

Arquitectura de Integración

INFORMACIÓN DEL DOCUMENTO

Información General	Nombre del Documento	Arquitectura de integración
	Ubicación	
	Documentos Relacionados	

Revisión	Rol	Nombre	Área	Firma	Fecha

Aprobación	Rol	Nombre	Área	Firma	Fecha

Historia del Documento

Versión	Fecha	Nombre	Razón del Cambio
1.0	30 jul. 13	Eduardo Rodriguez	Creación del documento

TABLA DE CONTENIDO

INFORMACIÓN DEL DOCUMENTO	1
TABLA DE CONTENIDO	2
1 INTRODUCCIÓN	3
1.1 PROPÓSITO	4
1.2 AUDIENCIA	4
1.3 GLOSARIO DE TÉRMINOS	4
1.4 ALCANCE	5
2 REPRESENTACIÓN ARQUITECTÓNICA	6
2.1 FLUJOS DE INTEGRACIÓN	6
2.1.1 Adaptadores.....	6
2.1.2 Seguridad.....	6
2.1.3 Manejo de errores	8
2.2 VISTA LÓGICA	10
2.2.1 Diagrama de Contexto	10
2.2.2 Diagrama de Flujo de datos.....	11
2.2.3 Diagrama BPMN.....	12
2.2.4 Diagrama de Secuencia.....	12
2.3 VISTA FÍSICA	13
2.3.1 Diagrama de Infraestructura.....	13
2.3.2 Diagrama de Sistema	14
3 DISEÑO.....	15
3.1 ERRORES Y EXCEPCIONES.	15
3.2 MAPEO DATOS	16
3.2.1 Servicio OrNOCBuscaIncidencia.....	16
3.2.2 Mapeo en Siebel	17
3.3 WSDL Y XSD	20
3.3.1 Servicio OrNOCBuscaIncidencia.....	20
4 ESTÁNDARES	21
4.1 ESTÁNDARES DE TECNOLOGÍA.....	21
4.2 ESTÁNDARES DE CONSTRUCCIÓN	21
4.2.1 Espacio de nombres.....	22
4.2.2 Lineamientos para la orquestación de los flujos BPEL	22
4.2.3 Reglas de nombrado para flujos BPEL	23
4.2.4 Estándares de documentación	27

1 Introducción

La Arquitectura Orientada a Servicios (SOA) es un estilo arquitectónico basado en estándares que facilita la integración de los procesos de negocio, se puede definir como un marco conceptual para la orquestación de procesos de negocio con soporte en la tecnología, a través de componentes con características estándares de tipo general y específicos de cada industria, las cuales son:

- Reutilización
- Granularidad
- Modularidad
- Composición
- Portabilidad
- Interoperabilidad

Se debe guiar por los siguientes principios:

- **Acoplamiento ligero del Servicio**
Los Servicios deben mantener una relación que reduzca al mínimo las dependencias entre estos.
- **Abstracción del Servicio**
La lógica del Servicio está en el Servicio mismo. El Servicio se invoca con base a las especificaciones del contrato.
- **Reutilización del Servicio**
La lógica de negocio se debe dividir en Servicios reutilizables.
- **Composición del Servicio**
Las colecciones de Servicios pueden ser coordinadas/orquestadas e incluidas en Servicios de tipo compuesto a fin de servir procesos de negocio complejos.
- **Autonomía del Servicio**
Los Servicios tienen control sobre la lógica que encapsulan.
- **Optimización del Servicio**
Los Servicios se optimizan a fin de lograr el QoS (la calidad de Servicio) y SLA aceptables al negocio.
- **Servicios Descubribles**
Los Servicios deben ser diseñados de manera que su definición y metadatos describan de forma clara su función, para que los consumidores los puedan descubrir a través de encuestas en el registro de Servicios.
- **Relevancia del Servicio**
La funcionalidad se debe llevar a un nivel granular reconocido por el consumidor como Servicio significativo.

1.1 Propósito

El presente documento de arquitectura define los elementos y componentes de integración entre los sistemas de la plataforma tecnológica del Proyecto Dr. Cable. Describe la tecnología, los mecanismos y decisiones de diseño utilizados, y establece los lineamientos de diseño y construcción; así como definir la implementación de la seguridad de los servicios web.

1.2 Audiencia

Este documento está dirigido a las personas involucradas en el proyecto Dr. Cable que conforman los siguientes equipos:

- TI Noc Cablevision
- Arquitectura Cablevision

1.3 Glosario de términos

Término	Definición
Auspice	Sistema Administrador de Servicios de voz
BPEL	<i>Business ProcessExecutionLanguage</i> , es un lenguaje de ejecución de procesos de negocio con Servicios Web
Cliente	Persona u organización que adquiere productos ofrecidos por la unidad de negocio
CRM	CustomerRelationship Management, por sus siglas en inglés. Sistema que permite la administración de las transacciones del Cliente
Fusion	Oracle Fusion Middleware constituye una suite de aplicaciones que abarca la administración de procesos, infraestructura de aplicaciones y herramientas de desarrolladores
Middleware	Aplicación que permite la integración entre diferentes tecnologías.
OSB	Oracle Service Bus, por sus siglas en inglés. Herramienta que nos permite enrutar y/o virtualizar los servicios Web
Siebel	Oracle Siebel es una aplicación CRM que permite gestionar la relación con el Cliente de manera integral
Usuario	Usuario final que tiene acceso a los sistemas CRM para gestionar la operación del negocio
Remedy	Sistema de gestión de servicio de TI

1.4 Alcance

El alcance del documento corresponde únicamente a la capa de integración de sistemas del proyecto Dr. Cable, en este contexto, se encuentran desarrollados los siguientes puntos de integración de sistemas:

- Diseño de la integración de sistemas, a través de la orquestación de procesos de negocio utilizando BPEL, de la plataforma Oracle SOA Suite
- Lineamientos de diseño para los documentos de especificaciones técnicas de los flujos de integración
- Lineamientos de construcción para la plataforma Oracle SOA Suite

Los sistemas considerados dentro del alcance de la solución de integración, son los siguientes:

- SIEBEL
- REMEDY
- AUSPI 

2 Representación Arquitectónica

2.1 Flujos de integración

Un Flujo de Integración es un proceso BPEL a través del cual son orquestados los servicios expuestos en los diferentes sistemas, incluidos en la plataforma tecnológica del proyecto Dr. Cable, para dar cumplimiento a la ejecución de los procesos de negocio que involucran el intercambio de información entre dichos sistemas.

El flujo de integración es también un servicio compuesto, pues a su vez involucra el llamado de otros servicios. La identificación de los servicios candidatos en el flujo de integración deriva de los requerimientos funcionales.

Finalmente, un flujo de integración representa la unidad de software mínima desplegada en un servidor de aplicaciones que estará disponible, como Servicio Web de negocio, a los demás sistemas del proyecto Dr. Cable.

2.1.1 Adaptadores

Oracle SOA Suite incluye un conjunto de adaptadores que habilita la integración con los Servicios de sistemas de diferentes tecnologías, para poder consumir la funcionalidad de dichos Servicios desde BPEL.

Los adaptadores son componentes de software reutilizables y genéricos que permiten consumir, desde un flujo BPEL, los Servicios expuestos en diferentes tecnologías como si fueran Servicios Web, esto es, mediante la manipulación de documentos XML validados y bien formados, con esquemas XSD que definen todos los tipos de datos para los parámetros de entrada y salida.

Para la arquitectura de integración de la plataforma tecnológica del proyecto Dr. Cable son requeridos los siguientes adaptadores:

Adaptador de Oracle	Sistemas proveedores de los servicios
Servicio Web	REMEDY
Data Base	AUSPICE


Adaptador de Servicio Web

Permite el consumo de otros Servicios Web.

Adaptador Data Base

Un adaptador de datos contiene comandos SQL para la manipulación de información en una DB.

2.1.2 Seguridad

Los servicios web oportan un mecanismo de seguridad (WS-Security) UserNameToken, el cual permite el envío y recepción de credenciales del usuario de una manera estándar y flexible, donde la autenticación consiste en validar el usuario y contraseña.

El soporte para el mecanismo UserNameToken, incluye lo siguiente:

- Permite a una solicitud de entrada SOAP contener credenciales de usuario que puedan ser proporcionadas a nivel de header en el mensaje SOAP y realizar la autenticación necesaria
- Permite a una solicitud de salida SOAP contener credenciales de usuario que puedan ser utilizadas por una aplicación externa

Lo siguiente es un ejemplo de pasar el nombre del usuario y contraseña por URL el cual **no debe utilizarse** por que se revelan las credenciales del usuario:

`http://webserver/eai_enu/start.swe?SWEEExtSource=WebService&SWEEExtCmd=Execute&Username=SADMIN&Password=SADMIN`

Con UserNametokens, la URL no revela las credenciales de usuario:

`http://webserver/eai_anon_enu/start.swe?SWEEExtSource=SecureWebService&SWEEExtCmd=Execute`

2.1.3 Manejo de errores

El manejo de errores incluye todas las excepciones que se generan dentro de cada proceso de orquestación y haciendo uso de los manejos de errores genéricos y específicos de sistema, descritos de la siguiente manera:

- **Servicios sin respuesta de la aplicación destino**

En este tipo de excepción contaremos con un posible disparador de la misma quedará definido que siempre que se solicite un Servicio se debe verificar que la aplicación tenga una respuesta después de tres reintentos o en caso contrario se manejará la excepción como Servicio sin Respuesta.

- **Excepción de origen desconocido**

Cuando se conoce que los datos son válidos, que los servicios funcionan correctamente y que no existe ningún error en el código de integración, se considera que es una excepción de tipo desconocida. Puede darse por un segmento de red dañado, un error en los sistemas operativos que soportan a las aplicaciones, etc.

- **No recibir datos**

Si la comunicación es correcta pero se reciben datos vacíos o un dato nulo se emitirá una excepción de tipo Datos Vacíos o Nulos.

- **Errores específicos de sistema**

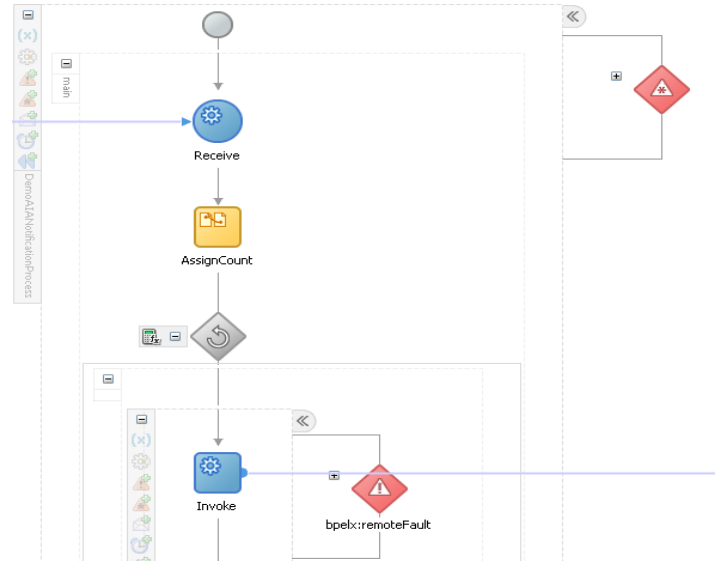
- *Timeout* ocurre cuando un servicio web síncrono al que invoca BPEL tarda demasiado en responder
- *RemoteFault* ocurre cuando un servicio al que intenta invocar BPEL no puede ser alcanzado
- *BindingFault* ocurre cuando existe una incongruencia entre el servicio proveedor y el servicio consumidor. Este error también puede ser ocasionado por una invocación no autorizada del servicio

Por otro lado es conveniente especificar que no se puede administrar un rollback del rollback, es decir si existe un error dentro del proceso de orquestación y este es atendido por medio de un rollback, se realizan las actividades para deshacer la ejecución del proceso inicial, si una de estas falla en el intento, ya se tendrá que verificar las pistas de auditoria en el Enterprise Manager (EM) y realizar la operación de manera manual de acuerdo al proceso de negocio.

Oracle SOA Suite cuenta con un mecanismo para el manejo de errores (Fault Management Framework) el cual permite el manejo de mensaje de error o excepciones dentro de los procesos de orquestación.

En el proyecto Dr. Cable, el manejo de errores se realiza mediante dicho mecanismo, en dos formas:

- **Manejo de excepción en el proceso.** Utiliza actividades de *Catch* y *Catch All* dentro del flujo de integración para la captura de errores en tiempo de ejecución u otras excepciones



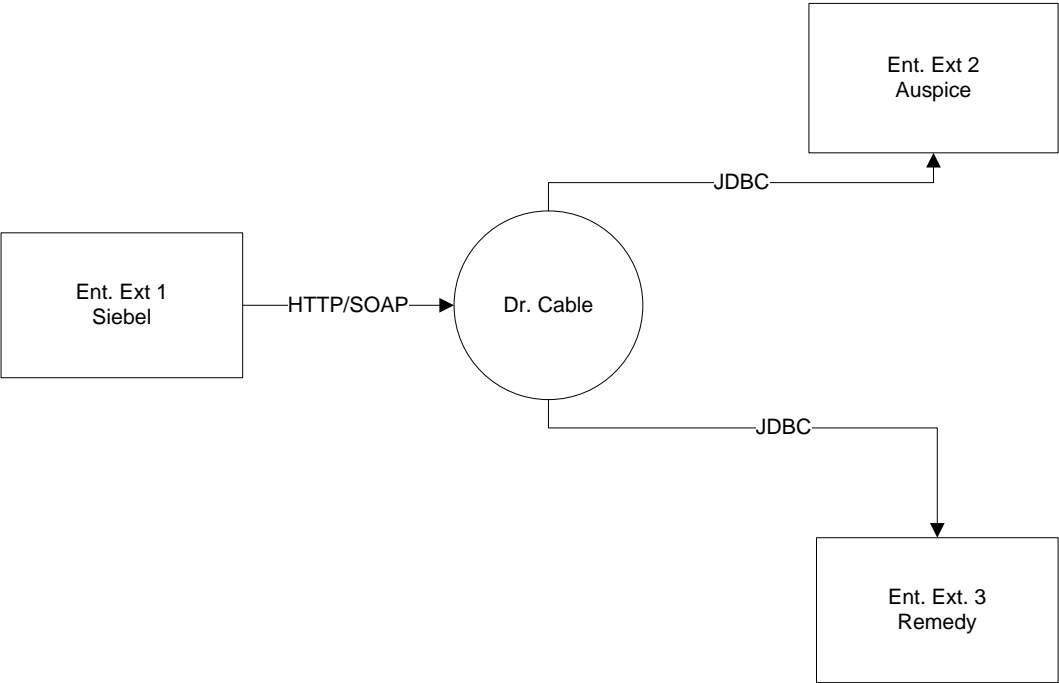
- **Manejo de excepción externa.** El framework captura la falla y realiza una acción especificada por el usuario, la cual se encuentra definida en un archivo de políticas de error (faultpolicy) asociada a la actividad

```
<faultName xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
  name="bpelx:remoteFault">
  <condition>
    <action ref="default-human-intervention"/>
  </condition>
</faultName>
<faultName xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
  name="bpelx:bindingFault">
  <condition>
    <action ref="default-terminate"/>
  </condition>
</faultName>
</Conditions>
<Actions>
  <!-- Generics -->
  <Action id="default-terminate">
    <abort/>
  </Action>
  <Action id="default-rethrow-fault">
    <rethrowFault/>
  </Action>
  <Action id="default-human-intervention">
    <humanIntervention/>
  </Action>
  <Action id="ora-retry">
    <retry>
      <retryCount>3</retryCount>
      <retryInterval>5</retryInterval>
      <exponentialBackoff/>
      <retryFailureAction ref="default-human-intervention"/>
    </retry>
  </Action>
</Actions>
</faultPolicy>
```

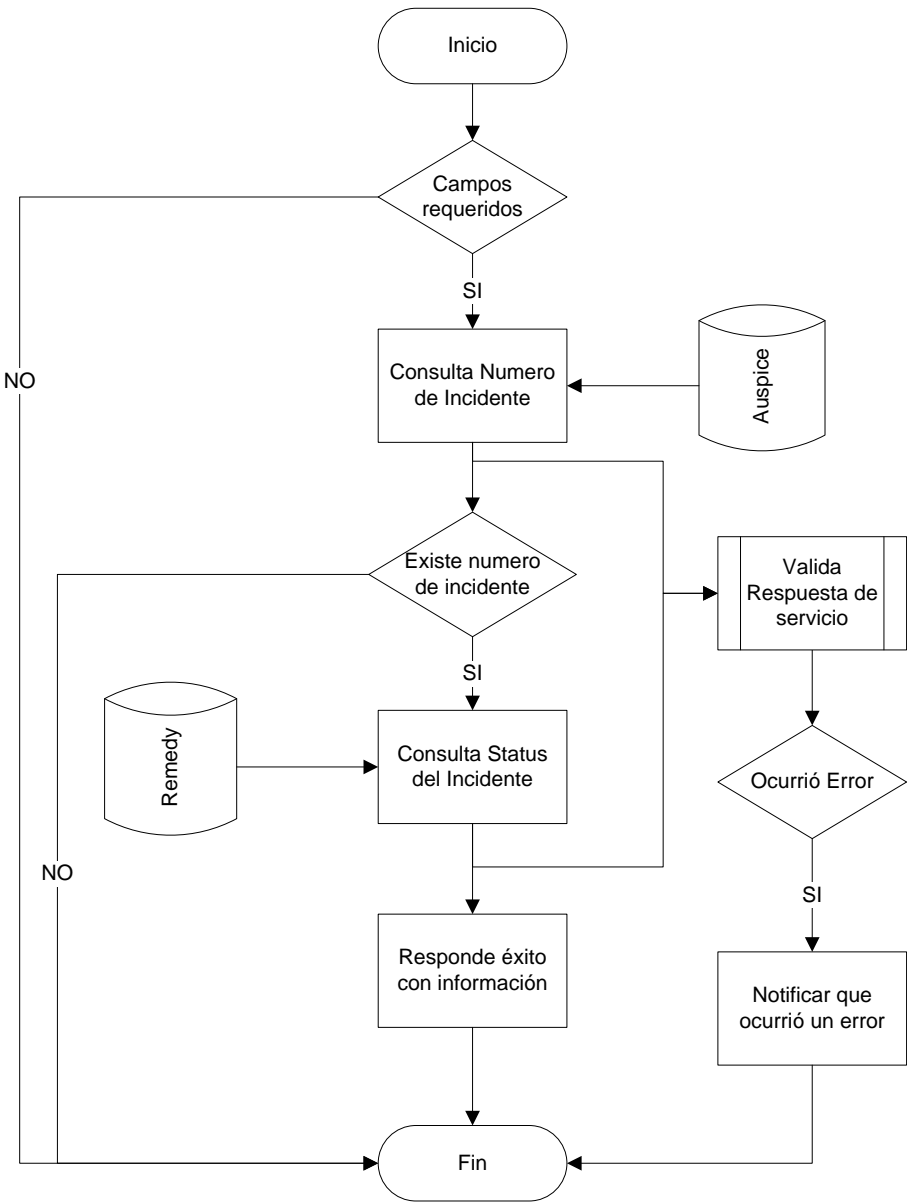
En el archivo de faultpolicy se configuran los reintentos que se tendrán para la ejecución de cada flujo BPEL en caso de existe algún error y de igual se especifica el tiempo de espera para la respuesta de cada servicio (timeout)

2.2 Vista Lógica

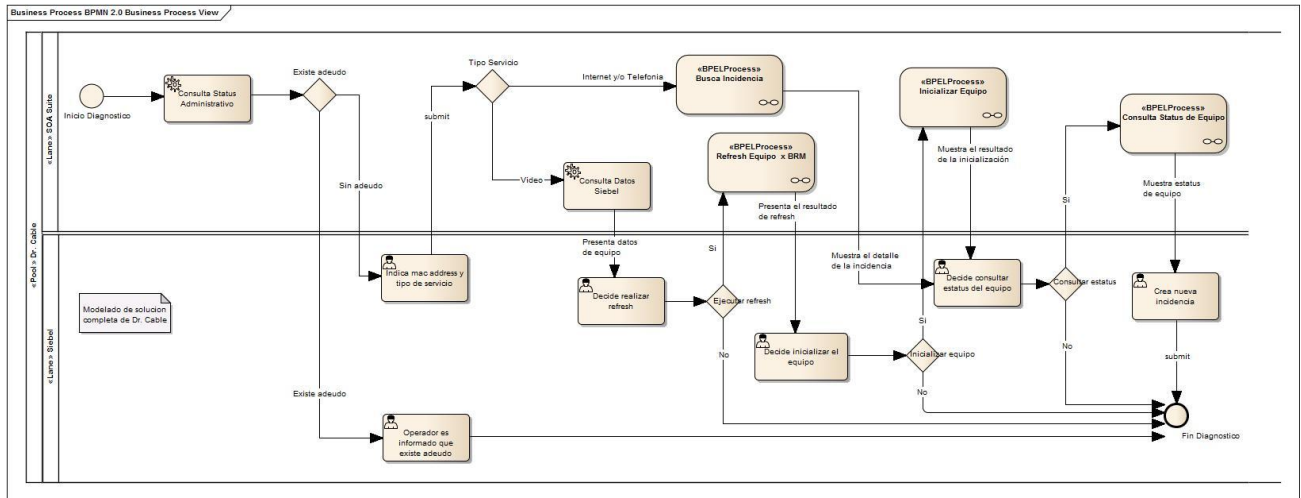
2.2.1 Diagrama de Contexto



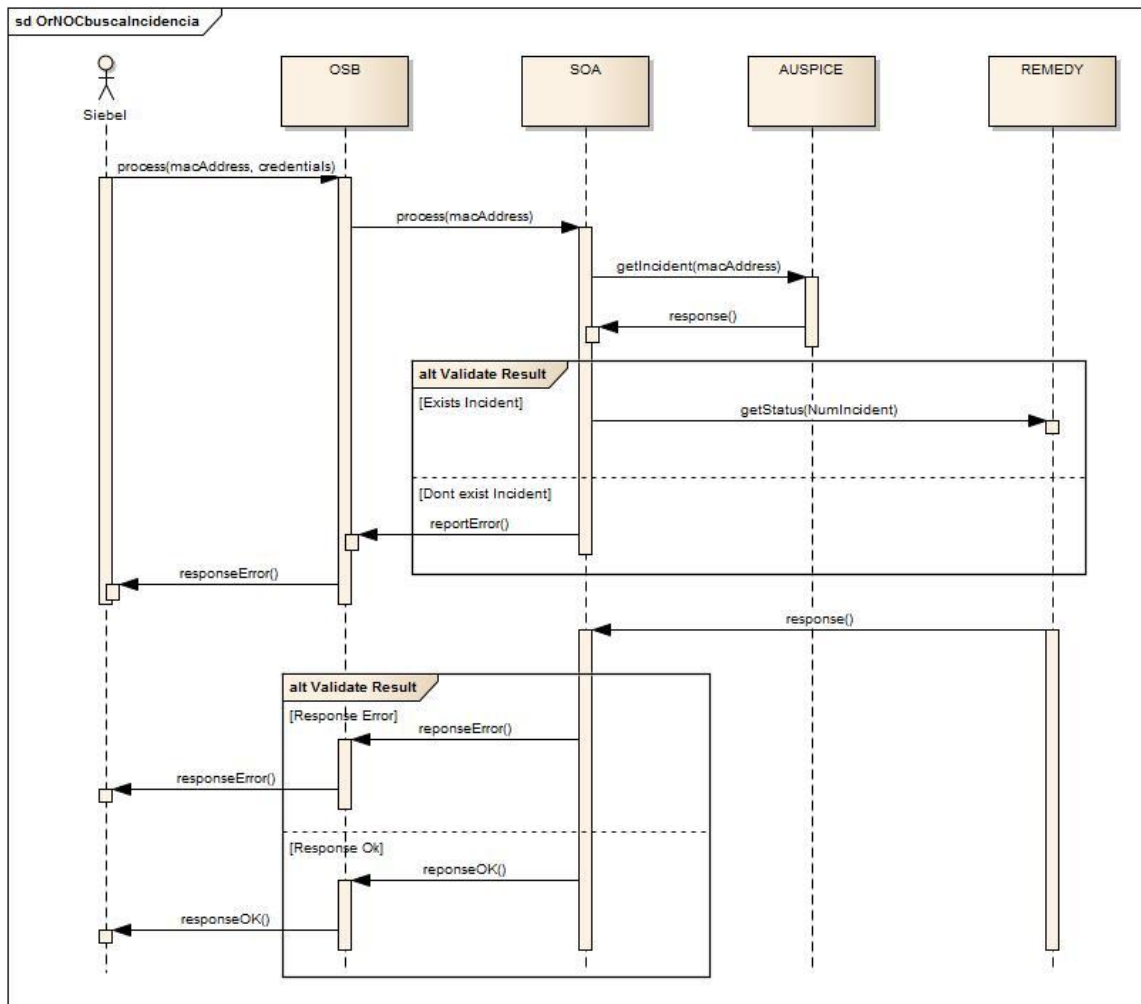
2.2.2 Diagrama de Flujo de datos



2.2.3 Diagrama BPMN



2.2.4 Diagrama de Secuencia



2.3 Vista Física

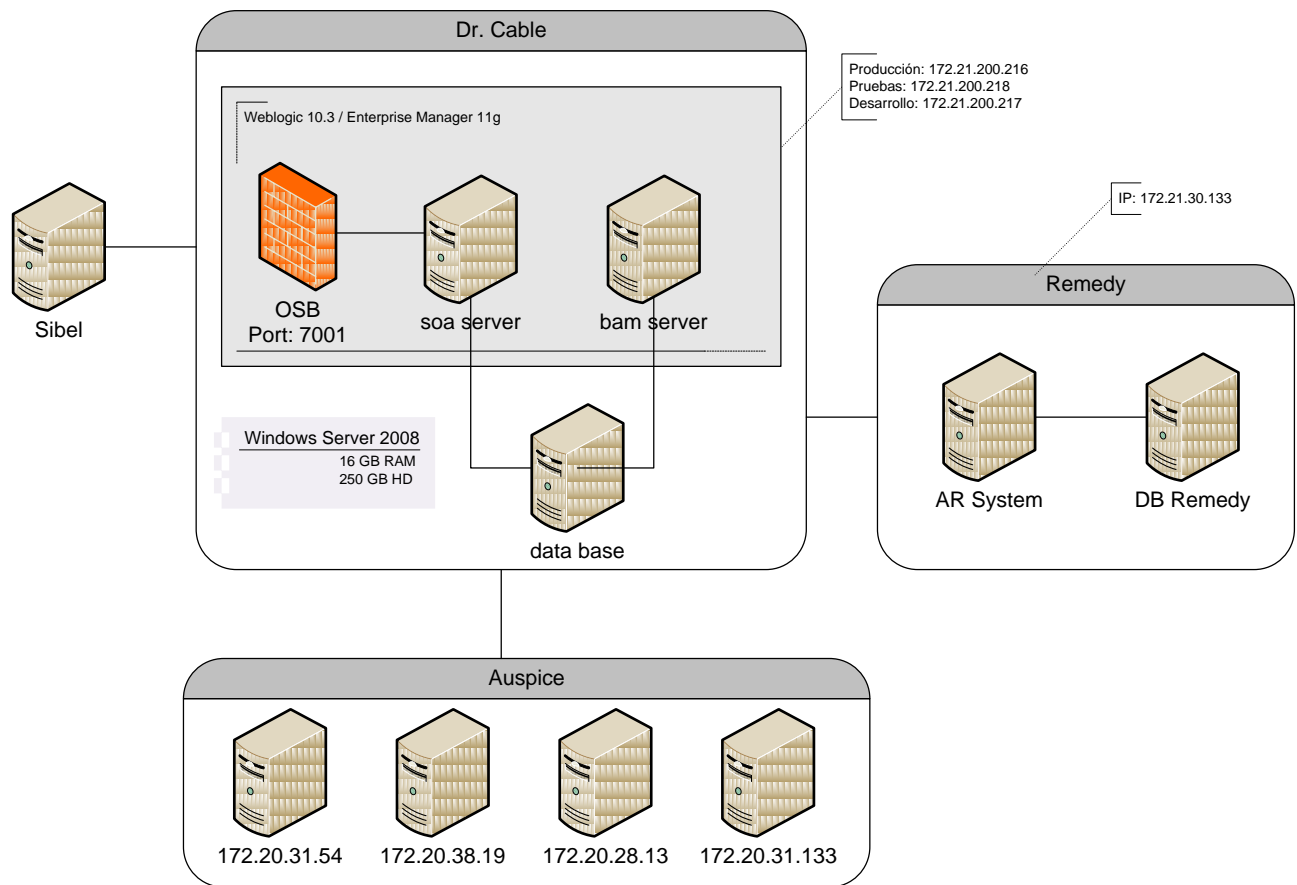
2.3.1 Diagrama de Infraestructura

El siguiente diagrama muestra los componentes de la infraestructura configurados para el funcionamiento de la SOA Suite. Se compone principalmente de los siguientes servidores:

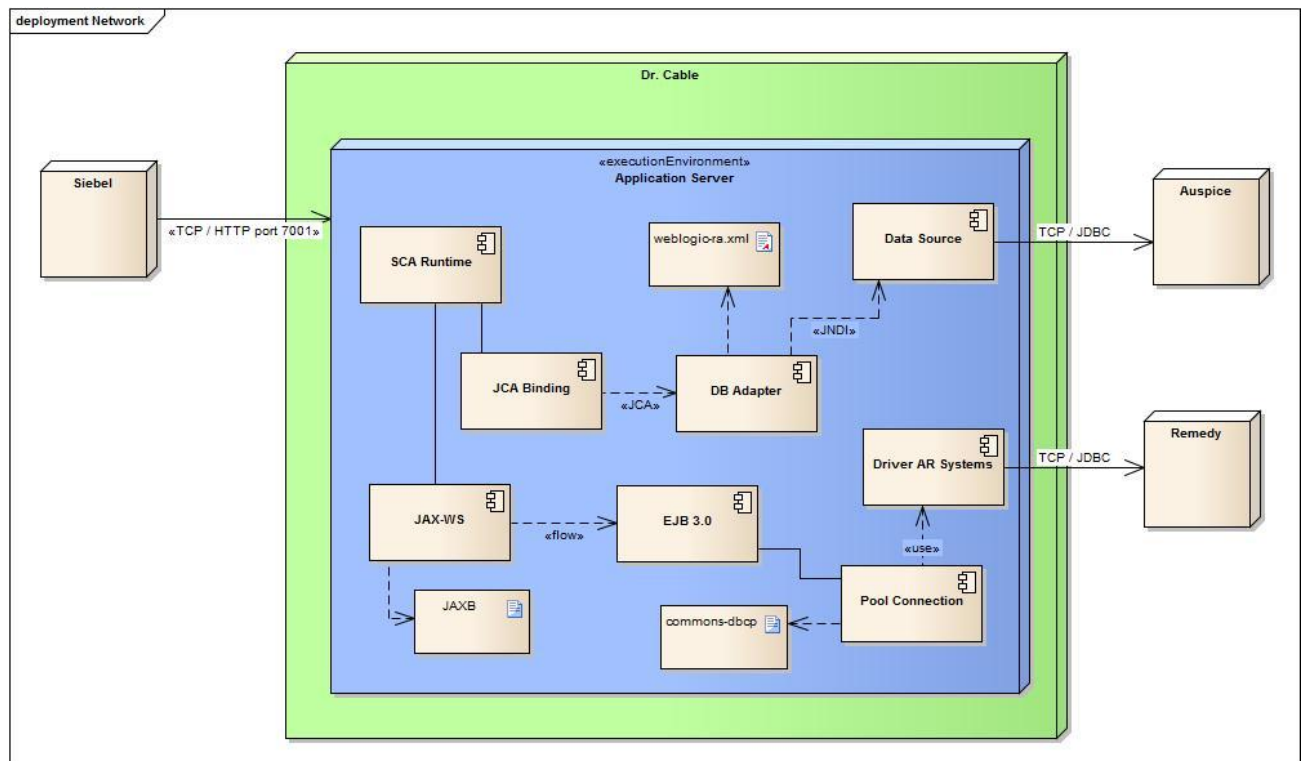
- **Servidor de Aplicaciones:** el servidor de aplicaciones de WebLogic ejecuta la Infraestructura de servicios de la SOA Suite, a través de la cual se ejecutan las aplicaciones compuestas que implementan los flujos BPEL.

En éste servidor se accede a las siguientes consolas de administración:

- **Consola de Administración de WebLogic:** Consola principal de administración del servidor de aplicaciones, a través de la cual se configuran objetos del contexto de ejecución como fuentes de datos.
 - **Consola del Enterprise Manager:** Consola de gestión y monitoreo del dominio de servidores, a través del cual se realizan actividades específicas para las aplicaciones compuestas como son pruebas unitarias, despliegues, bitácoras de ejecución, etc.
- **Servidor de Base de Datos:** Servidor de base de datos Oracle que almacena los esquemas de los Meta datos requeridos por la infraestructura de servicios.



2.3.2 Diagrama de Sistema



3 Diseño

3.1 Errores y Excepciones.

Catalogo de Error

Codigo	Descripción
0	Éxito
1001	Error al consultar Remedy
1002	No se encontro Ticket para la Mac Address <macAddress>
1003	Habilidades Extra no encontro informacion.
1004	Error al consultar estatus en Auspice
1005	No se encontro definicion del diagnostico en XML.
1006	No se encontro tecnologia asociada a mac address en XML.


Catalogo de Excepciones

Codigo	Descripcion
2001	concat(ora:getCompositeName()),':',ora:getComponentName()),':',ora:getFaultName())
2002	El formato de Mac Address es Incorrecto
2003	El tiempo de espera de respuesta del servicio se ha agotado

3.2 Mapeo Datos

3.2.1 Servicio OrNOCBuscaIncidencia

Datos de Entrada

Siebel	SOA	Remedy	Auspice
	macAddress		macAddress
	tipoServicio		
	rol		
	usuario		
		incidentNumber	

Datos de Respuesta

Siebel	SOA	Remedy	Auspice
	result		
	msg		
	det_error		
	siguienteOperacion		
	det_incidente :		
		assignedGroup	
		description	
		incidentNumber	
		lastModifiedBy	
		lastModifiedDate	
		productCatTier1	
		productCatTier2	
		resolution	
		resultMsg	
		status	
		submitDate	
		submitter	
		summary	
	det_equipo :		
			TROUBLE_TICKET_ID
			IP_ADDRESS
			MAC_ADDRESS
			HFC_GROUP
			CMTS_NAME
			US_CHANNEL_NAME
			US_CHANNEL_INDEX

			DOCSIS_VERSION
			DEVICE_TYPE
			VENDOR
			MODEL
			SOFTWARE_VERSION
			FIRMWARE_VERSION
			BOOTR
			LAST_STATE_CHANGE

3.2.2 Mapeo en Siebel

En las siguientes tablas se describen los campos de entrada, correspondientes al sistema de Siebel, y los campos que se pretenden recibir de respuesta de los sistemas de Auspice y Remedy:

Datos de Entrada

SIEBEL								
Business Component	Nombre del Campo	Tag	Requerido	Entrada/Salida	Tipo	Tamaño	Descripción	Reglas de Transformación
Asset Mgmt - Asset - Header	License Number	Unique Id/Mac Address	Y	Entrada/Salida	Varchar	30	Identificador único del equipo del cual se realizará la consulta en Remedy y en Auspice.	

Datos de Respuesta

REMEDY							
Business Component	Nombre del Campo	Tag	Requerido	Entrada/Salida	Tipo	Tamaño	Descripción
Dr. Cable VBC Remedy	Ticket	Ticket	N	Salida	Varchar		Número de ticket en Remedy que se encuentra asociado al equipo consultado.
Dr. Cable VBC Remedy	Status	Status	N	Salida	Varchar		Estado actual del ticket en Remedy.

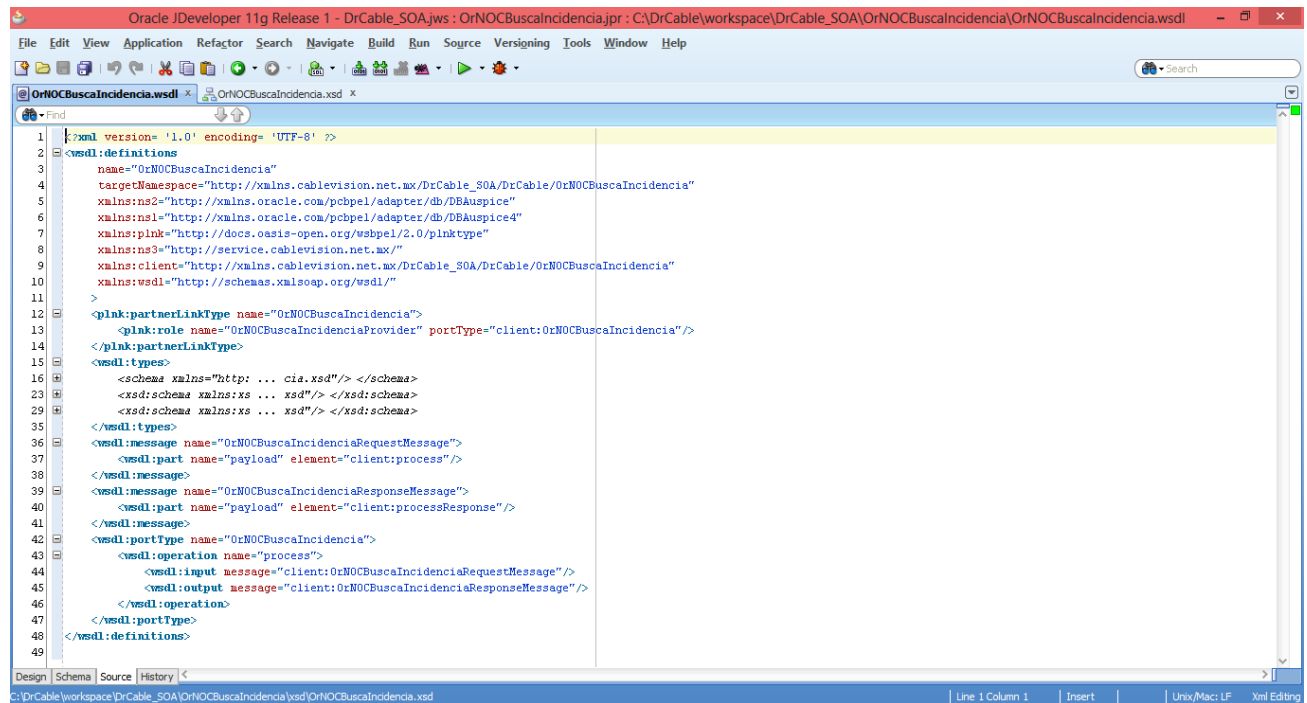
Dr. Cable VBC Remedy	Grupo Asignado	Grupo Asignado	N	Salida	Varchar		Grupo al cual ha sido asignado el ticket en Remedy (Call center, Soporte, etc...)
Dr. Cable VBC Remedy	Fecha de creación	Fecha de creación	N	Salida	Date		Fecha en la que fue creado el ticket en Remedy.
Dr. Cable VBC Remedy	Generado por	Generado por	N	Salida	Varchar		Nombre del responsable que generó el ticket de Remedy
Dr. Cable VBC Remedy	Fecha de última modificació n	Fecha de última modificació n	N	Salida	Date		Fecha de la última modificación realizada en el ticket de Remedy.
Dr. Cable VBC Remedy	Última modificació n realizada por	Última modificació n realizada por	N	Salida	Varchar		Nombre del responsable que realizó la última modificación en el ticket de Remedy
Dr. Cable VBC Remedy	Descripción	Descripción	N	Salida	Varchar		Descripción detallada del problema del equipo consultado
Dr. Cable VBC Remedy	Resolución	Resolución	N	Salida	Varchar		Descripción de la solución al problema descrito en el ticket en Remedy.
Dr. Cable VBC Remedy	IP Address	IP Address	N	Salida	Varchar		Dirección IP del equipo
Dr. Cable VBC Remedy	CMTS	CMTS	N	Salida	Varchar		Nombre de CMTS

Datos de Respuesta

AUSPICE							
Business Component	Nombre del Campo	Tag	Requerido	Entrada/Salida	Tipo	Tamaño	Descripción
Dr. Cable VBC Auspice	Modem Status	Modem Status	N	Salida	Varchar		Estado actual del modem en Auspice
Dr. Cable VBC Auspice	DHCP Server	DHCP Server	N	Salida	Varchar		Dirección del servidor DHCP
Dr. Cable VBC Auspice	CMT FTP Server	CMT FTP Server	N	Salida	Varchar		Dirección del servidor CMT FTP
Dr. Cable VBC Auspice	Upstream Channel	Upstream Channel	N	Salida	Varchar		Canal Upstream del equipo
Dr. Cable VBC Auspice	Upstream Channel Mode	Upstream Channel Mode	N	Salida	Varchar		Modalidad del canal de Upstream
Dr. Cable VBC Auspice	Upstream Transmit	Upstream Transmit Power	N	Salida	Varchar		Potencia de transmisión Upstream del equipo
Dr. Cable VBC Auspice	Downstream Receive	Downstream Receive Power	N	Salida	Varchar		Potencia de recepción Downstream del equipo
Dr. Cable VBC Auspice	CM	CM Configuration File	N	Salida	Varchar		Archivo de configuración CM
Dr. Cable VBC Auspice	Maximum Upstream	Maximum Upstream Bandwidth	N	Salida	Varchar		Máximo ancho de banda Upstream
Dr. Cable VBC Auspice	Maximum Downstream	Maximum Downstream Bandwidth	N	Salida	Varchar		Máximo ancho de banda Downstream

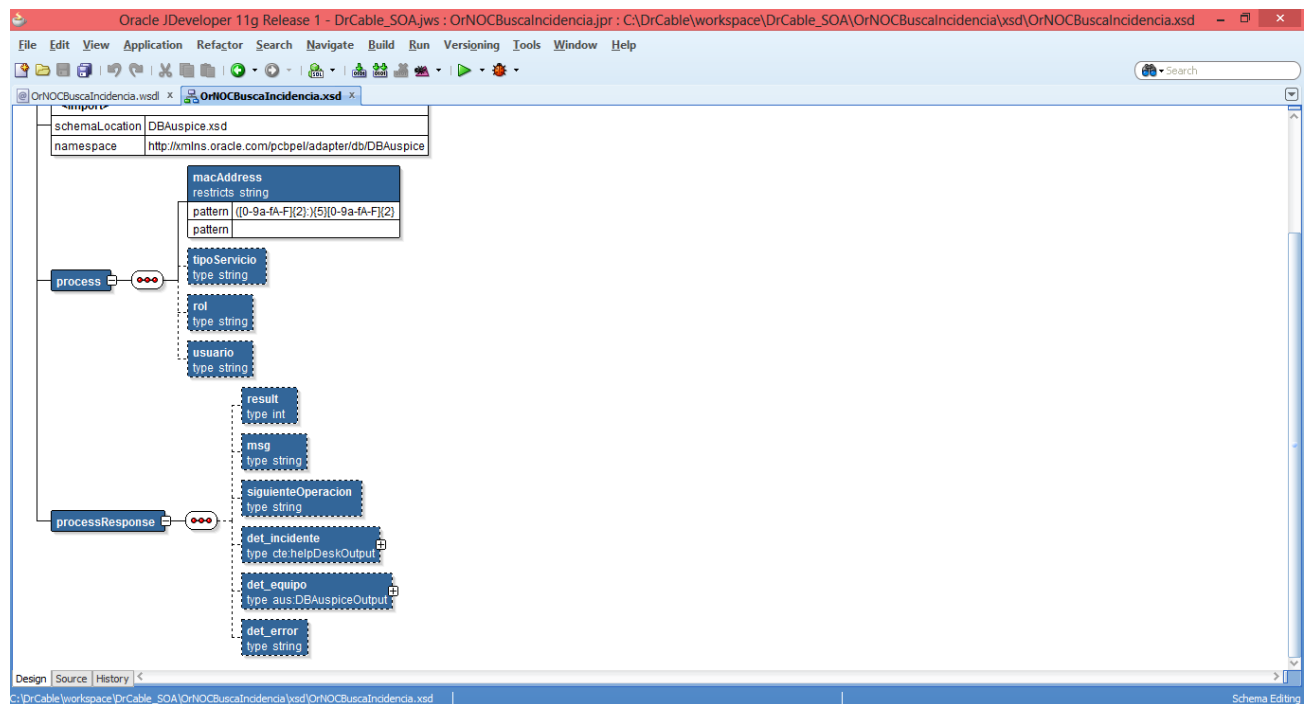
3.3 WSDL y XSD

3.3.1 Servicio OrNOCBuscaIncidencia



The screenshot displays the Oracle JDeveloper 11g interface with the WSDL file 'OrNOCBuscaIncidencia.wsdl' open. The XML content is as follows:

```
<?xml version='1.0' encoding='UTF-8' ?>
<wsdl:definitions
  name="OrNOCBuscaIncidencia"
  targetNamespace="http://xmlns.cablevision.net.mx/DrCable_SOA/DrCable/OrNOCBuscaIncidencia"
  xmlns:ns2="http://xmlns.oracle.com/pcbpel/adapater/db/DBAuspice"
  xmlns:ns1="http://xmlns.oracle.com/pcbpel/adapater/db/DBAuspice4"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plinktype"
  xmlns:ns3="http://service.cablevision.net.mx/"
  xmlns:client="http://xmlns.cablevision.net.mx/DrCable_SOA/DrCable/OrNOCBuscaIncidencia"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
>
  <plink:partnerLinkType name="OrNOCBuscaIncidencia">
    <plink:role name="OrNOCBuscaIncidenciaProvider" portType="client:OrNOCBuscaIncidencia"/>
  </plink:partnerLinkType>
  <wsdl:types>
    <schema xmlns="http:// ... cia.xsd"/> </schema>
    <xsd:schema xmlns:xs ... xsd"/> </xsd:schema>
    <xsd:schema xmlns:xs ... xsd"/> </xsd:schema>
  </wsdl:types>
  <wsdl:message name="OrNOCBuscaIncidenciaRequestMessage">
    <wsdl:part name="payload" element="client:process"/>
  </wsdl:message>
  <wsdl:message name="OrNOCBuscaIncidenciaResponseMessage">
    <wsdl:part name="payload" element="client:processResponse"/>
  </wsdl:message>
  <wsdl:portType name="OrNOCBuscaIncidencia">
    <wsdl:operation name="process">
      <wsdl:input message="client:OrNOCBuscaIncidenciaRequestMessage"/>
      <wsdl:output message="client:OrNOCBuscaIncidenciaResponseMessage"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```



4 Estándares

Los estándares de nombrado y guías de construcción dentro de un proyecto aseguran que los resultados producidos sean:

- Consistentes
- Comprensibles
- Eficientes
- Robustos
- Fáciles de dar mantenimiento

4.1 Estándares de tecnología

Los estándares de integración definidos entre los organismos para asegurar la consistencia de la implementación en la orquestación de los procesos de negocio de Cablevision, proveen dirección a la industria y estándares de tecnología, los cuales se enlistan a continuación:

- **W3C**
 - SOAP
 - WSDL
 - WS-Policy
 - WS-Security
 - XML Schema
 - XSLT
- **Oasis Open**
 - WebXML
 - SAML
 - UDDI
 - WS-BPEL
 - WS-ReliableMessaging
 - Web Service Addressing

4.2 Estándares de Construcción

Los estándares de construcción son reglas de programación que rigen a todos los componentes de software en los flujos BPEL que serán implementados. Dichos estándares no están enfocados a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código.

Los estándares son importantes dentro de la construcción por las siguientes razones:

- Disminuir el costo del mantenimiento de un programa
- Mejorar el entendimiento de lo construido, con mayor rapidez
- Facilitar la transferencia del conocimiento de forma clara y concisa

4.2.1 Espacio de nombres

El espacio de nombres (namespace) para los tipos de datos de los Servicios Web en los flujos de integración, es el siguiente:

<http://xmlns.cablevision.net.mx/>

4.2.2 Lineamientos para la orquestación de los flujos BPEL

Los siguientes lineamientos de construcción corresponden a la implementación de los flujos de integración, diseñados para el aprovechamiento de las capacidades que provee la herramienta JDeveloper para el desarrollo de aplicaciones SOA.

- Reutilizar los servicios BPEL que conforman los Partner Links del flujo, para su mayor funcionamiento
- Utilizar las funciones XPath proporcionadas por la herramienta, para el mejor funcionamiento de BPEL
- Todos los flujos BPEL deberán ir comentados. Más adelante, en la sección de Estándares de Documentación se muestra cómo hacerlo
- No se deben importar o añadir esquemas XSD dentro de un archivo ZIP, a un proyecto BPEL. Los esquemas deberán ser desempaquetados antes de ser importados al proyecto BPEL.
- No incluir ningún carácter especial en el nombre del proyecto. Podrían aparecer errores en la compilación
- El contenido del archivo bpel.xml sólo se lee en memoria cuando se abre el archivo. Por lo tanto, si cambia el contenido de bpel.xml después de que el archivo se abre, los cambios no se hacen en la memoria. Después de cambiar el contenido del archivo BPEL, es necesario reabrir el archivo para que los cambios surtan efecto
- JDeveloper deberá ser el único editor de código utilizado en la modificación de archivos bpel.xml, y demás archivos de configuración y de definiciones de datos (esquemas XSD, transformaciones, WSDL, composite).
- No editar el archivo bpel.xml, archivos BPEL, WSDL y archivos mientras se cambia el diseño del proceso.
- Cuando se despliega un proceso con el mismo nombre y la versión, se debe elegir entre sobrescribir el proceso BPEL actualmente desplegado, o implementar un nuevo número de versión. La directriz en general, es desplegar siempre a una nueva versión

4.2.3 Reglas de nombrado para flujos BPEL

Las reglas de nombrado para los objetos y actividades que se mencionan en las siguientes tablas, pueden tomar uno de los siguientes tipos de nombrado:

lowerCamelCase

Establece que la primera letra de cada palabra contenida en el nombre compuesto, con excepción de la primera palabra, deberá ser minúscula y todas las demás letras deberán ser mayúsculas.

Ej. bindingFaultHandlerProcess

UpperCamelCase

Establece que la primera letra de cada palabra contenida en el nombre compuesto deberá ser mayúscula.

Ej. BindingFaultHandlerProcess

4.2.3.1 General

Objeto	Nomenclatura	Descripción
BPEL Process	<Descripción>Process<ID Interfaz>	La palabra "Process" debe ser anexada al final del nombre. Los nombres de los procesos deben describir qué hace dicho proceso. Las interfaces de los procesos son la excepción y deben ser nombrados por su id de interfaz. El nombrado es de tipo <i>UpperCamelCase</i> Ejemplo correcto: BindingFaultHandlerProcess Ejemplo incorrecto: AP-INT-01
Partner Link	< Descripción>	Los Partner Links deben ser nombrados como sustantivos de tipo <i>UpperCamelCase</i> . Ejemplo: EmployeeCreator

4.2.3.2 Actividades

Actividad	Nomenclatura	Descripción
Assign	assign< Descripción >	<p>“assign” es seguido de la descripción de la asignación</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: assignPayroll</p>
Compensate	compensate<Scope>	<p>“compensate” es seguido del nombre del alcance.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: compensateEmployeeCreator</p>
Empty	empty<Descripción>	<p>“empty” es seguido de la descripción que indica para qué es utilizada la actividad vacía.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: emptyErrorHandler</p>
Flow	flow<Descripción>	<p>“flow” es seguido de la descripción de la actividad.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: flowUserCreation</p>
FlowN	flowN<Descripción>	<p>“flowN” es seguido de la descripción de la actividad, N es el valor especificado.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>,</p> <p>Ejemplo: flow5UserCreation</p>
Invoke	invoke<Descripción>	<p>“invoke” es seguido de la descripción de lo que está siendo invocado.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: invokeOracleDBIntegrator</p>
Java Embedding	embed<Descripción>	<p>“embed” es seguido de una descripción de lo que hace el código incrustado.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: embedUserValidation</p>
Notification	notification<Descripción>	<p>“notification” es seguido de una descripción de la notificación enviada.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: notificationEmployeeCreation</p>

Actividad	Nomenclatura	Descripción
Pick	pick<Descripción>	<p>“pick” es seguido de una descripción que indique el proceso que se está evaluando.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: pickGLEntryUpload</p>
Receive	receive<Descripción>	<p>“receive” es seguido de una descripción de lo que se está recibiendo.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: receivePayroll</p>
Reply	reply<Descripción>	<p>“reply” es seguido de una descripción de lo que se está enviando.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: replyInvoices</p>
Scope	scope<Descripción>	<p>“scope” es seguido de la descripción de las actividades que se están realizando dentro del scope.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: scopeInboundToOracleTransform</p>
Sequence	sequence<Descripción>	<p>“sequence” es seguido de la descripción de lo que la secuencia representa.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: sequenceEmployeeCreation</p>
Switch	switch<Descripción>	<p>“switch” es seguido de la descripción de para qué es utilizada la actividad switch.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: switchMessageType</p>
Terminate	terminate<Descripción>	<p>“terminate” es seguido de una descripción del estado del proceso que se está terminando.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: terminateSuccess</p>
Throw	throw<Descripción>	<p>“throw” es seguido de una descripción de la excepción que se está arrojando.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: throwBindingFault</p>

Actividad	Nomenclatura	Descripción
Transform	transform<Descripción>	<p>“transform” es seguido de una descripción de la transformación en progreso.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: transformInboundToOracle</p>
UserTask	userTask<Descripción>	<p>“userTask” es seguido de una descripción de qué tarea se tiene que realizar por un usuario.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: userTaskEmployeeApproval</p>
Wait	wait<Descripción>	<p>“wait” es seguido de una descripción de porqué o para qué hay espera.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: waitDailyScheduled</p>
While	while<Descripción>	<p>“while” es seguido de una descripción de la condición que debe ser verdadera para que continúe el ciclo.</p> <p>El nombrado es de tipo <i>lowerCamelCase</i>.</p> <p>Ejemplo: whileIsControlRecord</p>

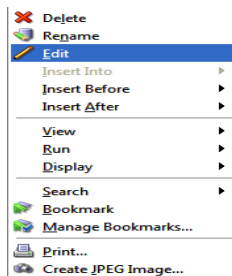
4.2.4 Estándares de documentación

Los flujos BPEL se deberán documentar con el código generado con la herramienta de desarrollo de JDeveloper, la cual permite la inserción de anotaciones en los distintos elementos del flujo BPEL (excepto el elemento Switch ya que contiene ramificaciones y cada elemento que lo integra deberá tener su propia anotación).

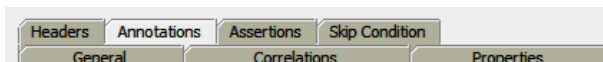
Las Annotations en JDeveloper habilitan una descripción de la actividad y tiene la opción de asignarle valores.

1. Dentro de un flujo BPEL se elige alguno de sus elementos y se accede al menú contextual:

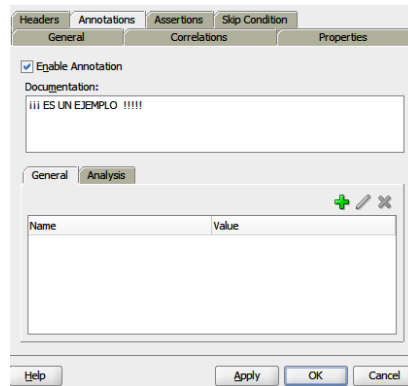
- Aparecerá el siguiente menú



- Se elegirá la opción Edit
- Se desplegará una nueva interfaz



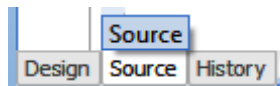
- Se elegirá la pestaña de Annotations
- Se habilitara la opción de Annotation
- Se agregara un texto alusivo y referente al elemento
- Se agregaran elementos (opcional)



- Se aceptan los cambios con Ok

2. Las anotaciones se visualizaran dentro del código fuente:

- Por default JDeveloper muestra la vista de diseño; se cambiara a la vista de código. fuente
- Se elegirá la pestaña de Source



- Desde esta perspectiva se visualiza lo desarrollado en las anotaciones correspondientes

```

76 <invoke name="Invoke_FileWriter2"
77     inputVariable="Invoke_FileWriter2_Write_InputVariable"
78     partnerLink="FileWriter2" portType="ns2:Write_ptt"
79     operation="Write">
80   <bpelx:annotation>
81     <bpelx:documentation>!!! ES UN EJEMPLO !!!!</bpelx:documentation>
82   </bpelx:annotation>
83 </invoke>

```