
 Banco Bolivariano <small>El Banco con Visión</small>	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 1 de 37

Arquitectura Tecnológica

Estándares de desarrollo en Plataforma Bus Empresarial

Versión 2.1
05/07/2016



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 2 de 37


Elaborado por: Arquitectura y Soporte de TI Fecha:05/07/2016	Revisado por: Arquitectura y Soporte de TI Fecha: 05/07/2016	Aprobado por: Subgerente de Arquitectura y Soporte de TI Fecha: 05/07/2016
--	---	---

Contenido

1. OBJETIVO.....	5
2. ALCANCE	5
3. CONDICIONES GENERALES.....	6
Entornos	6
4. DESARROLLO DEL DOCUMENTO.....	7
Principios SOA.....	8
Servicio	8
Identificación de servicios SOA	8
Guía para Identificar los servicios de aplicación	9
Patrones de diseño	10
5. Capas de la Arquitectura SOA.	11
6. Herramienta estándar de Modelado de análisis y diseño (Enterprise Architect)	12
Servicios de Aplicación.	13
7. Orquestación de Servicios	14
8. Estrategia de conexión entre servicios	14
Modelo canónico	14
9. Mecanismos de comunicación.....	15
Excepciones al uso del ESB.	15
10. Arquitectura de servicio basada en componentes SCA.....	16
11. Patrones de tipo de mensaje.....	16
Document message	16
Event message	16
12. Patrones de canal	17


ESTA PROHIBIDA LA IMPRESIÓN DE ESTE DOCUMENTO. UNA VEZ TERMINADA LA REVISIÓN DEL MISMO, EL DOCUMENTO WORD COPIADO EN SU MÁQUINA DEBE SER ELIMINADO. LA ÚNICA VERSIÓN VÁLIDA DISPONIBLE ES LA PUBLICADA EN EL SISTEMA DE GESTIÓN DOCUMENTAL.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 3 de 37


Bus de mensajes.....	17
Publish-subscribe.....	17
13. Patrones de implementación del servicio	17
Patrón de fachada.....	17
14. Patrones de servicio de mensajes	17
Asynchronous Queuing	17
Event-Driven Messaging	17
15. Patrones de enrutado	17
Basado en contenido	17
Implementación redundante	17
WS-Addressing	17
WS-ReliableMessaging	17
WS-Policy	18
WS-MetadataExchange	18
16. Seguridad	18
Estándares de Seguridad propuestos.....	18
Servicios de seguridad Web	18
WS-Security	19
Archivos Adjuntos en los mensajes.....	19
Tipos de transporte	19
A nivel de Transporte.....	19
A nivel de aplicación	20
Uso de la herramienta Framework de seguridad,.....	20
17. Gobierno SOA	20
18. Arquitectura de productos Middleware	22
19. Estándares de Desarrollo	23
Recomendaciones de desarrollo.....	23
20. Restricciones de diseño	26
a. Bases de datos.....	26
b. Aplicaciones	26



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 4 de 37

c.	Integración.....	26
21.	Entorno Metodológico	26
22.	Política de Versionado.....	27
23.	Política de certificados.	27
	Certificados admitidos.....	27
	Organización de los certificados.....	28
24.	Política de Documentación.....	29
25.	GLOSARIO	35
26.	DOCUMENTOS DE REFERENCIA	36
27.	ANEXOS.....	36
28.	CONTROL DE CAMBIOS.....	37



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 5 de 37

1. OBJETIVO

El presente documento tiene por objeto describir los estándares, herramientas y metodologías que rigen los servicios y desarrollos de TI. Esta guía y todos sus anexos se encuentran publicados en el portal departamental de sistemas

Estándares de desarrollo

El responsable del contenido de este documento es el área de Arquitectura y Soporte de TI.

Cualquier sugerencia o duda, petición de reunión, reporte de incompatibilidad del documento con las directrices corporativas, o petición de adaptación a casos concretos de los estándares definidos en este documento deberá dirigirse a Subgerencia de Arquitectura y Soporte de TI.

El presente documento establece los entregables y uso común de herramientas, estando permitidos cuantos entregables y actividades adicionales se consideren oportunos desde la jefatura de cada proyecto o en las reuniones de definición con los Arquitectos designados para la revisión.


Documentar los estándares de programación a ser utilizados en la construcción de aplicaciones y de los servicios a ser desplegados en la plataforma de Bus empresarial.

El presente documento detalla los elementos de diseño, desarrollo, configuración y despliegue, de las aplicaciones de integración del Banco y de los proveedores de servicios.

2. ALCANCE

Entorno	Definir los estandares a nivel de la infraestructura, software base, esquemas de seguridad, etc.
Principios de Diseño	Definir los estandares de diseño de aplicaciones, servicios web, flujos de integración, etc.
Desarrollo:	Definir los estandares de escritura de codigo, incluye nombres de los componentes, nombres de variables, nombres de objetos, clases, etc. Definir desarrollo estandar para el desarrollo en los flujos de integración.
Configuración	Definir configuración estandar del software base y de las aplicaciones a ser utilizadas en los flujos de integración.
Gobierno	Definir estandares de comunicación para poder mantener y controlar el crecimiento de los servicios en la organización



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 6 de 37

3. CONDICIONES GENERALES

Entornos

A continuación se describe la infraestructura tecnológica y metodológica sobre la que se construyen los Servicios de la Institución que sean considerados multicanal.

Entorno de Servidores

Entorno estándar

La plataforma tecnológica estándar para el desarrollo de servicios SOA o microservicios dentro de la institución es la siguiente:

Hardware Los equipos host están basados en plataforma Integrity HP tecnología Itanium IA64. Los diferentes maquinas virtuales que conforman la plataforma esta virtualizadas con Tecnología HP VPAR.

Sistema Operativo. HP-UX versión 11.31

Servidores Sistemas Integración

- Servidor Web (Entorno Unix)
 - Oracle HTTPS Server (Web Tier)
- Servidor de Aplicaciones (entorno Unix):
 - Oracle Weblogic Suite en un entorno cluster
 - Oracle SOA Suite en un entorno cluster.
- Java EE 6

Gestor de Bases de Datos para repositorio de la plataforma de integración:

- Oracle Database 11G en un entorno cluster.

Conexión a aplicaciones internas

La integración entre sistemas se realiza mediante servicios de negocio.

La integración entre aplicaciones se evaluara el uso de micro-servicios.

Los servicios de negocio y/o micro-servicios se publican con mecanismos estándar.

- Servicios WEB.

Se estable como Protocolo estándar de comunicación entre aplicaciones web o de escritorio internas HTTPS.

Se estable como formato de mensaje estándar entre aplicaciones web o de escritorio internas XML


Tipos de servicios web:

- SOAP
- REST

La seguridad de los servicios web queda definida:

- A nivel de la capa de transporte mediante el uso de certificados digitales



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 7 de 37

- HTTPS - TLS 1,2
- Algoritmo de firma SHA2
- A nivel de servicios web con WS-Security, administrado por la herramienta de la plataforma del ESB -Web Service Manager
 - Cifrado de contenido XML mediante el uso de certificados digitales
 - Algoritmo de firma SHA2

Conexión a proveedores externos.

Se establece como estándar de integración entre aplicaciones el uso de servicios web.

Se establece como Protocolo estándar de comunicación entre aplicaciones web o de escritorio internas HTTPS.

Se establece como formato de mensaje estándar entre aplicaciones web o de escritorio internas XML.

Se establece como excepción al estándar de integración entre aplicaciones de proveedores externos que usen TCP/IP/ISO el uso de servicios web como enlace hacia las aplicaciones internas.

Se establece como Protocolo estándar de comunicación entre aplicaciones web o de escritorio internas HTTPS.

Se establece como formato de mensaje estándar entre aplicaciones con proveedores externos ISO8583. La seguridad de los servicios web queda definida:

- A nivel de la capa de transporte mediante el uso de certificados digitales
 - HTTPS - TLS 1.2
 - Algoritmo de firma SHA2
- A nivel de servicios web con WS-Security, administrado por la herramienta de la plataforma del ESB - WEB SERVICE MANAGER-
 - Cifrado de contenido XML mediante el uso de certificados digitales
 - Algoritmo de firma SHA2

Otros entornos.


Los desarrollos realizados sobre entornos distintos al descrito en el punto anterior deberán ser presentados al área de Arquitectura y soporte de TI para su evaluación, análisis y autorización.

Cualquier formato de mensaje no contemplado deberá tener la aprobación de excepción de la Subgerencia de Arquitectura y Soporte de TI.

4. DESARROLLO DEL DOCUMENTO

Para el desarrollo de los servicios se ha establecido como estándar la arquitectura de aplicaciones orientada a servicios (SOA), se enumeran a continuación los puntos a considerar para el desarrollo de los servicios en la arquitectura seleccionada.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 8 de 37

Principios SOA

1. **Contratos de servicio estandarizados.**- Los servicios adhieren a un acuerdo de comunicación definido en los descriptores de servicios. Por medio de WSDL (Web Service Description Language) como lenguaje estándar definido por el W3C.
2. **Bajo acoplamiento.**- La relación entre los servicios minimiza las dependencias, los contratos logran un bajo acoplamiento con el consumidor del servicio y también con el entorno.
3. **Abstracción.**- Fuera de lo que se describe en los contratos de servicio, los mismos ocultan su lógica al mundo exterior.
4. **Reutilización.**- Utilizar todos estos activos encapsulándolos en servicios para que a su vez puedan ser reutilizados por otros servicios.
5. **Autonomía.**- Los servicios tiene control total sobre la lógica que encapsulan.
6. **Sin estado.**- Los servicios minimizarán la retención de información específica de una actividad.
7. **Capacidad de descubrimiento.**- Debe existir un catálogo donde se puedan buscar o descubrir los servicios existentes en el dominio de negocio.
8. **Composición.**- Se combinarán colecciones de servicios coordinándolos y ensamblándolos de manera que se formen servicios compuestos.
9. **Interoperabilidad.**- Los contratos estándares ayudan mediante la armonización de los modelos de datos. El desacoplamiento ayuda tener menos dependencias y por lo tanto que los consumidores y proveedores de servicios puedan tener una naturaleza totalmente distinta.

Servicio


Una función sin estado, auto-contenida, que acepta una(s) llamada(s) y devuelve una(s) respuesta(s) mediante una interfaz bien definida.

Identificación de servicios SOA

Para la identificación de servicios se realizó las siguientes técnicas y métodos, que deben ser utilizadas como insumo para el cumplimiento del estándar:

- Tener en cuenta los principios de SOA.
- Encuadrar los servicios según su tipo, por ejemplo, servicios de negocio, servicios de proceso, servicios de aplicación, servicios de datos, micro-servicios.
- A partir de los procesos de negocio en descomposición, los servicios que creamos responden a una necesidad de negocio, los objetivos o metas de la empresa. Por ejemplo, “apertura de cuentas”.
- Análisis de las necesidades de las aplicaciones FrontEnd que sirve de interfaz con el cliente, para determinar el contrato del servicio necesario para resolver la necesidad.
- Asegurarse que los nombre de los servicios sean claros dentro del contexto de creación del servicio.
- Los nombres de los servicios deben tener al menos dos partes: verbo + entidad.
- El nombre de los servicios deben usar el estándar lower camel case (estándar de nombramiento) – elimina la necesidad de caracteres especiales y permite leer correctamente los servicios, e.g. primeraPalabraConMinusculasLuegoCadaPalabraIniciaConMayusculas.
- No mencionar detalles de implementación en el nombre.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 9 de 37

- No incluir el protocolo del servicio en el nombre.
- No incluir la palabra “servicio” o sus siglas en el nombre.
- No incluir la versión del servicio en el nombre.

Guía para Identificar los servicios de aplicación

En la implementación de los flujos de integración se utilizarán los siguientes estándares de la industria a ser implementado son:

Composición de Aplicaciones (SCA): permite crear aplicaciones o soluciones orientadas a servicios por medio de la coordinación y ejecución de componentes. Los componentes que pueden estar dentro de una composición son:

Productos adquiridos	Productos usados en la implementación
Flujo de integración BPEL	Flujo de integración BPEL
Mediador	Mediador
Reglas de Negocio	Reglas de Negocio
Adaptadores	Adaptadores

Las aplicaciones SCA se implementarán en el Oracle SOA Suite.

Orquestación (WS-BPEL): permite orquestar servicios e implementar flujos de integración. Este estándar es implementado en el componente de orquestación BPEL dentro del Oracle SOA Suite.

Políticas (WS-Policy): Nos permitirá definir reglas o políticas sobre los servicios que van hacer expuestos con la ayuda de estándares WS-Security para aplicarlos en los servicios expuestos en el OSB (Oracle Service Bus).


Entre las especificaciones soportadas, las que se van usan en la aplicación son las siguientes:

- **Policy:** Se refiere al conjunto de información que se expresa como aserciones de directivas.
- **Policy Assertion:** Representa una preferencia individual, la necesidad, capacidad, etc.
- **Policy Expression:** Conjunto de uno o más aserciones de directivas.
- **Policy Subject:** Una entidad a la que una expresión de la política se puede enlazar.

Seguridad (WS-Security): Asegurará las composiciones de aplicaciones y componentes expuestos como servicios en el OSB. Se aplicará en conjunto al WS-Policy.

Transacción (WS-BusinessTransaction/WS-AtomicTransaction): Se podrá administrar la transacionabilidad. WS-BusinessTransation permite administrar transacciones de larga duración. WS-AtomicTransaction permiten administrar transacciones atómicas entre servicios. Los dos estándares son aplicados en el componente de orquestación BPEL dentro del ámbito Oracle SOA Suite.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 10 de 37

Coordinación (WS-Coordination): Podremos permitir la coordinación y ejecución de servicios en una composición de servicios. Este estándar es aplicado parcialmente desde el OSB, y utilizado en la composición de aplicaciones SCA.

Direccionamiento (WS-Addressing): Se manejará el direccionamiento en la coordinación de servicios. Este estándar es manejado para la replicación de mensajes o respuestas asincrónicas (callback) desde una composición de aplicaciones SCA.

Mensajería Fiable (WS-ReliableMessaging): La fiabilidad en el mensaje en la implementación nos permitirá garantizar la entrega de mensajes entre componente. Este estándar es aplicado parcialmente desde el OSB.

Patrones de diseño

Los patrones de diseño a utilizar en el diseño e implementación de los servicios son:

Patrones de Centralización de Inventario.

- **Centralización de Políticas.-** Estandariza y alinea políticas de seguridad de los servicios por ser implementados mediante token-clave y mensaje seguro.
- **Centralización de Esquemas.-** Define un modelo general de esquema usando el modelo canónico actual del banco.

Patrones de Transformación.

- **Transformación de Formato de Datos (Data Format Transformation).-** Se implementará para la transformación de ISO8583 – XML y XML-ISO8583.
- **Transformación de Modelo de Datos (Data Model Transformation).-** Transformaciones de SP a definiciones necesarias para el modelo de negocio.
- **Puente entre Protocolos (Protocol Bridging).-** Se va usar para el manejo de SOCKET-SOAP y de SOAP-SOCKET.

Patrones de composición de aplicaciones

- **Compensación en Transacción de servicios.-** Mediante la compensación de reverso necesario para los flujos.


Patrones de encapsulación

- **Legacy Wrapper.-** Patrón para usarlos en los SP y exponer su funcionalidad.

Patrones de Mensajería Servicios

- **Encolamiento Asincrónico servicios.-** Necesario para los servicios altamente transaccionales permitiendo un manejo adecuado de transacciones.
- **Enrutamiento intermedio.-** Enrutamiento necesario en el transcurso del flujo de servicios.
- **Mensajería Confiable.-** Necesario para la confirmación que el mensaje haya llegado a su destino para su debido funcionamiento.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 11 de 37

5. Capas de la Arquitectura SOA.

Toda implementación debe cubrir los conceptos de SOA, los mismos que deben ser validados por el área de arquitectura.

✓ **Capa de componentes de servicio (Service Components Layer)**

- Es la implementación de un servicio a través de la Arquitectura de Componentes (SCA).

✓ **Capa de servicios (Services Layer)**

- Esta capa contienen los servicios de negocio. Tiene un descriptor, WSDL, que igualmente es estándar y que define el interfaz del servicio (con qué parámetros se invoca y cuáles devuelve).
- Se permite crear servicios compuestos, combinando otros servicios más simples.

✓ **Capa de procesos de negocio (Business Process Layer)**

- En esta capa se encuentran los procesos BPM (coreografía de servicios) que modelan casos de uso específicos y procesos de negocio. Cada proceso es un flujo de tareas automáticas o humanas.


✓ **Capa de consumidores (Consumers Layer)**

- Aquí se encuentran todos los consumidores de los servicios. Debido a la característica de multicanalidad del ESB y dado la cantidad de dispositivos que se tienen actualmente para acceder a los mismos:
 - Aplicaciones de escritorio de la empresa,
 - Switch de cajeros automáticos,
 - Aplicaciones con interfaz web del canal internet banca personas y empresas,
 - Aplicaciones para dispositivos móviles del canal Banca móvil, tales como smartphones y tablets
 - Socios del banco (esquema B2B),
- Todos los servicios están dados de alta en el registro de servicios a fin de que los consumidores puedan localizar el servicio que necesiten y sean independientes de la localización física del servicio.

✓ **Ámbito de integración (Integration Layer):**

- Esta es una capa de infraestructura que proporciona las capacidades de mediación, enrutado y transporte de la petición desde el cliente hasta el servicio concreto.
- Estas capacidades se implementan mediante el Enterprise Service Bus (ESB)
 - Como herramienta para aplicar Arquitectura Orientada a Servicios; se define como estándar para integración multicanal; el uso de un ESB. El mismo se usa para integrar los procesos de negocio, sistemas operacionales y datos.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 12 de 37

- Para poder comunicar a través de distintos canales la información y los datos.
- El ESB usa WSDL y XML como estándares abiertos e interoperables, sobre una capa de transporte, HTTP asegurada mediante SSL/TLS.

✓ Capa de calidad del servicio (QoS Layer)

- Esta capa proporciona la capacidad de realizar los requerimientos no funcionales necesarios, por ejemplo, monitorización de los servicios, trazas, alertas de no cumplimiento del nivel de servicio acordado (SLA), etc.
- Asegura que se cumplen los niveles mínimos de escalabilidad, disponibilidad y seguridad necesarios.
- Desde el punto de vista de negocio, proporciona la funcionalidad de Business Activity Monitor (BAM). Esto se hace mediante el desarrollo dentro del proceso de KPI (Key Performance Monitor) que son recogidos por el BAM.

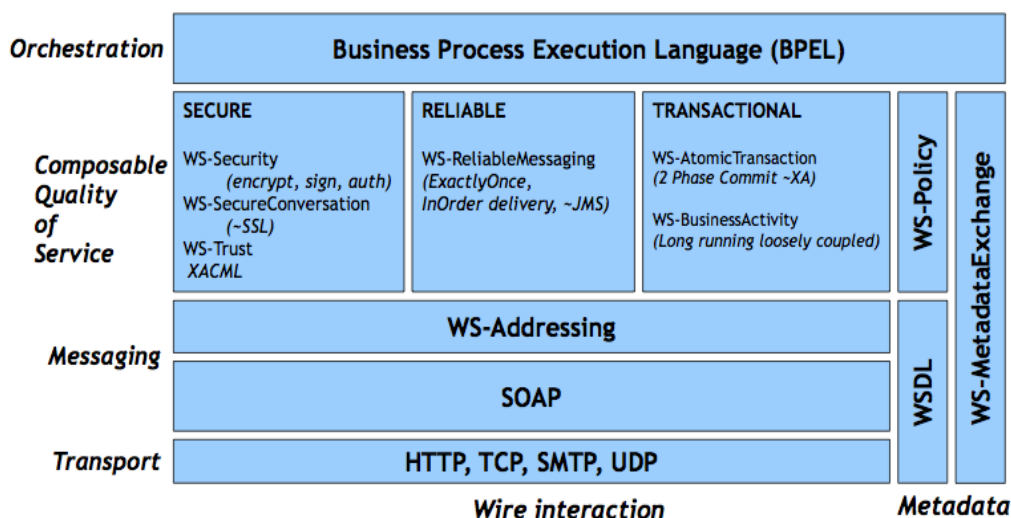


Fig- Esquema de interacción del ESB.


6. Herramienta estándar de Modelado de análisis y diseño (Enterprise Architect)

Enterprise Architect es la herramienta seleccionada para el modelado de los diferentes diagramas necesarios para el diseño de las aplicaciones.

Esto incluye los diversos diagramas de diseño, las orquestaciones de los servicios de aplicación, la definición del portafolio de servicios y modelo canónico de datos.

Cumple la función de repositorio, permite identificar gráficamente la reutilización de servicios, el grado de acoplamiento de un nuevo proceso e impacto de los cambios.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 13 de 37

Para la construcción de los modelos Enterprise Architect, se ha definido como estándar 2 repositorios de datos:

Como área de trabajo para el modelado de los diagramas de aplicativos, servicios, infraestructura que se encuentren en etapa de desarrollo se ha creado el repositorio DES_REPO.

Una vez que los proyectos son aprobados y asados a producción se pasara al repositorio definitivo ARQ_REPO, para resguardo.

Adicionalmente dado las limitaciones en licenciamiento se ha definido como método de consulta de los diversos diagramas de los servicios; en los diferentes repositorios desarrollo y producción; el uso de la misma herramienta Enterprise Architect en su versión Lite, la misma que no tiene costo y funciona como visor, manteniendo toda la funcionalidad de la herramienta, limitando la capacidad de edición.

Adicional y como medida de contingencia en caso de daños al repositorio, se ha procedido al levantamiento de los servicios de aplicación (Desarrollo-Arquitectura) del banco, un documento básico de referencia **“MAPA DE SERVICIOS Y RECAUDACIONES POR CANAL.xls”**.

Para mantener actualizado este documento se usara los registros de casos de uso revisados por Arquitectura.

Servicios de Aplicación.

Se seguirá el siguiente formato para cada **servicio de aplicación** nuevo o modificación que se requiera:

Tipo de servicio: identifica el tipo de servicio puede ser este: Servicio público, Servicio privado

Servicio: describe el servicio que se está consumiendo o proveyendo puede ser este: servicios básicos, Recargas prepago, servicios de tarjetas de crédito, de debito, etc.

Clase: identifica la diversidad del servicio que se presta, Agua Potable, Energía Eléctrica, Teléfono,

Descripción: Explicación en forma detallada, clara y ordenada del servicio de manera funcional. Por ejemplo para el servicio de *“consultar saldo”*: Entrega la consulta de saldo para un cliente del banco y una cuenta asociada al mismo.


Aplicativos: Nombre del aplicativo que genera el servicio. Por ejemplo: Banca en Línea, Activos, BPM, Pasivos, SAT entre otros.

Expone y/o Consume: Definir si es un servicio que consume otros servicios de proveedores externos. O si es un servicio con el cual proveemos alguna funcionalidad al proveedor externo.

Reutilizable: Si el servicio en mención está siendo utilizado por otro aplicativo. Si se considera reusable o no.

Ambiente: En qué ambiente de integración esta publicado el servicio. ESB, CSP, OTRO



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 14 de 37

Nombre del Proveedor: Nombre del proveedor externo del Banco. Está relacionado con el campo origen solo debe ser llenado si origen es proveedor. En caso de ser un servicio del banco Se define como INTERNO

Formato de mensaje: definir el formato del mensaje si es: ISO8586; XML, Trama propietaria, X11, QueryString, etc.

Mecanismo de Comunicación: Como el servicio se está o debería comunicarse con su entorno. Por ejemplo: TCP/IP, WS, RPC, RMI, otros.

Documento adjunto: “**MAPA DE SERVICIOS Y RECAUDACIONES POR CANAL.xls**”

7. Orquestación de Servicios

Se definen orquestaciones de los servicios de aplicación por cada uno de los servicios de negocio definidos. Las orquestaciones están definidas en lenguaje de modelaje de procesos de negocio BPMN del Open Group (<http://www.omg.org/spec/BPMN/2.0>).

En el siguiente documento está explicado brevemente como crear una orquestación de servicios dentro del Enterprise Architect.

Aplican los procedimientos y el estándar de diseño y desarrollo actual del grupo de BPM.

8. Estrategia de conexión entre servicios

Para realizar el diseño de mensajes se ha seguido el modelo:

Canonical Message Model (CMM)

Con la finalidad de reducir la complejidad en el intercambio de mensajes por medio del ESB, aplica cuando proveedores y consumidores de servicios intercambian entre sí mensajes en diferentes formatos, sean estos de carácter técnico o de negocio.


El ESB entiende todos los modelos de mensaje propietario de los consumidores y proveedores de servicios.

El ESB debe proveer la transformación para cada una de las interacciones entre servicios.

Modelo canónico

CMM estandariza los modelos de mensajes entre los consumidores y proveedores de servicios. Si un nuevo servicio es introducido, sólo se debe crear la transformación entre el servicio y el CMM. Reduce el número de transformaciones en un ambiente heterogéneo.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 16 de 37

- ☑ Se deberá evaluar otros mecanismos de comunicación para los servicios que durante las pruebas no cumplan con el rendimiento esperado.
- ☑ El ESB no proveerá funcionalidad de sistemas legados –estos sistemas deben proveer interfaces o WRAPPERS para su conexión y acceso.
- ☑ El ESB no manejará procesos en batch o lógica de negocio – el ESB habilita el intercambio de mensajes entre diferentes aplicaciones – Esta limitante debe ser analizada con el especialista asignado a la revisión del diseño.

10.Arquitectura de servicio basada en componentes SCA

Es una tecnología de software creada por IBM y Oracle. Es un conjunto de especificaciones que describen un modelo para construir aplicaciones y sistemas basadas en una Arquitectura Orientada a Servicios. Este define los componentes que consumen y exponen servicios al igual que los contratos. Cada componente al menos tiene una interfaz.

Acogiendo el símil de una placa base de PC, la arquitectura SCA está formada por componentes internos integrados entre sí, que se comunican con sistemas externos mediante conectores, todo ello a través de cables (*wires*). Ya sea en grandes sistemas SOA (Enterprise SOA) como pequeños (uso de Web Services para conectar sistemas), la pieza restante es un estándar que agrupe los servicios individuales en un servicio compuesto de alto nivel. Esta pieza es **SCA**.

Los tres elementos principales que apoyan el desarrollo de aplicaciones compuestas SCA son:

1. Herramienta de desarrollo en JDeveloper
2. Infraestructura de ejecución unificada en SOA Suite
3. Capacidad de gestión y configuración desde Enterprise Manager

La infraestructura unificada se presenta como un conjunto de **motores** de ejecución que proporcionan la funcionalidad necesaria para la ejecución de los elementos que componen el compuesto SCA (BPEL, Reglas de negocio, invocaciones, etc.).

11.Patrones de tipo de mensaje

Document message

Estos servicios puede permitir el intercambio de documentos entre los servicios que lo invoquen


Por ejemplo: referencias personales, comerciales, en el caso de empresas documentos de constitución de la misma, etc.

Event message

Estos servicios necesitan coordinar las acciones en las que manejan movimientos en las cuentas, al generar una notificación del evento de acreditar o debitar en la cuenta, permitirá que una herramienta de monitoreo pueda reaccionar en el caso de que la acción realizada no esté dentro de los parámetros normales del comportamiento del cliente.

De la misma forma va a permitir hacer un mejor seguimiento de los movimientos si se define un log centralizado.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 17 de 37

12. Patrones de canal

Bus de mensajes

Permite que los sistemas se comuniquen con interfaces compartidos, para poder usarlos desde diferentes canales.

Publish-subscribe

Al implementar este patrón, va a permitir que las áreas involucradas reciban la solicitud de crédito y pueda iniciarse el proceso en paralelo.

13. Patrones de implementación del servicio

Patrón de fachada.

Para permitir la lógica pueda evolucionar de manera independiente. Debe implementarse en todos los servicios.

14. Patrones de servicio de mensajes

Asynchronous Queuing

Debe implementarse en todos los servicios, para el incremento de la velocidad y disponibilidad.

Event-Driven Messaging

En este caso es de mucha ayuda para validar que bajo ciertas condiciones se dispare un evento de notificación, por ejemplo: este puede generar un evento en el caso de que se trate de evaluar a un cliente vip o alto staff del banco.

15. Patrones de enrutado

Basado en contenido

Estos servicios necesitan revisar el contenido de sus acciones en las que manejan movimientos en las cuentas, al revisar el contenido se puede generar una notificación del evento que permitirá reaccionar en el caso de que el contenido no esté dentro de los parámetros establecidos y dirigir el flujo a otra área para su seguimiento y control.

Implementación redundante.

Es de gran utilidad desplegar los servicios que necesitamos y para disponer de alta tolerancia a fallos.

Especificaciones WS-*

Dentro de la arquitectura SOA para la implementación de Servicios Web, la interoperabilidad es tal vez el principio más importante. Como método de implementación de SOA, Web Services debe ofrecer importantes beneficios de interoperabilidad, y permitir la ejecución de servicios Web distribuidos en múltiples plataformas de software y arquitecturas de hardware.

La WS-I ha organizado los estándares que afectan a la interoperabilidad de los servicios *web* basada en funcionalidades.


WS-Addressing

Mayor interoperabilidad ya que mediante este estándar, es el mismo mensaje el que dice cómo hay que tratarlo. Delega la lógica al Middleware.

WS-ReliableMessaging

Para proveedores externos, esta mensajería es fiable, con garantía que la secuencia correcta en la recepción de los mensajes.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 18 de 37

El servicio de origen envía el mensaje al Middleware y éste se encarga de transmitirlo al destino.

Define un protocolo que permite que los mensajes SOAP se envíen de forma fiable entre consumidor y proveedor a pesar de los posibles fallos de red o de aplicación que puedan ocurrir. El protocolo define varios tipos de entregas: AtLeastOnce, AtMostOnce, ExactlyOnce, InOrder.

WS-Policy

Proporciona un mecanismo para anexar propiedades (reglas, comportamientos, requerimientos, preferencias) a los servicios web y otros recursos. Restricciones y reglas que puede describir, requerimientos a nivel de negocio, que tipos de datos se pueden intercambiar.

WS-MetadataExchange

Permite a los consumidores obtener metadatos de los proveedores de servicios. Se estandariza la petición de información que hace el consumidor. Facilita el desacoplamiento de los servicios al ser más autónomos en cuanto a los metadatos que gestionan. Complementa los metadatos que ya proporciona el WSDL.

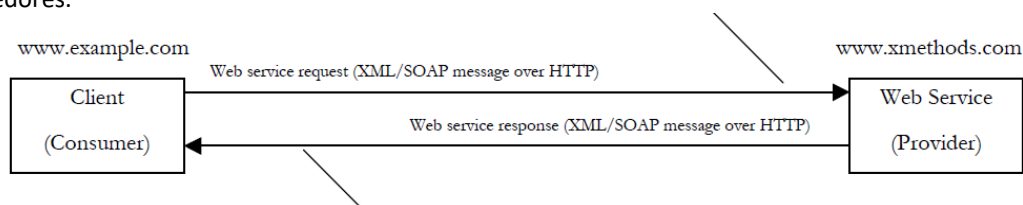
Operaciones incluidas: Get WSDL, Get Schema, Get Policy, Get Metadata.

16. Seguridad

Debido a la implementación del ESB, ahora los servicios están expuestos en la Web, para asegurar su origen e integridad, se recomienda el siguiente estándar:

Estándares de Seguridad propuestos.

El propósito de una infraestructura de SOA es que los consumidores puedan invocar servicios expuestos por los proveedores.




Los servicios Web se basan en estándares de la industria basados en XML: El formato de datos que permite la comunicación uniforme entre los consumidores de servicios web y los proveedores de servicios web es el XML:

- Un framework que describe vocabularios XML que se utilizan en las transacciones comerciales (XML Schema).
- Un sobre usado para enviar solicitudes estructuradas, y recibir respuestas estructuradas a partir del proveedor de servicios web (SOAP).
- Un lenguaje que define lo que hace un servicio Web (WSDL)
- Un framework para publicar y buscar servicios web en Internet (Universal Description, Descubrimiento e integración - UDDI).

Servicios de seguridad Web

Debido a su naturaleza (conexiones débilmente acoplados) y su uso del acceso abierto (principalmente HTTP), las infraestructuras SOA implementadas por los servicios web añaden un nuevo conjunto de requisitos para la seguridad.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 19 de 37

Servicios de seguridad Web incluye varios aspectos:

- **Autenticación:** verificar que el usuario es quien dice ser. La identidad de un usuario se verifica en base sobre las credenciales presentadas por el usuario, como nombre de usuario/ contraseña, certificado digital, Standard Security Assertion Markup Language (SAML) token. En el caso de los servicios web, las credenciales se presentan por un cliente aplicación en nombre del usuario final.
- **Autorización (o de control de acceso):** La concesión de acceso a los recursos específicos.
- **Confidencialidad, privacidad:** mantener información en secreto. Se puede lograr mediante la encriptación de los contenidos de la solicitud o respuesta mensajes utilizando el estándar XML Encryption.
- **La integridad:** Asegurarse de que un mensaje permanece inalterado durante el tránsito por tener una autoridad de firmar digitalmente el mensaje, una firma digital también valida el remitente y el proporciona un sello de tiempo. Mensajes XML se firman mediante el estándar XML Signature. En muchos casos, las soluciones de seguridad de servicios web, tales como Oracle WSM dependen de la clave pública (PKI).

Requisitos de seguridad de servicios Web son compatibles con estándares de la industria, tanto en el transporte nivel (Secure Socket Layer).

WS-Security

Para firmar mensajes, podemos saber en este caso quien envía una petición a un servicio web o de quién es la respuesta con WS-Signature.

También sirve para asegurarnos de que alguien no ha modificado el mensaje (Integridad).

Para cifrar el mensaje y asegurarnos de que nadie puede leer el contenido utilizaremos XML-Encryption.

Lightweight Directory Access Protocol; es un protocolo que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

Archivos Adjuntos en los mensajes.

Se define que se les de el tratamiento como si fueran archivos adjuntos en e-mails.

- Bloquear los anexados.
- Filtrado de los tipos sospechosos,
- Examinarlos con un anti-virus antes de utilizarlos.

Se usará un Gateway de XML que “controle la frontera” de la empresa; mediante otro servidor de aplicaciones. Para lo cual utilizaremos una solución hardware: firewall XML.

Tipos de transporte


A nivel de Transporte

Se aplica la seguridad el protocolo mediante el que viajan los datos por la red (no los datos en sí). Normalmente se hace sobre HTTP, añadiendo Secure Sockets Layer (SSL) que proporciona Autenticación, Integridad, Confidencialidad.

La seguridad de los servicios web queda definida:

- A nivel de la capa de transporte mediante el uso de certificados digitales
 - HTTPS - TLS 1,2
 - Algoritmo de firma SHA2
- A nivel de servicios web con WS-Security, administrado por la herramienta de la plataforma del ESB - Web Service Manager - XML-Encryption.
 - Cifrado de contenido XML mediante el uso de certificados digitales



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 20 de 37

- Algoritmo de firma SHA2

A nivel de aplicación

La seguridad a nivel de aplicación complementa la seguridad de nivel de transporte. La seguridad a nivel de aplicación es basado en XML que definen marcos de confidencialidad, la integridad, la autenticidad, la estructura del mensaje; gestión de la confianza, la propagación de la identidad.

Uso de la herramienta Framework de seguridad,

El uso del framework de seguridad dependerá del diseño de la aplicación; este será requerido en el escenario en el que la aplicación/ el canal deba usar alguna de las funcionalidades que brinda la herramienta, autenticación, manejo de llaves de cifrado, etc.

Desde el Web Services que tiene el núcleo del FrameWork se podrá disponer de los siguientes Servicios:

- ❖ ServicioAdministracion.asmx
- ❖ ServicioCatalogo.asmx
- ❖ ServicioComunicacion.asmx
- ❖ ServicioEDirectory.asmx
- ❖ ServicioMTRA.asmx
- ❖ ServicioParticipante.asmx
- ❖ ServicioSeguridad.asmx
- ❖ ServicioSQL.asmx
- ❖ ServicioSybase.asmx
- ❖ ServicioUnix.asmx

A estos se los podrá acceder remotamente vía URL mediante la una dirección similar a la que sigue:

<http://NombreServidorWeb/WSFrameWork/ServicioAdministracion.asmx>

o

<http://NombreServidorWeb/WSFrameWork/ServicioCatalogo.asmx>


El nombre del servidor dependerá del ambiente en el que se encuentre en desarrollo, preproducción o producción. Confirmar con el área de arquitectura cual es el nombre del servidor a utilizar dependiendo del ambiente.

Todos los Servicios Web (exceptuando los de Seguridad que están indicados) cuentan con métodos que reciben uno o dos parámetros tipo XML. Siempre el Primer (o único) XML que reciben estos métodos contendrá información de los permisos de la Sesión o Login y generalmente será un elemento Raíz "Usuario".

17. Gobierno SOA

El esquema de gobierno, tiene como propósito estructurar la toma de decisiones de TI, desde la definición de la visión de TI hasta el desarrollo de nuevos componentes.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 21 de 37

La publicación de los servicios en la plataforma cuenta con dos herramientas:

Service Registry, donde está la definición de los contratos y servicios que han sido publicados.

Oracle Service Registry es el registro de los servicios empresariales que proporciona una base para el gobierno y la gestión del ciclo de vida de sus servicios de negocio.

Proporciona lo que necesita para obtener una visión de toda la empresa, el control y la influencia económica de los activos de servicios empresariales de la organización.

Mucho más que un registro UDDI, el Registro de captura y hace descripciones de servicios empresariales que pueden descubrirse en una ubicación gestionada de forma centralizada, fiable y de búsqueda, convirtiéndose en el sistema de registro para sus servicios de negocio.

<http://172.16.30.152:7101/registry/uddi/web>

Registry Control

Administración del Service registry, se necesitan permisos para registrar servicios.

<http://172.16.30.152:7101/registry/uddi/bsc/web>

Business Service Control

Acceso público del Service registry, no se necesita credenciales.

Enterprise Repository, es donde reside la documentación y se administra el gobierno del servicio.

Oracle Enterprise Repository (OER), es un registro de los servicios de negocio, proporciona la base para el gobierno y administración del ciclo de vida de los mismos. El registro provee de lo necesario para una visión de toda la empresa, el control y la influencia económica de los activos de servicios empresariales.


Permitiendo el almacenamiento y gestión de metadatos extensible para compuestos, servicios, procesos de negocio y otros activos relacionados con TI, actúa como la principal fuente de información SOA, que le permite encontrar los servicios planificados, existentes, y retirados.

En OER permite realizar las siguientes tareas:

- Registro de activos (Contratos, esquemas, proxys, etc.)
- Revisión de información registrada por el usuario con rol de submitter
- Asignación de recurso para revisión de documentación del servicio
- Registro de información del activo
 - Overview
 - Clasificación
 - Relaciones
 - Información técnica
 - Versionamiento
 - Soporte
- Registro de autorizaciones
- Liberación al Repositorio

Una vez que se tiene los datos registrados y se registre el servicio en el repositorio, este podrá ser consultado desde el portal web de la herramienta para conocer su información.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 23 de 37

19. Estándares de Desarrollo

La forma común de invocar un compuesto es utilizar SOAP sobre HTTP. Es posible mediante la creación de un servicio SOAP para el compuesto utilizando web service binding.

Este esquema está permitido en servicios que no sean muy transaccionales y bajo la aprobación del especialista designado para la revisión del servicio

Sin embargo, también se puede utilizar la opción direct binding, que ofrece una alternativa de integración más estrecha. Direct binding permite a los clientes Java invocar directamente los composite service, sin pasar por la conversión intermedia a XML requerido con el enlace de servicio web.

Por temas de rendimiento en el desarrollo del OSB y SOA se ha definido que en la invocación componentes internos se use el método de invocación direct binding.

Direct binding proporciona dos tipos de estilos de invocación:

- Inbound direct binding
- Outbound direct binding (or direct reference binding)

El estilo a ser usado dependerá de la implementación, pero por omisión se utiliza Outbound Direct Binding.

El transporte SOA-DIRECT ofrece conectividad nativa entre Oracle Service Bus y componentes de servicio Oracle SOA Suite. Oracle SOA Suite proporciona un marco de "unión directa" que le permita exponer los componentes del servicio Oracle SOA Suite en una aplicación compuesta, y el transporte Oracle Service Bus SOA-DIRECT interactúa con los servicios expuestos a través del marco de unión directa SOA, dejando que interactúen los componentes de servicio en la capa de bus de servicios y aprovechar las capacidades y características de Oracle Service bus.

Recomendaciones de desarrollo.

19.1.1 Revisar la siguiente URL para entender como implementar DirectBinding en JDeveloper (39.1 , 39.3 , 39.3.1, 39.3.2)

https://docs.oracle.com/cd/E23943_01/dev.1111/e10224/invocapi.htm

Revisar la siguiente URL para entender como implementar sbTransport en los proxy services (27.2)


https://docs.oracle.com/cd/E28280_01/dev.1111/e15866/sb.htm

Funcionamiento de sb transport y soa direct transport (24.2.1)

https://docs.oracle.com/cd/E23943_01/dev.1111/e15866/soa.htm

Nota: Todas los servicios reutilizables en las composiciones soa ya tienen el proxy expuesto con protocolo sbTransport solo se debe hacer la referencia desde los nuevos desarrollos.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 24 de 37

19.1.2 Considerar que al desarrollar en JDeveloper revisar el archivo composite.xml, en la sección de las referencias a las bases de datos, se deben eliminar las propiedades que aplican reintentos, estas son:

- jca.retry.count,
- jca.retry.interval,
- jca.retry.backoff,
- jca.retry.maxInterval.

19.1.3 Se debe hacer la revisión de los programas fuentes y en el config.plan; para que se haga la modificación necesaria para que sean configurados las url del ambiente de producción de modo que se reemplacen completamente las del ambiente de desarrollo.

19.1.4 Se debe establecer en los Business Services las siguientes configuraciones, dependiendo de la forma como se comporta el proveedor de servicio y el acuerdo de disponibilidad del mismo.

- Tiempo de lectura, valor numérico mayor a 0
- Tiempo de conexión, valor numérico mayor a 0
- Reintentar tras Errores de Aplicación, se debe establecer en NO.

El especialista en la revisión del servicio, definirá si alguno de estos parámetros debe ser configurado fuera de estándar.

19.1.5 Se debe validar que los servicios no apunten a los equipos de OSB/SOA (soa-infra); se debe siempre usar el http server (web tier).

19.1.6 Validar en los esquemas .xsd que hacen referencia a campos TINYINT en la base de datos, ya que en la conversión los rangos seleccionados por la herramienta son diferentes, por ese motivo en el .xsd debe ser reemplazarlo por INT.

19.1.7 Cuando se necesite que los datos en el XML respeten los espacios en Blanco en las aplicaciones que usan DOM del lado del cliente, se debe configurar el atributo xml:space="preserve".

Ejemplo en ASP:


```
Dim RequestRoot, docReceived
Set docReceived = Server.CreateObject("MSXML.DOMDocument")
docReceived.PreserveWhitespace=true
docReceived.loadxml(VLRequest)
```

19.1.8 Se define como obligatorio no invocar el servicio web SecuencialReverso, se debe invocar directo a la base de datos al stored procedure pa_CanalesNoCobis_interfase para los flujos en caso que necesiten recuperar el secuencial o parámetros por defecto de COBIS. Si algún flujo es modificado y no tiene este estándar debe ser modificado.

19.1.9 Las referencias desde el SOA al OSB realizadas a:

- HomologarMensaje,
- RegistrarError,
- ValidarRespuesta




	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 25 de 37

Deben utilizar protocolo t3 (Direct Binding).

- 19.1.10 Para las referencias a proveedores externos se debe crear un proxy con protocolo sb Transport y será llamado desde el SOA con Direct Binding, esto aplica para cualquier proxy que seá invocado desde el SOA.
- 19.1.11 Las referencias internas del OSB al SOA , el BPEL debe ser expuesto como DirectBinding y desde el OSB el Bussines Service debe ser protocolo SOA-Direct.
- 19.1.12 Para mejorar el tiempo de respuesta, la llamada al BAM debe ser realizada después del replyOutput.
- 19.1.13 En los BPELS, se debe manejar un control de errores General y uno por cada invocación de un Partner links.
- 19.1.14 No se debe usar Mediadores a nivel del SAO, la mediación se deberá realizar en el OSB. La revisión con el Especialista designado indicará si se debe aplicar un mediador o no.
- 19.1.15 Se debe reducir al mínimo el número de archivos de transformaciones en los flujos BPEL, es mejor realizar una asignación directa.
- 19.1.16 Se debe validar los .xsd de entrada y de salida de los mensajes y siempre se debe devolver un error genérico de schema.
- 19.1.17 Los documentos de Diseño del servicio debe ser revisado y aprobados antes de empezar el desarrollo en los cuales se debe indicar los nombres de los métodos descripción de cada operación y solo los parámetros de entrada y salida de cada operación no es necesario ni útil indicar todo el modelo de negocio si no se lo utiliza.
- 19.1.18 El tiempo de revisión para los fuentes es de 3 días, luego de que el desarrollo esté terminado todo problema o cambio que se solicite debe ser ejecutado ya que es para prevenir problemas en producción.
- 19.1.19 Se ha implementado un componente de reversos General, el mismo que sirve para reversar transacciones de pago o reverso de las aplicaciones SOA y en los proxys del OSB. No pueden crearse esquemas de reverso independientes.
- 19.1.20 Se debe fomentar y reutilizar los componentes ya desarrollados en el ESB, de modo que se mantengan los flujos con un estándar que sea fácil de mantener y realiza cambios o mejoras.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 26 de 37

20. Restricciones de diseño

a. Bases de datos

Cualquier sugerencia o duda, petición de reunión, reporte de incompatibilidad del documento con las directrices corporativas, o petición de adaptación a casos concretos de los estándares definidos en este documento que aplique como una restricción deberá dirigirse a Subgerencia de Arquitectura y Soporte de TI.

b. Aplicaciones

Cualquier sugerencia o duda, petición de reunión, reporte de incompatibilidad del documento con las directrices corporativas, o petición de adaptación a casos concretos de los estándares definidos en este documento que aplique como una restricción deberá dirigirse a Subgerencia de Arquitectura y Soporte de TI.

c. Integración

Cualquier sugerencia o duda, petición de reunión, reporte de incompatibilidad del documento con las directrices corporativas, o petición de adaptación a casos concretos de los estándares definidos en este documento que aplique como una restricción deberá dirigirse a Subgerencia de Arquitectura y Soporte de TI.

21. Entorno Metodológico

Para los aspectos no definidos en el presente documento, en lo referente a metodología, se tomarán como referencia los siguientes marcos de trabajo:

Para proyectos enmarcados en la gestión tradicional


- Prácticas y recomendaciones de **Project Management Body of Knowledge (PMBOK)**.
- **Rational Unified Process (RUP)**, para el control del ciclo de vida del proyecto
- **Mejores prácticas ITIL**, para las actividades de gestión de servicio y puesta en marcha de nuevos servicios.

Para proyectos enmarcados en la gestión Ágil.

- Prácticas y recomendaciones de **Agile Lean o Scrum**.
- **Rational Unified Process (RUP)**, para el control del ciclo de vida del proyecto.
- **Mejores prácticas ITIL**, para las actividades de gestión de servicio y puesta en marcha de nuevos servicios.

A nivel metodológico, los equipos de desarrollo podrán añadir cualquier práctica o técnica contemplada en cualquiera de estos marcos de trabajo, siempre y cuando no entre en conflicto con lo establecido como norma.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 27 de 37

22. Política de Versionado

El patrón de nombrado de la versión es Vxx.yy.zzz

Por ejemplo: V01.04.010, (la tabla siguiente indica el valor de cada uno de ellos) donde: VARIABLE	POSIBLES VALORES	SIGNIFICADO
xx	Entero positivo. [1..99]	Número de versión mayor. Aumenta cuando se añaden cambios significativos y/o nuevas funcionalidades. Fundamentalmente cuando se inicia un nuevo desarrollo (o pliego)
yy	Entero positivo. [1..99]	Número de versión menor. Aumenta cuando se añaden pequeños cambios y/o funcionalidades poco importantes. A modo de ejemplo, una forma de identificar el cambio de funcionalidad si conlleva una generación de nuevos scripts de pruebas. Mantenimiento adaptativo de menor importancia.
zzz	Enteros positivos. [0..999]	Número de revisión. Aumenta con cada una de las entregas de mantenimiento e incluyen resoluciones de errores.

23. Política de certificados.

En este apartado se va a tratar todo lo relativo a la política de certificados digitales en las plataformas de integración.

El certificado digital tiene como función principal:

- Autenticar la identidad del usuario, de forma electrónica, ante terceros.
- Firmar electrónicamente de forma que se garantice la integridad de los datos transmitidos y su procedencia. Un documento firmado no puede ser manipulado, ya que la firma está asociada matemáticamente tanto al documento como al firmante
- Cifrar datos para que sólo el destinatario del documento pueda acceder a su contenido

Certificados admitidos


Los certificados deben ceñirse a una lista de autoridades emisoras de certificados digitales reconocidas.

En el caso de la institución las empresas que se usan en los diferentes ambientes son

1. Symantec
2. Digicert

En el caso de ser ambientes de desarrollo pueden ser utilizados certificados autofirmados con la aprobación del área de seguridad informática.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 28 de 37

Los certificados proporcionados por las empresas proveedoras de servicios, empresas estatales o privadas deben proporcionar los certificados necesarios y en los formatos que admiten las aplicaciones de la institución.

En este caso los ambientes de desarrollo deberán utilizar los certificados que sean equivalentes a los que serán utilizados en producción, para poder evaluar el proceso de instalación de los mismos en un ambiente controlado.

Organización de los certificados

Tanto para el caso de Web Services, como en el caso de aplicaciones web, cuando se vea la necesidad de utilizar certificados digitales para autenticación o cifrado de los datos, se ha resuelto organizar la infraestructura necesaria de la siguiente manera:

- Por cada uno de los aplicativos o servicios, se ha de crear un almacén de certificados para cada entorno. Estos almacenes contendrán:
 - ☒ Certificado servidor.
 - ☒ Certificados de la jerarquía de las Autoridades de Certificación de los distintos certificados cliente y servidor.
 - ☒ En el caso de Web Services, certificados con clave pública de cliente, enviados por las unidades, empresas, organizaciones, etc. consumidoras de la aplicación.

Considerar que puede crear un almacén de certificados para cada aplicativo, pero se puede reutilizar un mismo almacén de certificados para dos o más aplicativos, teniendo en cuenta que éstos deberán tener autorizados los mismos usuarios.

El formato de almacenamiento de los certificados debe ser JKS.

Obtención de Certificados a clientes de servicios o aplicativos web

Para las tareas de desarrollo o certificación el área responsable del aplicativo en Tecnología dispone de certificados de pruebas, así como la posibilidad de crear certificados de servidor de tipo autofirmado.

Para el uso en producción, debe ceñirse a los procedimientos del área de seguridad informática, dependiendo del escenario de uso de la aplicación o servicio web

Entorno de Desarrollo


Como norma general se utilizará un certificado cliente para acceder a cada aplicativo que necesite certificados para sus operaciones. Es decir, si un aplicativo es utilizado por varios clientes, se utilizará el mismo certificado.

Como certificado de servidor se dispone de un certificado para el entorno de desarrollo emitido por el área de Infraestructura.

Será responsabilidad del aplicativo:

Incorporar entre sus ficheros de recursos (dependientes del entorno) el almacén con los certificados en que confía dicha aplicación, así como el certificado de autenticación o firma que debe utilizar.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 29 de 37

Mantener el inventario de los certificados a su cargo, llevando el control de al menos los siguientes datos: Aplicativo dueño del certificado, aplicación que lo usa, fechas de caducidad, entidad certificadora.

Mantener actualizado dicho almacén de certificados ante caducidad o cambio de los mismos, así como solicitar su subida a los entornos de desarrollo, preproducción y producción con la suficiente antelación para minimizar las paradas de servicio.

Configurar adecuadamente la conexión SSL haciendo uso de la plataforma designada, para el efecto. Las aplicaciones en su desarrollo deben contemplar el tratamiento de excepción en caso de recibir cualquier tipo de estos certificados.

Se sugiere que en caso de no aceptar alguno de ellos, así como para cuando el usuario no dispone de certificado o cancela la negociación (error 403), se deberá redirigir al usuario a una página HTML de aviso donde se indique la necesidad de disponer de un certificado cliente y los tipos aceptados por el aplicativo.

Es responsabilidad del aplicativo verificar si la petición llega con certificado cliente y enviar si la persona/entidad asociada a dicho certificado está autorizada o no para el acceso al aplicativo.

El requisito de seguridad (comunicación SSL y certificado digital del usuario requerido) debe establecerse para toda la aplicación.

24. Política de Documentación.

Documentación

	Entregable	Localización
Análisis	Maqueta (Mokup)	/Documentación/
Análisis	Interfaz de Servicio	/Documentación/
Análisis	Requisitos - Casos de uso	/Documentación/
Análisis	Modelado	DES_REPO/<Nombre Aplicación> DES_REPO/<Nombre servicio>
Diseño	Diseño de Arquitectura	DES_REPO/<Nombre Aplicación> DES_REPO/<Nombre servicio>
Diseño	Modelado Aplicación/Servicio	DES_REPO/<Nombre Aplicación> DES_REPO/<Nombre servicio>
Diseño	Modelado de DATOS (Aplicativo, de Negocio)	DES_REPO/<Nombre Aplicación> DES_REPO/<Nombre servicio>
Implementación	Despliegue de procesos BATCH	/Documentación
Implementación	Plantilla 13. Manual Técnico de Despliegue de Aplicación Web	/Documentación
Implementación	Plantilla 14. Publicación de Servicios Web	/Documentación
Implementación	Plantilla 20. Plan de Implantación General	/Documentación



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 30 de 37

Implementación	Plantilla 24. Despliegue aplicaciones SOA-BPM.dot	DES_REPO/<Nombre Aplicación> DES_REPO/<Nombre servicio>
----------------	---	--



Carpeta Despliegues

En la siguiente imagen se muestra el contenido de la carpeta despliegues de un modo grafico.


Ilustración 4. Estructura de la Carpeta Despliegues

Dependiendo de los elementos que contenga la aplicación podremos incluir en la carpeta Despliegues:	Entregable	Localización
FASE		
Para aplicaciones J2EE		
Aplicación	<acrónimo de la aplicación>.ear (Aplicativo .EAR)	trunk/Despliegues/Aplicación
Aplicación	<acrónimo de la aplicación>_estaticos.zip (Archivo de ficheros estáticos)	trunk/Despliegues/Aplicación
Aplicación	<acrónimo de la aplicación>_recursos.zip (Archivo de ficheros de propiedades)	trunk/Despliegues/Aplicación/<Entorno> (desa, pre, pro, forma)
Para aplicaciones Con BPM		
BPM	<acrónimo de la aplicación>.ear (Aplicativo .EAR)	trunk/Despliegues/BPM
BPM	Otros ficheros	trunk/Despliegues/BPM /<Entorno> (desa, pre, pro, forma)
Para aplicaciones Con SOA		
SOA	<nom_fichero>.ear	trunk/Despliegues/SOA
Otras actuaciones SOA + BPM		
Modificación fichero CDQP	CDQP.txt	Trunk Despliegues BPM + SOA
Creación de Colas IPE-EMS		Dentro de documentación indicar nombre de la cola a crear
Para aplicaciones Con Gestor Documental		
Gestor Documental	<Ficheros>.dql	trunk/Despliegues/Gestor Documental
Extensión de las DFCS	<Ficheros>.tbo	trunk/Despliegues/Gestor Documental
Para aplicaciones con BBDD		
BBDD	<Ficheros>.sql	trunk/Despliegues/BBDD

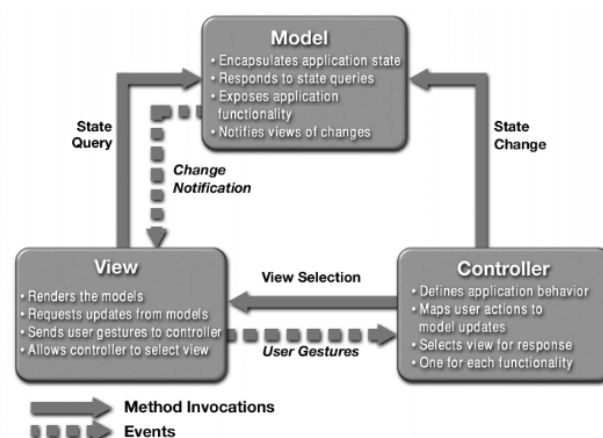
Fase de Análisis

Interfaz de Servicios Web.

Para Servicios Web, deberá construirse en la herramienta EA y elaborar un documento con las especificaciones del servicio, del mismo modo que con las Mokups, se crearán cuantos documentos de Interfaz sean necesarios, para comprensión del requerimiento.

	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 33 de 37


Seguirá el patrón de diseño MVC



- **Presentación:**
Proporciona el interfaz de usuario (IU) y gestiona el flujo de navegación de la aplicación. Gestiona los eventos y el intercambio de datos entre el usuario y la aplicación.
- **Lógica de Negocio:**
Proporciona la lógica de la aplicación y funciona como nexo entre las capas de presentación y persistencia.
En un servicio o componente de negocio distinguimos los siguientes elementos:
 - Interface del servicio proporcionado a la capa de presentación. Estos elementos son el contrato de la capa de negocio, es decir, son los elementos que indican lo que va a hacer la capa.
 - Implementación del servicio o componente de negocio que contiene la lógica de la aplicación. Deben implementar los elementos de la interfaz de negocio.
- **Persistencia:**
Incluye las entidades persistentes del dominio de la aplicación y los servicios de acceso a la base de datos.
En la capa de persistencia podemos distinguir los siguientes elementos:
 - Interface de persistencia proporcionado a la capa de negocio. Estos elementos son el contrato de la capa de persistencia, es decir, son los elementos que indican lo que va a hacer la capa.
 - Implementación de la persistencia. Estos elementos son los encargados reales de realizar la persistencia. Deben implementar los elementos de la interfaz de persistencia.
- **Servicios transversales:**
Los componentes comunes que proporcionan funcionalidades reutilizables por las diferentes aplicaciones.
Se podrá utilizar el conjunto de servicios proporcionadas y soportados por el servidor de aplicaciones (JavaMail, JNDI, JMS, etc...)

Además debemos considerar los siguientes aspectos relevantes:

- La seguridad de la aplicación se realizará del siguiente modo:

	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 34 de 37

La seguridad de la aplicación web/servicio web debe ser validada con el área de Seguridad Informática. Otros casos de cambios de seguridad, deberá ser tratada y aprobada en el Comité de Arquitectura.

La gestión de las transacciones debe considerarse en cada una de las capas. Será responsabilidad de cada aplicación realizar una adecuada gestión transaccional que evite las inconsistencias en los datos de la aplicación.

Los procesos batch son ejecutados fuera de las plataformas de integración y su programación puede planificarse con la frecuencia que se determine o lanzarse de forma manual desde un equipo cliente.

Arquitectura de Integración (estándar)

Los estándares a considerar están basados en los soportados por la especificación J2EE 1.4. Pese a eso no se permite de forma general usar:

- Enterprise Java Beans
- RMI-IIOP
- ORB
- JAVA-IDL
- JDBC
- JMX
- Uso programático del “User Transaction”

Arquitectura en alta disponibilidad

Hay que tener en cuenta que el despliegue de las aplicaciones debe acomodarse a esta arquitectura y que las necesidades de alta disponibilidad van a incidir directamente en la forma de implementar los componentes java.


En particular, se recomienda tener en cuenta que los servidores de aplicaciones Oracle, se encuentran en **CLUSTER** y por ello se debe seguir las recomendaciones de desarrollo en estos entornos, entre las que podemos destacar:

- Todo lo que esté incluido en la sesión debe ser serializable.
- El tamaño de la sesión debe reducirse al mínimo posible.
- No use el sistema de archivos local de la máquina. Usar un sistema centralizado como Base de Datos o NAS.
- Previo al paso a producción, revalidar los parámetros de configuración del cluster.

En la fase de diseño de la aplicación se deben indicar los distintos dispositivos (servidores web, servidores de base de datos, servidores de aplicaciones, etc.) de los que va a hacer uso la aplicación.

En la herramienta EA, hay diagramas de la Arquitectura de despliegue, No se deben agregar los dispositivos de los que no haga uso la aplicación y crear solo los artefactos propios de la aplicación indicando donde son desplegados.



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 35 de 37

25. GLOSARIO

Plataforma de integración

IBM: Software base de integración, CSP, CTS

Oracle: Software base de integración, ESB

CSP: COBIS Service Processor.

CTS: COBIS Transaction Services.

ESB: Enterprise Service Bus.

Arquitectura Orientada a servicios ver SOA

SOA (siglas del inglés Service Oriented Architecture) está pensado para diseñar y desarrollar sistemas distribuidos. Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio, disminuyendo los costos de implementación y estimulando la innovación de servicios a clientes mediante una adaptación ágil y flexible ante los rápidos cambios del mercado.

Oracle Service Bus transforma arquitecturas complejas y débiles, en redes de integración ágiles y robustas, mediante la conexión, la mediación, y la gestión de las interacciones entre servicios y aplicaciones. Oracle Service Bus ofrece un bajo coste de integración basada en estándares, rendimiento extremo y escalabilidad.

Servicio de negocio es una funcionalidad de negocio disponible en nuestros sistemas y que se caracteriza por disponer de un conjunto de operaciones con parámetros definidos de entrada y salida.

Contrato de servicio garantiza la interoperabilidad y facilita el desarrollo de los servicios, debe ser independiente de la tecnología en que se implemente el servicio, se debe tener las siguientes características:

- Operaciones
- Parámetros de entrada
- Parámetros de salida
- Formato de mensajes


Proveedor de servicio, será aquella aplicación o sistema que publica un conjunto de sus funcionalidades como servicio disponible para otros sistemas.

Consumidor aquella aplicación o sistema que utiliza un servicio de negocio.

WSDL, describe el mensaje que un servicio puede recibir y enviar. Es el más básico lenguaje de “contrato” utilizado para describir la funcionalidad ofrecida por un servicio.

WS_Policy, describe la calidad de las características del servicio y los requerimientos asociados a este servicio. Típicamente las políticas describen los requerimientos de seguridad de un servicio, la optimización soportada por un servicio tal como MTOM y si un servicio utiliza WS-ReliableMessaging. Oracle Web



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 36 de 37

Services Policy Manager provee las herramientas para construir y apoyar las políticas para Oracle Fusion Middleware.

WS-MetadataExchange es un protocolo de “handshake” que permite al usuario el obtener documentos WSDL y WS_Policy asociados a un servicio.

UDDI es un modelo utilizado para el registro de los servicios web. Provee un repositorio común de metadata relacionado a los servicios que pueden ser utilizados para descubrir que servicios están disponibles y para seleccionar los servicios que están disponibles para ser utilizados para la construcción de nuevos servicios “compuestos” y procesos de negocio.

Oracle Service Registry representa la implementación de UDDI bajo Oracle Fusion Middleware.

JAX-RPX y JAX-WS son los estándares definidos por la comunidad de Java que describe como los desarrolladores de Java pueden crear servicios Web. JAXRPC es el estándar que es utilizado en J2EE 1.4; JAX-WS, como muchos otros de las nuevas especificaciones de JavaEE eliminan mucha de la complejidad asociada con el desarrollo de servicios Web. Ejemplo de esto es que el JAX-WS provee un modelo simple para la implementación de la lógica del negocio y exponen el “contrato” como una interfaz WSDL utilizando anotaciones en el código implementado. Al apalancarse en JAX_WS, los desarrolladores pueden construir servicios portables.

26. DOCUMENTOS DE REFERENCIA


Normativa

1. ISO 9001:2000. Sistemas de Gestión de la Calidad Fundamentos y Vocabulario.
2. Manual de Políticas Arquitectura Tecnológica.

27. ANEXOS

1. Anexo A1.- Estándar de Diseño Infraestructura.
2. Anexo A2.- Estándar de Diseño Integración.
3. Anexo B1.- Estándar de Diseño XSD.
4. Anexo B2.- Estándar de Diseño XLST.
5. Anexo B3.- Estándar de Diseño Mediadores.
6. Anexo C1.- Estándar de Desarrollo JEEE
7. Anexo C2.- Estándar de Desarrollo Composite
8. Anexo C3.- Estándar de Desarrollo Proxys
9. Anexo C4.- Estándar de Desarrollo Business Service
10. Anexo C5.- Estándar de Desarrollo XQuery y funciones.
11. Anexo C6.- Estándar de Configuración Adapters
12. Anexo C7.- Estándar de uso de Servicios Utilitarios.
13. Anexo D1.- Documento de control de revisión de diseño de servicios
 - a. Como referencia, este documento lo maneja el área de Arquitectura para control



	Desarrollo en Plataforma Bus Empresarial		
	Estándares	ARQT-EST-001	
	ARQUITECTURA Y SOPORTE DE TI	Versión: 2.0	Página 37 de 37

14. Anexo D2.- Documento de Especificación de Servicio
a. Este documento es un ejemplo del contenido.

28.CONTROL DE CAMBIOS

Versión	Componente estructura del documento	Fecha	Observaciones	Responsable
1	Documento original	05/11/2008		Jefe Departamental de Arquitectura y Soporte IT Subgerente de Ingeniería de Sistemas
2	Migración al nuevo formato de procedimientos según validación del Departamento de Gestión de Calidad y Procesos.	21/11/2014		Subgerente de Arquitectura y Soporte de TI
3	Cambio de nombre al procedimiento para cubrir todas las plataformas de integración actuales. modificación del contenido para ser más coherente a las plataformas actuales	04/09/2015		Subgerente de Arquitectura y Soporte de TI
4	Reestructuración del documento de estándares separando elementos de: <ul style="list-style-type: none"> Diseño Desarrollo Configuración Documentación 	04/10/2016		Subgerente de Arquitectura y Soporte de TI

