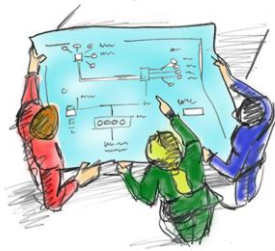




Universidad Católica de Santiago del Estero
Departamento Académico Rafaela

Gestión de Procesos de Negocios



Docentes:

- Ing. Lorena D'Iorio
- Ing. Ariel Rossanigo
- Ing. Román Zenobi

Bibliografía

- ▶ Villareal, Pablo: *Apuntes cátedra Gestión e Integración de Procesos de Negocio en Empresas y en Ambientes Business-to-Business*, EMISI, UTN, Santa Fe 2011.
- ▶ Erl, Thomas. *SOA Principles of Service Design*. Prentice Hall PTR, 2007.
- ▶ Mike P. Papazoglou: *Service-Oriented Computing: Concepts, Characteristics and Directions*. WISE 2003: 3-12.
- ▶ Papazoglou, M.P. (2006): *Web Services Technologies and Standards*. ACM Computing Surveys.

Unidad 6: : Arquitecturas Orientadas a Servicios – SOA

AGENDA:

- Paradigma de Orientación a Servicios
- Arquitectura Orientada a Servicios (SOA)
- Diseño Orientado a Servicios

Servicio

Es un programa o aplicación de software independiente.

- Que soporta los principios de la computación orientada a servicios.
- Que provee una colección de funciones (operaciones) agrupadas de acuerdo a un contexto funcional establecido por el servicio.

Servicios son elementos computacionales autónomos e independientes de la plataforma.

- Que se describen por sí mismo.
- Ejecutan funciones de negocio, las cuales van desde responder simples solicitudes hasta ejecutar procesos de negocio complejos.
- permiten a las organizaciones exponer sus funcionalidades descritas en un contrato.
- Independientes del estado o contexto de otros servicios.

Computación Orientada a Servicios (Orientación a Servicios)

Computación Orientada a Servicios

- *Paradigma de computación que usa a los servicios como los bloques de construcción fundamentales de una aplicación o sistema (distribuido).*
- *Comprende una serie de principios de diseño orientados a servicios.*
- *Evolución de la computación distribuida basada en el paradigma de interacción solicitud/respuesta y los paradigmas de orientación a objetos y componentes.*

Servicios

*Un
Servicio
incluye
los
siguientes
aspectos:*

Un contrato.

Una implementación del servicio.

Detalles de acceso.

Servicios

Contrato

- Describe el comportamiento del servicio.
- Es independiente de los detalles de implementación.

Incluye

- **Interfaz del Servicio:**
 - El conjunto de **operaciones** que provee el servicio (las funciones que provee y hace públicas un servicio para que sean invocadas).
 - Una operación representa una **unidad lógica de trabajo**.
 - Una operación posee **parámetros de entrada y salida**. → Los mensajes intercambiados entre un solicitador del servicio y el servicio.
 - La interfaz también puede definir **el formato** de los mensajes.
- **Calidad del Servicio:**
 - Características que no involucran detalles de acceso o de la interfaz del servicio.
 - Ejemplo: detalles de autorización, seguridad, disponibilidad.

Servicios

Implementación del Servicio

- La lógica de negocio que soporta el servicio implementada en una tecnología específica.
- Servicios Web, J2EE, .Net

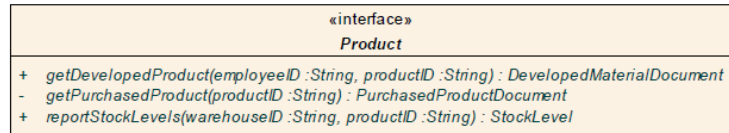
Detalles de Acceso:

- **Ubicación:** la dirección de red donde reside el servicio (conocido como el **EndPoint**)
- **Binding:** indica el protocolo de transporte a usar en la comunicación

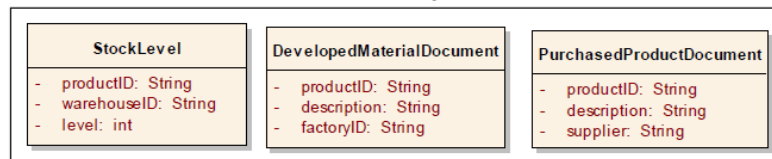
Servicios

Ejemplo: *Diseño del servicio de la entidad de negocio Producto*

Contrato: Interfaz del Servicio



Estructura de Datos de los Mensajes del Servicio

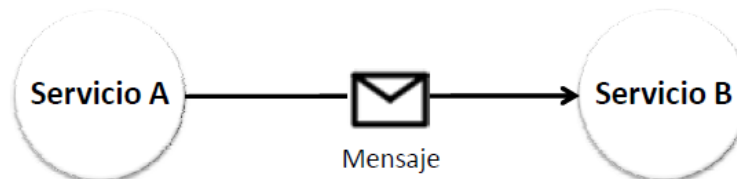


Servicios



Comunicación entre servicios

- Los servicios se comunican a través del intercambio de mensajes
- Una operación de un servicio puede tener mensajes de entrada y/o salida



Servicios

Patrones de interacciones (tipos de operaciones del servicio):

One-way:

- *Involucra un único mensaje en un sólo sentido*
- *Un cliente invoca un servicio con una solicitud (enviando un mensaje de entrada)*
- *No existe una respuesta a la solicitud*
- *Interacción asíncrona*

Request-response:

- *Involucra el intercambio de dos mensajes*
- *Un cliente invoca un servicio con una solicitud y una respuesta es retornada al mismo*
- *Interacción síncrona*

Servicios

Patrones de interacciones (tipos de operaciones del servicio):

Solicit-response:

- *Involucra el intercambio de dos mensajes*
- *El servicio envía una notificación a un cliente y recibe una respuesta*
- *Interacción síncrona*

Notification:

- *Involucra un único mensaje*
- *El servicio es quien envía el mensaje*
- *Interacción asíncrona*

Servicios

Servicio Compuesto

- Es una agregación de servicios simples y compuestos.

Servicio Compuesto

- Es aquel cuya lógica de negocio involucra la invocación de operaciones ofrecidas por otros servicios

Un servicio compuesto está típicamente asociado con la automatización de un proceso de negocio, a través del cual se integran las aplicaciones de la empresa

Servicio Simple

- Es aquel cuya lógica interna es implementada accediendo a un sistema de información local

Principios de Orientación a Servicios

Servicios comparten un contrato:

- Para la interacción entre servicios, todo servicio debe definir y compartir un contrato que describe el servicio y los términos del intercambio de mensajes.
- Un consumidor de un servicio, accederá a este mediante el contrato, logrando así la independencia entre el consumidor y la implementación del propio servicio.

Servicios débilmente acoplados:

- Servicios deberían ser diseñados para interactuar sin altas dependencias entre servicios.
- Lógica de los servicios tiene que ser independiente de los demás servicios tanto como sea posible.

Abstracción de Servicios

- Ocultamiento de los detalles de implementación e infraestructura tecnológica del ambiente de ejecución.

Principios de Orientación a Servicios

Reuso de Servicios:

todo servicio debe ser diseñado y construido pensando en su reutilización en diferentes contextos.

Es un principio fundamental en orientación a servicios.

La lógica de servicios reusables necesita ser agrupada dentro de un contexto agnóstico.

Servicios con contextos funcionales agnósticos proveen un mayor potencial de reuso.

Autonomía de Servicio:

Se refiere al grado de libertad que un componente de software posee para controlar y tomar sus propias decisiones sin tener que recurrir a un componente externo.

Cuanto mayor control un programa posee, mayor autonomía ofrece.

Cada servicio puede tener su propio entorno de ejecución.

Principios de Orientación a Servicios

Servicios sin Estado (Stateless):

- Servicios simples no deberían requerir mantener información o estado entre solicitudes de sus funciones (operaciones) cuando son implementados.
- Servicios simples no deberían ser dependientes del estado o contexto de otros servicios.
- Si se requiere mantener el estado entre solicitudes, estas deberían mantenerse en términos de procesos de negocio (Servicios Compuestos) o servicios de entidades.

Descubrimiento de Servicios:

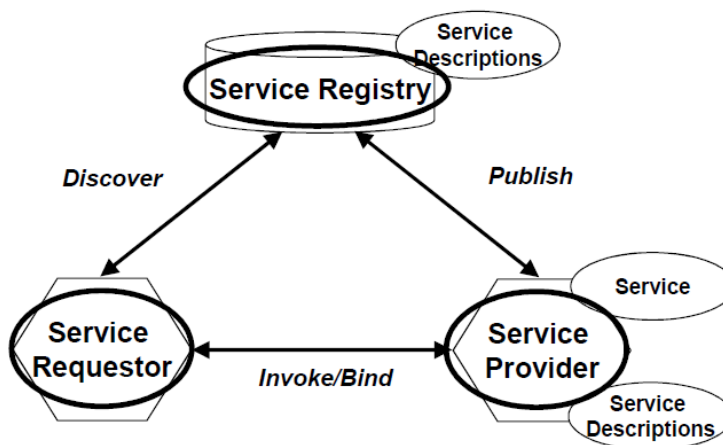
- Todo servicio debe poder ser descubierto de alguna forma para que pueda ser utilizado.

Arquitectura Orientada a Servicios

- *Estilo arquitectónico basado en la computación orientada a servicios.*
- *Establece una forma de organizar y diseñar un sistema de software para proveer servicios a aplicaciones de usuarios finales u otros servicios distribuidos en la red, a través de interfaces publicadas que pueden ser buscadas y encontradas.*
- *Como filosofía de diseño, es independiente de la tecnología (Ejemplos: servicios Web o Enterprise Java Beans).*

Arquitectura Orientada a Servicios

Roles



Arquitectura Orientada a Servicios

Roles (Tipos de Componentes):

Service Provider:

- La parte que provee funcionalidades de aplicaciones y negocio como servicios.
- Publica servicios.
- **Punto de vista de negocio:** el owner del servicio.
- **Punto de vista arquitectónico:** el componente que implementa el servicio.

Service Requestor:

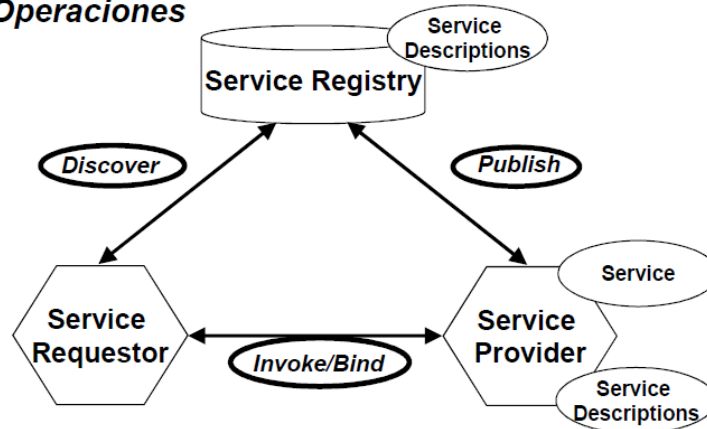
- La parte con un requerimiento que puede ser cumplido por un servicio.
- **Punto de vista de negocio:** quien requiere una función de negocio.
- **Punto de vista arquitectónico:** la aplicación que busca e invoca servicios.

Service Registry/Broker:

- La parte que provee un repositorio de descripciones de servicios donde proveedores de servicio publican servicios y solicitadores encuentran y obtienen información a través de los mismos.

Arquitectura Orientada a Servicios

Operaciones



Arquitectura Orientada a Servicios

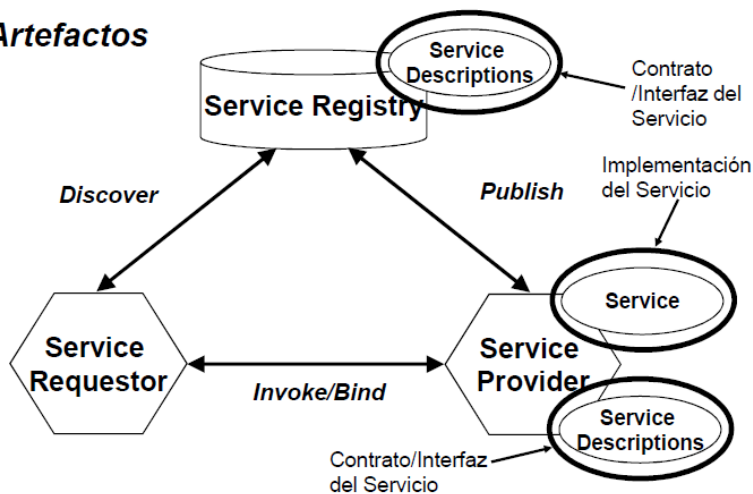
Operaciones esperadas en SOA:

- Creación de servicios
- Descripción de servicios
- Publicación de servicios en repositorios (Intranet o Internet)
- Descubrimiento de servicios por potenciales usuarios
- Invocación/binding de servicios
- Eliminación de la publicación cuando no esté mas disponible o no se requiera

Gestión de Procesos de Negocios – UCSE– DAR – 2012 21

Arquitectura Orientada a Servicios

Artefactos



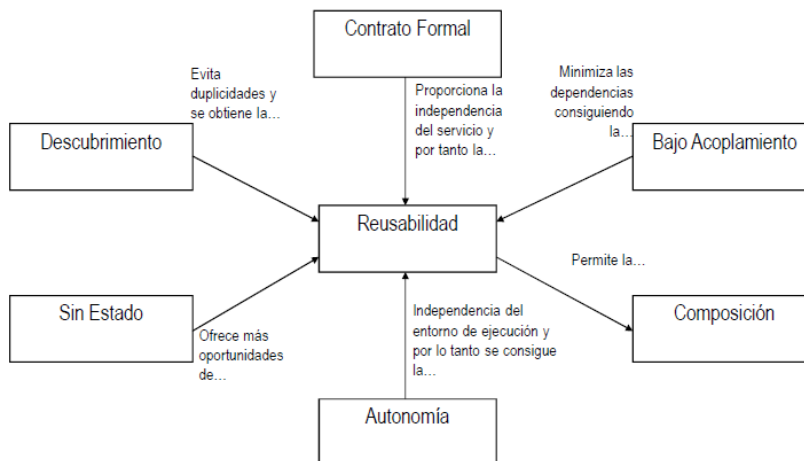
Gestión de Procesos de Negocios – UCSE– DAR – 2012 22

Arquitectura Orientada a Servicios

Beneficios:

- Encapsulación y modularización de componentes de software autónomos, heterogéneos y débilmente acoplados:
 - *Servicios son definidos, desarrollados y gestionados de manera independiente.*
- Reuso de servicios en diferentes aplicaciones.
- Incremento de la Interoperabilidad:
 - *Menores requerimientos de integración*
- Alineación entre los procesos de negocio y las aplicaciones:
 - *Alineación de los procesos de negocio (requerimientos de negocio) con los requerimientos de software/sistemas.*
 - *Facilita la integración de aplicaciones para la automatización de proceso.*
- Alineación de la solución tecnológica con la solución de negocio:
 - *Servicios en niveles mayores de abstracción y de granularidad, representando lógica o tareas de negocio.*
 - *Un servicio encapsula un segmento o funcionalidad de la empresa.*
- Agilidad ante cambios organizacionales
 - *Reuso de servicios independientes de los procesos de negocio y aplicaciones específicas.*
 - *Reemplazo de servicios.*
 - *Cambio en la lógica sin implicar cambios en el contrato.*

Interrelación entre los Principios de Orientación a Servicios



Diseño Orientado a Servicios

Clasificación de Servicios

Los servicios pueden ser clasificados según:

- El tipo de lógica que encapsulan.
- El potencial reuso que dicha lógica tiene.
- Cómo esta lógica se relaciona con dominios existentes dentro de una empresa.

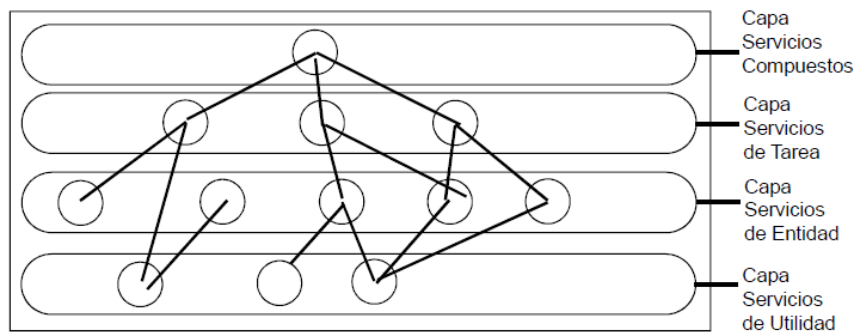
Tipos de Servicios:

1. **Servicios de Entidad**
2. **Servicios de Utilidad**
3. **Servicios de Tareas**
4. **Servicios de Integración/Composición**

Gestión de Procesos de Negocios – UCSE- DAR – 2012 25

Diseño Orientado a Servicios

Capas de abstracción de tipos de servicios



Estas capas tienden a formar una jerarquía de Composición.

Gestión de Procesos de Negocios – UCSE- DAR – 2012 26

Diseño Orientado a Servicios

Servicio de Entidad (Entity Service)

- *Define su funcionalidad y contexto sobre una o más entidad de negocio.*
- *Entidades de negocio se pueden derivar de modelos de dominio, modelos de datos, ontologías, etc.*
- *Ejemplos de Entidades: Cliente, Empleado, Factura, Orden, Producto, etc.*
- *Independiente de la lógica de los procesos de negocio.*
- *Es altamente reusable en diferentes procesos de negocio o composiciones de servicios.*
- *También conocido como: entity-centric business services, business entity service.*

Diseño Orientado a Servicios

Servicio de Tarea (Task Service)

- *Un servicio con funcionalidades directamente asociadas a tareas de negocio.*
- *Provee funciones o capacidades de negocio básicas que soportan tareas específicas en el contexto de procesos de negocio.*
- *Generalmente tiene un alcance funcional a una entidad de negocio.*
- *Ejemplos: Servicio de Gestión Pago, Servicio de Gestión de Facturación, Servicio de Gestión de Inventario:*
 - *Cada uno de ellos brindando operaciones (definidas en su interfaz) y expresando funciones de negocio.*
- *Independiente de los procesos de negocio.*
- *Puede ser reusado en diferentes procesos de negocio o composiciones de servicios, pero el nivel de reuso generalmente es menor al de un servicio de entidad.*
- *También conocido como: business services, task centric business service.*

Diseño Orientado a Servicios

Servicio de Integración/Composición

- *Servicio compuesto que expresa la lógica de un proceso de negocio, a través del cual se invoca a otros servicios.*
- *Ejemplos: Servicio de Gestión de Ordenes de Compra, Servicio de Gestión de Reclamos.*
- *Es aplicable al contexto de un proceso de negocio.*
- *Poca probabilidad de reuso.*
- *También conocido como: orchestration services, business process service.*

Diseño Orientado a Servicios

Servicio de Utilidad (Utility Service)

- *Los anteriores tipos de servicios se enfocan en la lógica de negocio.*
- *Servicios de Utilidad proveen funciones y capacidades independientes de la lógica de negocio.*
- *Ejemplos: Servicio de Loggin, Notificación, Manejo de Excepciones.*
- *Altamente reusable en diferentes contextos.*
- *También conocido como: application service, infrastructure service.*

Diseño Orientado a Servicios

Identificación de Servicios

Enfoque Top-Down

- Servicios son identificados como actividades de los procesos de negocio.
- Enfoque centrado en procesos → requiere partir de modelos de procesos de negocio.
- Se identifican primero los servicios compuestos que soportan los procesos de negocio.
- Luego se identifican servicios de tareas que soportan actividades de los procesos.
- Luego se identifican servicios de entidades y de utilidad.

Diseño Orientado a Servicios

Identificación de Servicios

Enfoque Bottom-Up

- Servicios son identificados a partir de las aplicaciones existentes.
- Se identifican las funcionalidades de las aplicaciones a ser expuestas en los servicios que las encapsulan.
- Las funcionalidades expuestas de los sistemas existentes son generalmente orientadas o dependiente de la tecnología → no soportan actividades o funciones de negocio.
- Estos servicios deben ser combinados para dar soporte a los procesos de negocio.

Diseño Orientado a Servicios

Identificación de Servicios

Se utiliza y recomienda usar una mezcla de ambos enfoques: Top-Down y Bottom-Up

- Comenzar con identificar los servicios compuestos que soportan a los procesos.
- Identificar servicios de tarea.
- Identificar servicios de aplicaciones existentes. Mapearlos con actividades requeridas en los procesos.
- Identificar el resto de los servicios.

Diseño Orientado a Servicios

Tipos de Granulación (Granularidad)

Granulación de Servicio:

- La granulación del alcance funcional del servicio (su contexto funcional).
- Refleja la cantidad de lógica funcional que podría encapsular, basado en su contexto.
- Ej: un servicio de granulación amplia (coarse grained) tendría un contexto funcional amplio, independientemente si provee una o diez funciones.

Granulación de Funciones (Operaciones):

- Representa el alcance funcional de una función (operación) específica de un servicio.
- Regla: una función de granulación fina (fine grained) realiza menos trabajo (o lógica de negocio) que una de granulación amplia.

Diseño Orientado a Servicios

Tipos de Granulación (Granularidad)

Granulación de Datos:

- La cantidad de datos que una función de un servicio necesita intercambiar para llevar a cabo su función.
- *Tipos de intercambios entre servicios*: mensajes centrados en documentos vs comunicación RPC.
- Mensajes centrados en documentos contienen conjuntos enteros de información (documentos de negocio) → **granulación amplia**.
- Comunicación RPC se basa en intercambios mas pequeños de parámetros de datos → **granulación fina**.

Diseño Orientado a Servicios

Tipos de Granulación (Granularidad)

Granulación de Restricciones:

- El nivel de detalle en que la lógica de validación está definida para un parámetro particular de una función o una función dentro del contrato del servicio.
- El esquema o modelo de datos de la información intercambiada por una función puede definir restricciones de validación específicas:
 - *Tipo, longitud y formato del dato, valores permitidos, etc.*
- Medida en relación a la lógica de validación presente sólo en el contrato del servicio, no la lógica de negocio de las funciones de un servicio.
 - *No existe un principio que dicte los niveles de granulación para un diseño orientado a servicios.*
 - *Varios principios de diseño impactan en forma diferente en los tipos de granulación.*

Principios de Diseño Orientado a Servicios

Diseño de contratos:

- *Contratos constituyen la base para la comunicación entre servicios y por lo tanto representan el elemento arquitectural mas importante en una SOA.*
- **Propósito:** *soportar nuevos requerimientos de automatización de negocio minimizando la cantidad de esfuerzo de desarrollo y priorizando la reutilización de servicios disponibles.*
- *Foco en la definición de la interfaz (operaciones) del servicio, los datos que intercambian el servicio y los aspectos de calidad de servicio.*

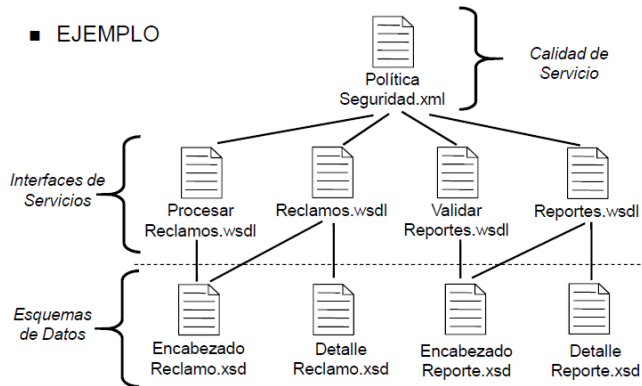
Principios de Diseño Orientado a Servicios

Diseño de contratos:

- *Evitar la transformación de datos siempre que sea posible en el intercambio de mensajes entre servicios.*
- *Transformación de datos → problemas:*
 - *Mayor esfuerzo de desarrollo en la integración.*
 - *Disminución de la performance debido a la ejecución del proceso de transformación de datos con cada intercambio.*
 - *Incremento de la complejidad arquitectural → incorporación de capas de transformación.*
 - *Necesidad de mantener y evolucionar las capas de transformaciones con el resto de la arquitectura.*
- *Dos servicios deberían utilizar el mismo esquema de datos en el intercambio de mensajes.*
- *Estandarización de la representación de los datos a ser utilizadas por todos los servicios.*
- *Reuso de esquemas de datos en diferentes servicios.*
- *Modularización y Centralización de las definiciones de calidad de servicio*
- *Reuso de dichas definiciones en diferentes servicios.*

Principios de Diseño Orientado a Servicios

Diseño de contratos:



Gestión de Procesos de Negocios - UCSE- DAR - 2012 39

Principios de Diseño Orientado a Servicios

Diseño de contratos:

Acoplamiento de Servicios

El acoplamiento siempre existe entre componentes de software

Propósitos:

- Servicios y sus consumidores deberían poder evolucionar en el tiempo con un mínimo impacto sobre cada uno de ellos.
- Reducir el número de conexiones entre servicios.
- Bajo nivel de acoplamiento entre servicios evita problemas de redundancia y duplicación de servicios.

Recomendación → Definir primero el contrato y luego la lógica de Implementación
Evitar acoplamiento del contrato con la implementación.

- Se produce cuando se deriva un contrato a partir de una implementación
- El contrato generado posee características del ambiente de implementación. Los detalles de los recursos a los que accede la implementación pueden estar embebidos dentro del contenido del contrato.
 - Ej: si el esquema de datos de los mensajes se deriva de tablas y vistas a las que accede la implementación, detalles del modelo físico de datos terminarán siendo parte del contrato del servicio.
- Esto disminuye el ciclo de vida del contrato del servicio y dificulta la evolución del mismo en el largo plazo.

Gestión de Procesos de Negocios - UCSE- DAR - 2012 40

Principios de Diseño Orientado a Servicios

Cohesión de Servicios:

Se refiere al grado de relación funcional que tienen las funciones (operaciones) que provee un servicio.

Es un principio importante en el diseño de servicios.

Alta **cohesión** en un servicio permite:

- *Aumento de la claridad y comprensión de las funciones del servicio.*
- *Simplifica el mantenimiento y futuros cambios.*
- *Granularidad del servicio en niveles razonables.*
- *Bajo acoplamiento.*

EJEMPLO:

Baja Cohesión

Interfaz Gestión Comercial

- cotizarOrden()
- generarFactura()
- crearOrden()

Alta Cohesión

Interfaz Gestión Factura

- generarFactura()
- ...

Interfaz Gestión Orden

- crearOrden()
- cotizarOrden()
- ...