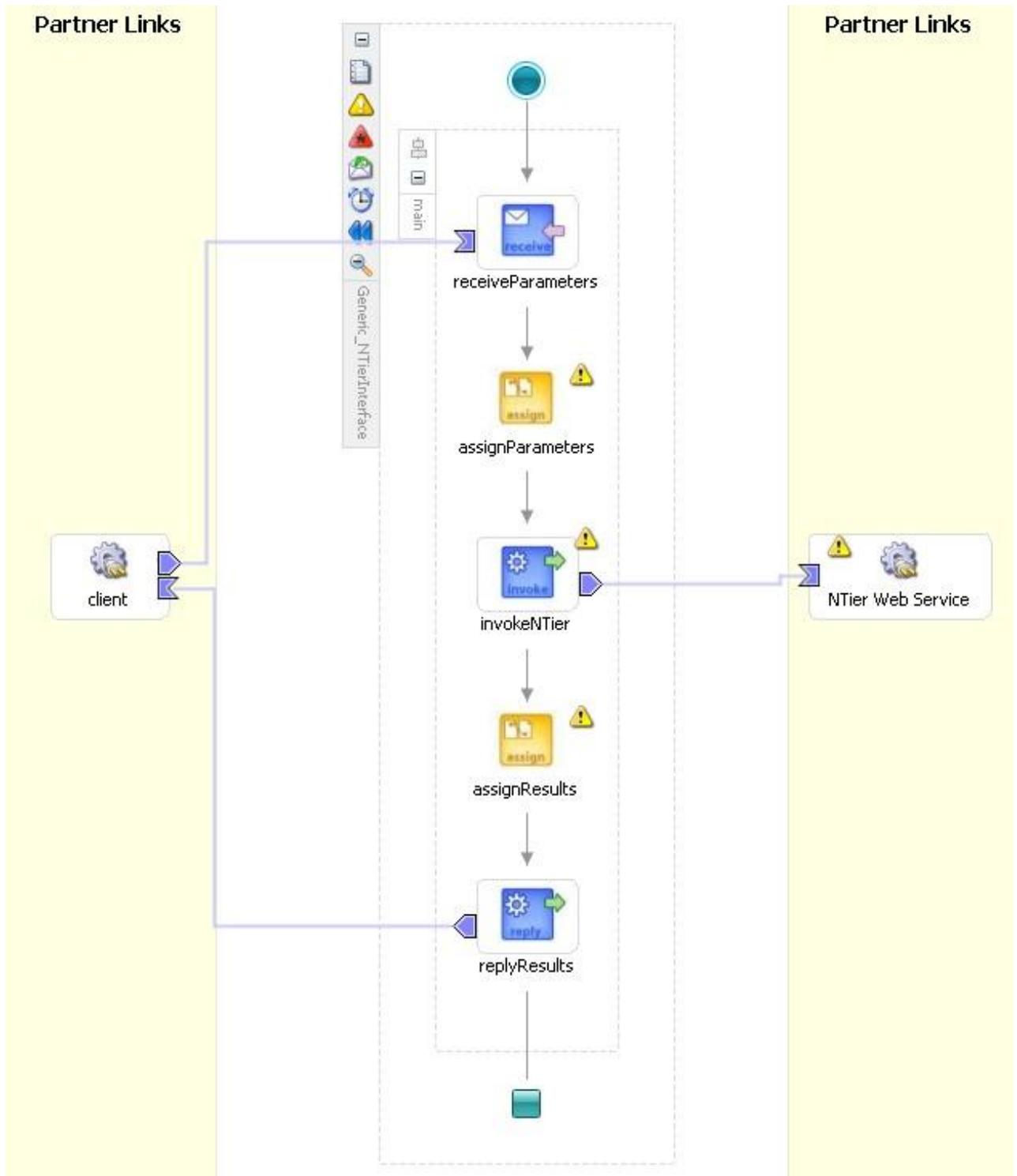
### **12.6.4 Add Service Location**

Permite a un proveedor previamente registrado agregar una localidad adicional de servicio a su perfil Medicaid.

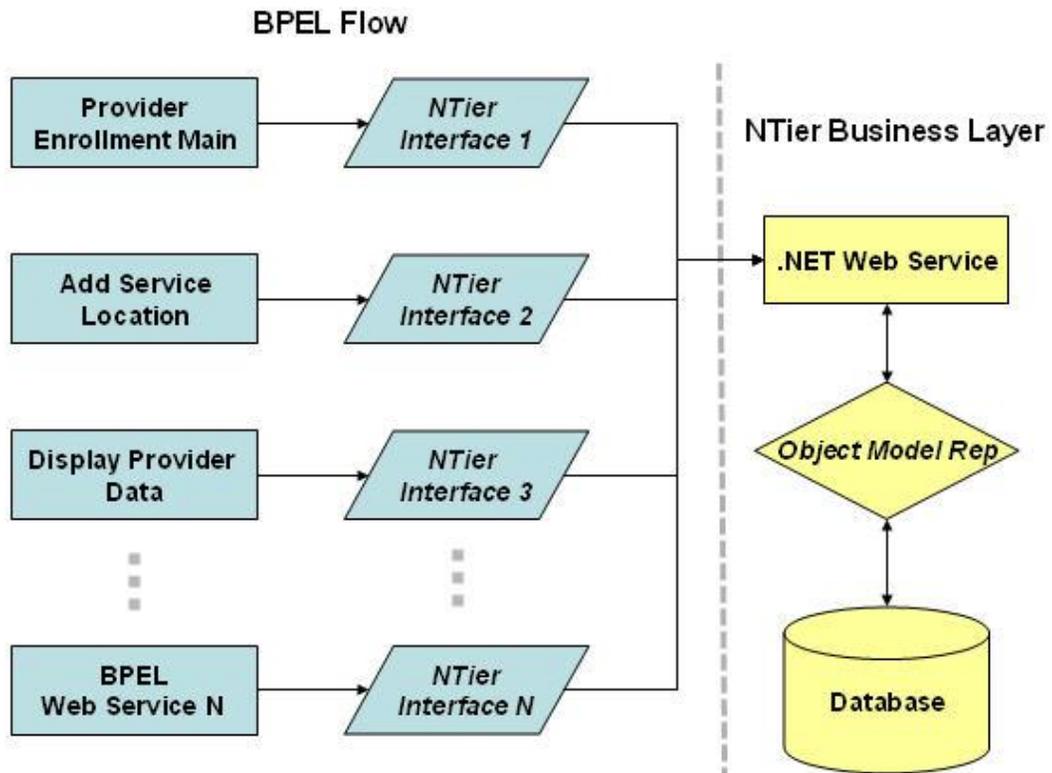
### **12.6.5 Display Provider Data**

Permite al usuario verificar la información de su perfil Medicaid.

## Apéndice A: Interfaz NTier Genérica, Flujo BPEL



## Apéndice B: Representación Gráfica del Diseño NTier



## Anexo A. Descripción de las Tablas

### T\_BPEL\_RSP\_TXT

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_RESP_TXT	PK	NUMBER(9)	N	
DSC_RESP		VARCHAR2(200)	N	
DSC_TEXT		VARCHAR2(4000)	N	

### T\_BPEL\_TITLE

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_TITLE	PK	NUMBER(9)	N	
CDE_TITLE		VARCHAR2(20)	N	
DSC_TITLE		VARCHAR2(250)	N	

### T\_BPEL\_TRACKING

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_TRACKING	PK	NUMBER(9)	N	
ID_PROVIDER	FK	VARCHAR2(15)	N	
CDE_STATUS		CHAR(1)	N	'P'
CDE_WEBSERVICE		VARCHAR2(20)	N	
CDE_CURRENT_WS		VARCHAR2(20)	N	
DTE_INSERT		NUMBER(8)	N	
DTE_LAST_UPDATED		NUMBER(8)	N	0
SAK_LETTER		NUMBER(9)	N	-1

### T\_BPEL\_WEBSERVICE

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_WEBSERVICE	PK	NUMBER(9)	N	
CDE_WEBSERVICE		VARCHAR2(20)	N	
DSC_WEBSERVICE		VARCHAR2(250)	N	

### T\_BPEL\_WKFL\_PARAMS

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_WEBSERVICE	PK	NUMBER(9)	N	
SAK_WORKFLOW	PK	NUMBER(9)	N	
CDE_ASSIGNEE	PK	VARCHAR2(20)	N	
DTE_START	PK	NUMBER(8)	N	
DTE_END	PK	NUMBER(8)	N	
IND_WKFL_ASG_TYPE		CHAR(1)	N	
CDE_TIME_PERIOD		CHAR(1)	N	
NUM_TIME_VALUE		NUMBER(3)	N	
NUM_ESC_RENEW		NUMBER(2)	Y	
SAK_TITLE		NUMBER(9)	N	
ID_WKFL_ASG		VARCHAR2(20)	N	

### T\_BPEL\_WORKFLOW

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_WORKFLOW	PK	NUMBER(9)	N	
CDE_WORKFLOW		VARCHAR2(20)	N	
DSC_WORKFLOW		VARCHAR2(250)	N	

## T\_BPEL\_WS\_RESPONSE

Columna	Llave	Tipo de Dato	Nulo	Valor
SAK_WEBSERVICE	PK	NUMBER(9)	N	
CDE_RESULT	PK	CHAR(1)	N	
SAK_WORKFLOW	PK	NUMBER(9)	N	
CDE_ACTIVITY_MIN	PK	VARCHAR2(6)	N	
CDE_ACTIVITY_MAX	PK	VARCHAR2(6)	N	
DTE_START	PK	NUMBER(8)	N	
DTE_END	PK	NUMBER(8)	N	
SAK_RESP_TXT		NUMBER(9)	N	
SAK_LETTER		NUMBER(9)	N	

## Anexo B. Glosario de Términos

---

BPEL. Acrónimo de *Business Process Execution Language*, lenguaje de ejecución de procesos de negocios. Lenguaje estándar de la industria para la orquestación de procesos, permite la interconexión de *Web Services* discretos y aplicaciones SOA dentro de un flujo de proceso de principio a fin.

MIDDLEWARE. Agentes de Software que actúan como intermediarios entre diferentes componentes de una aplicación. Se utilizan normalmente para dar soporte a aplicaciones distribuidas complejas. Los agentes de software involucrados pueden ser uno o varios. Es una capa de software que se sitúa entre el sistema operativo y las aplicaciones en cada punto del sistema.

Orquestación. La orquestación describe la organización automatizada, coordinación y administración de sistemas computacionales complejos, *middleware* y servicios.

SOA. Acrónimo de *Service Oriented Architecture*, Arquitectura Orientada a Servicios. Concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requerimientos de software del usuario. Las SOAs están formadas por servicios de aplicación débilmente acoplados y altamente interoperables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación. La definición de la interfaz encapsula (oculta) las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo (como Java o .NET). Con esta arquitectura, se pretende que los componentes de software desarrollados sean altamente reutilizables, ya que la interfaz se define siguiendo un estándar; así, un servicio C Sharp podría ser usado por una aplicación Java.

SOAP. Protocolo para el intercambio de mensajes basados en XML sobre una red de computadoras, normalmente utilizando HTTP. SOAP forma la capa fundamental en el esquema de *web services*, proporcionando un marco de trabajo simple para mensajes sobre la cual otras capas más abstractas pueden construirse. SOAP facilita el patrón arquitectónico de *Service-Oriented*. Existen muchos patrones diferentes en SOAP, pero el más común es el patrón RPC (*Remote Procedure Call*, Llamada a Procedimientos Remotos), donde un nodo de red (el cliente) envía un mensaje de petición a otro nodo (el servidor), y el servidor envía un mensaje de respuesta inmediatamente al cliente.

W3C. World Wide Web Consortium. Es un consorcio internacional donde las organizaciones miembros, un *staff* de tiempo completo y el público, trabajan juntos para desarrollar los estándares para el World Wide Web. El W3C tiene el máximo potencial al desarrollar protocolos y guías para asegurar el

Web Service. Sistema de software diseñado para soportar la interoperabilidad de la interacción máquina-a-máquina sobre una red. Tiene una interfaz que está descrita en un formato procesable por una máquina como WSDL. Otros sistemas interactúan con el *web service* de la manera predefinida por su interfaz utilizando mensajes, los cuales se pueden adjuntar en un sobre SOAP. Estos mensajes son transportados típicamente utilizando HTTP, y comprenden normalmente XML en conjunto con otros estándares relacionados con Web. Las aplicaciones de software escritas en diferentes lenguajes de programación y ejecutándose en diversas plataformas pueden utilizar *web services* para intercambiar datos sobre redes de computadoras como el Internet de manera similar a la comunicación entre procesos en una computadora. Esta interoperabilidad (por ejemplo, entre Java y Python, o Microsoft Windows y aplicaciones Linux) se logra gracias al uso de estándares abiertos. OASIS y el W3C son los comités principales responsables de la arquitectura y estandarización de los *web services*. Para mejorar la interoperabilidad entre implementaciones de *web services*, la

organización WS-I ha desarrollado una serie de perfiles para definir más adecuadamente los estándares involucrados.

WSDL. Acrónimo de *Web Services Description Language*, Lenguaje de Descripción de Servicios Web. Es un formato XML publicado para describir *Web Services*. Describe la interfaz pública de un *web service*. Esta es una descripción de servicio basada en XML de como comunicarse utilizando un *web service*; particularmente, las ligas de los protocolos y los formatos de mensaje requeridos para interactuar con los *web services* listados en su directorio. Las operaciones y mensajes soportados se describen de manera abstracta, y después se asocian a un protocolo concreto de red y un formato de mensaje. WSDL se utiliza normalmente en combinación con SOAP y XSD para proporcionar *web services* en el Internet. Un programa cliente que se conecta a un *web service* puede leer el WSDL para determinar qué funciones están disponibles en el servidor. Cualesquiera tipos de datos especiales utilizados se incrustan en el archivo WSDL en la forma de un esquema XML (o XSD). El cliente puede usar SOAP para llamar alguna de las funciones listadas en el WSDL.

XSD. Un XSD define un tipo de documento XML en términos de las restricciones en las cuales los elementos y atributos pueden caer, su relación con otros, qué tipos de datos pueden contener, y otras cosas. Puede ser utilizado con software de validación para cerciorarse de que un documento XML es de este tipo. Los XSDs reemplazan a los DTDs.

# Anexo C. Prototipos de las Interfaces Gráficas

Application for Provider Enrollment - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail Stop Links

Address <http://usplddssc005.edsdmdssc.dssc.edcs.com/ProvEnrollWiz/ProvInfo.aspx> Go Links

**Inter Change**  
Governmental Health Portfolio

### Applicant

As a condition for entering into or renewing a provider agreement all applicants must complete an application. A true, accurate and complete disclosure of all requested information is required by the Federal and State regulations that govern the Medical Assistance Program. Failure of an applicant to submit the requested information or the submission of inaccurate or incomplete information may result in refusal by the Medical Assistance program to enter into, renew or continue a provider agreement with the applicant. Furthermore, the applicant is required by Federal and State regulations to update the information submitted on the application.

\* denotes a required field

Applicant Type			
Name Type*	personal		
First name*	<input type="text"/>	Middle name	<input type="text"/>
Last name*	<input type="text"/>		
Organization name	<input type="text"/>		
Provider ID Type*	tax	Provider ID*	<input type="text"/>
Gender*	<input checked="" type="radio"/> Male <input type="radio"/> Female	UPIN*	<input type="text"/>

Provider Address			
Street*	<input type="text"/>		
City*	<input type="text"/>	State*	AK
Zip code*	<input type="text"/>	Zip Extension	<input type="text"/>
Phone*	<input type="text"/>	Extension	<input type="text"/>
Fax	<input type="text"/>		

Application for Provider Enrollment - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Address http://usplddssc005.edsdmdssc.dssc.eds.com/ProvEnrollWiz/ProvInfo.aspx

Organization name

Provider ID Type\* tax  Provider ID\*

Gender\*  Male  Female UPIN\*

**Provider Address**

Street\*

City\*  State\* AK

Zip code\*  Zip Extension

Phone\*  Extension

Fax

**Provider Contact Information**

Title\*

First name\*  Middle name

Last name\*

Phone\*  Extension

**Provider License**

License Number\*  License Type prescriber

State\* AK

Certification Date\*  Certification End Date\*

Next ->

Done Local intranet

Applicant Contact Information - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail Stop Links

Address http://usplddssc005.edsdmsdssc.dssc.eds.com/ProvEnrollWiz/ProvEnrollAppInfo.aspx

**Inter Change**  
Government Health Portfolio

\* denotes a required field

**Applicant Information**

**Application Source\*** web

**Response Media\*** email

**Email Address\***

Phone Number 1234567890

Phone Extension

Fax Number

Next ->

Done Local intranet

Additional Information - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Address <http://usplddssc005.edsdmdssc.dssc.eds.com/ProvEnrollWiz/ProvAddInfo.aspx> Go Links



### Additional Information

\* denotes a required field

#### Account Information

Routing Number	<input type="text"/>
Account Number	<input type="text"/>
Bank Account Type	None <input type="button" value="v"/>

#### Group Membership

Provider ID Type*	tax <input type="button" value="v"/>	Tax ID*	123456789 <input type="text"/>
Name Type	None <input type="button" value="v"/>		
First name	<input type="text"/>	Middle name	<input type="text"/>
Last name	<input type="text"/>		
Organization name	<input type="text"/>		
Effective Date	<input type="text"/>		
End Date	<input type="text"/>		

#### Certification

Certification Title Type	<input type="text"/>		
Start Date	<input type="text"/>	End Date	<input type="text"/>

Next ->

Add Service Location - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Print Mail News RSS

Address http://usplddssc005.edsdmdssc.dssc.edcs.com/ProvEnrollWiz/AddServiceLocation.aspx Go Links

**Inter Change**  
Government Health Portfolio

\* denotes a required field

**Provider Name**

Name type personal

First Name\* a Middle Name

Last Name\* b

Provider ID Type tax Provider ID \* 123456789

**Service Location**

County Adair

Address Type homeOffice  Remaining addresses identical?

Street\*

City \* State AK

Zip code\* Zip Extension

Phone\* Extension

Fax Add Another Address

**IRS W9 Info**

Tax ID Type SSN Name\*

Street\*

Done Local intranet

Add Service Location - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Address <http://usplddssc005.edsdmdssc.dssc.edc.com/ProvEnrollWiz/AddServiceLocation.aspx> Go Links

### IRS W9 Info

Tax ID Type	SSN	Name*	
Street*			
City*		State	AK
Zip Code*		Zip Extension	
Date Effective*		Date End	
Phone Number*		Phone Extension	
IND Tax Exempt	<input type="radio"/> Yes <input checked="" type="radio"/> No		
IND Form 147	<input type="radio"/> Yes <input checked="" type="radio"/> No		
IND W9 Form	<input type="radio"/> Yes <input checked="" type="radio"/> No		
Provider Type	Physician	Provider License*	
Provider Specialty*	318		
Language Spoken*	None		
Language Start Date*			
Language Stop Date*		Add Another Language	
Billing	<input type="radio"/> Yes <input checked="" type="radio"/> No		
Handicap Access	<input type="radio"/> Yes <input checked="" type="radio"/> No		

Add Service Location - Microsoft Internet Explorer provided by EDS COE

File Edit View Favorites Tools Help

Address http://usplddssc005.edsdmdssc.dssc.eds.com/ProvEnrollWiz/AddServiceLocation.aspx

IND Tax Exempt	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
IND Form 147	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
IND W9 Form	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
Provider Type	<input type="text" value="Physician"/>		Provider License*	<input type="text"/>
Provider Specialty*	<input type="text" value="318"/>			
Language Spoken*	<input type="text" value="None"/>			
Language Start Date*	<input type="text"/>			
Language Stop Date*	<input type="text"/>		<input type="button" value="Add Another Language"/>	
Billing	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
Handicap Access	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
Public / Private	<input type="radio"/> Private	<input checked="" type="radio"/> Public		
Withhold FICA	<input type="radio"/> Yes	<input checked="" type="radio"/> No		
HIPAA837 Capable	<input type="radio"/> Yes	<input checked="" type="radio"/> No		<input type="button" value="Add Another Service Location"/>

Done Local intranet

## **Anexo D. Referencias Web**

---

### ***Información***

EDS

<http://www.eds.com>

EDS Agility Alliance

<http://www.eds.com/services/alliances/agility/>

HIPAA

<http://www.hhs.gov/ocr/hipaa/>

CMS

[http://www.cms.hhs.gov/hipaageninfo/01\\_overview.asp](http://www.cms.hhs.gov/hipaageninfo/01_overview.asp)

Business Process Management Initiative

<http://www.bpmi.org/>

### ***Software de Oracle***

Oracle BPEL Process Manager

[http://www.oracle.com/appserver/bpel\\_home.html](http://www.oracle.com/appserver/bpel_home.html)

Información Técnica del Oracle BPEL Process Manager

<http://www.oracle.com/technology/products/ias/bpel/index.html>

Servidor de Aplicaciones Oracle

<http://www.oracle.com/appserver/index.html>

Base de Datos Oracle 10g

<http://www.oracle.com/database/index.html>

Oracle JDeveloper

<http://www.oracle.com/technology/products/jdev/101/tutorials/index.htm>

Oracle SQL Developer

[http://www.oracle.com/technology/products/database/sql\\_developer/index.html](http://www.oracle.com/technology/products/database/sql_developer/index.html)

## ***Software Libre***

WinMerge

<http://winmerge.sourceforge.net/>

NUnit

<http://www.nunit.org/>

Emacs

<http://www.gnu.org/software/emacs/>

JEdit

<http://www.jedit.org>

Framework de Aplicaciones Web Dot Net Nuke

<http://www.dotnetnuke.com/>

Vault de Sourcegear

<http://www.sourcegear.com/vault/>

## ***Software de Sun***

Solaris

<http://www.sun.com/software/solaris/>

Plataforma Java, Edición Enterprise

<http://java.sun.com/javaee/index.jsp>

## ***Software de Microsoft***

Microsoft Visual Studio.NET 2003

<http://msdn.microsoft.com/vstudio/>

Microsoft SQL Server 2000

<http://msdn.microsoft.com/sql/>

Microsoft .NET Framework 1.1

<http://msdn.microsoft.com/netframework/>

Microsoft C#

<http://msdn.microsoft.com/vcsharp/>