**Gestión de procesos de negocio**

* Para administrar, optimizar y mejorar las actividades de negocio.
* Busca satisfacer los objetivos de la empresa.
* Mayor compatibilidad entre las actividades de la empresa y el sistema.
* Estrategia de mejoramiento continuo.
* Es la aplicación de métodos, técnicas y software para dar soporte al ciclo de vida. Involucran personas, organizaciones, aplicaciones, documentos, etc.
* Mejora la efectividad de la empresa.

**Proceso**

* Conjunto de actividades estructuradas y medibles que toman entradas y crean salidas.
* Actividad: tarea atómica.

**Proceso de negocio**

* Conjunto de actividades atómicas o compuestas ordenadas e interrelacionadas para alcanzar un objetivo de negocio.
* Define la estructura del flujo de trabajo.
* Conjunto de actividades 🡪 medibles, estructuradas, ordenadas, interrelacionadas, con comienzo y fin.

**Ciclo de vida**

* Modelamiento de los procesos
* Implementación
* Ejecución de procesos
* Control y gestión

**Pasos para GPN**

* Definir arquitecturas de procesos:
  + Entender la organización actual.
  + Identificar las cadenas de valor (conjunto de procesos) y los objetivos a alcanzar.
  + Identificar los procesos en cada cadena de valor. Para cada proceso:
    - Identificar y definir responsables.
    - Definir medidas de rendimiento según los objetivos.
    - Identificar problemas.
  + Definir la arquitectura de procesos para cada cadena de valor:
    - Identificar los procesos operativos
    - Definir la jerarquía de los procesos operativos
    - Identificar los procesos de soporte y de gestión.
    - Determinar el responsable de la gestión de cada proceso.
    - Definir los recursos para cada proceso.
    - Identificar los problemas de cada proceso.
* Rediseñar o mejorar los procesos:
  + Definir un proyecto de rediseño para un proceso específico.
  + Analizar el proceso.
  + Rediseñarlo.
  + Implementarlo.
  + Asignarlo al proyecto de IT para implementarlo.
* Implementar los procesos

**Sistemas de información orientados a procesos (SIOP)**

* Hacer que los sistemas se adapten a los procesos y no al revés.
* El sw se tiene que poder reusar y componer.
* Se tiene que poder adaptar el sw rápidamente.

**SIOP**

* Sistema de sw que administra y ejecuta procesos operacionales basándose en modelos de procesos.
* Los procesos involucran personas, aplicaciones y fuentes de información.
* El modelo es una estructura de un proceso. Se representa en un lenguaje visual. Se instancia muchas veces.
* Objetivo: Separar la lógica de aplicación de la lógica de proceso.
* Ventajas:
  + Facilita la comunicación entre el analista de negocios y el personal de IT.
  + Permite cambiar los procesos modificando solamente los modelos.
  + Control y monitoreo de los procesos y recursos.

**Desarrollo de SIOPs**

* Sistema para procesos específicos.
* Sistema genérico.
  + Configura el sistema de acuerdo a sus necesidades.

**Clasificación de SIOPs**

* Diseño vs implementación
  + BPMT: Business Process Modeling Tools 🡪 Para diseño/modelado de procesos.
  + WMS: Workflow Management Systems 🡪 Para implementación y ejecución.
  + PMT: Project Management Tools 🡪 Para todo.
* Personas vs aplicaciones de sw
  + Procesos persona a persona
    - Tareas no automatizables que requieren la intervención humana.
  + Procesos aplicación a aplicación
    - Sólo involucran tareas ejecutadas por sistemas de sw.
  + Procesos persona a aplicación
    - Involucra personas y tareas automatizadas. Para que trabajen de manera integrada.
* Estructurados vs desestructurados
  + No estructurado
    - No existe un modelo de proceso
  + Estructurado ad hoc
    - Modelo ejecutado pocas veces antes de ser descartado o modificado.
  + Débilmente estructurado
    - Modelo con un conjunto de restricciones.
  + Fuertemente estructurado
    - La ejecución del proceso sigue el modelo.
* Intra-organizacional vs inter-organizacional
  + Intra-organizacional
    - Personas y aplicaciones de una empresa.
  + Inter-organizacional
    - Personas y aplicaciones de diferentes empresas.

**Workflows**

* Automatización de un proceso de negocio en la que los documentos, información o tareas son pasados desde un participante a otro para su ejecución.
* Ejecución coordinada de múltiples tareas para cumplir un objetivo común.

**Especificación de workflow/proceso**

* Representación de un proceso de negocio en un formato que puede ser interpertado por un sistema de GPN para dar soporte a su ejecución.

**Gestión de workflow**

* Gestión, comunicación y control de las actividades realizadas por personas a través de aplicaciones.
* El propósito es soportar el ruteo de actividades para que el trabajo se realice eficientemente.

**Sistemas de gestión de workflows**

* Define, crea y gestiona *la ejecución* de workflows.
* Se ejecutan en una o varias máquinas de workflow.
* Interpreta definiciones de procesos, interactúa con los participantes del workflow e invoca herramientas y aplicaciones IT

**Sistemas de gestión de procesos de negocio**

* Define, ejecuta, controla y monitorea *los procesos* de workflow.
* Soporta procesos P2A y A2A.
* Es un sw genérico dirigido por modelos explícitos de procesos para ejecutar y gestionar procesos de negocio.

**Actividad/Tarea**

* Paso lógico o atómico dentro de un proceso.
* Realizado por un actor o aplicación.
* Puede ser ejecutado en diferentes instancias de proceso.

**Caso/instancia de proceso/workflow**

* Objeto (producto, información, servicio) a ser procesado.
* Representa una ejecución específica del proceso.
* Cada instancia tiene su propio estado.
  + Variables de instancia del proceso: valor de los atributos.
  + Condiciones sobre la estructura del proceso: dependencia entre actividades, actividades ejecutadas y habilitadas.
* Ciclo de vida**:**
  + Iniciado
  + En ejecución
  + Suspendido
  + Interrumpido
  + Activo
  + Finalizado

**Instancia de una actividad**

* Ejecución de una actividad en una instancia de proceso específica.
* Actividad + instancia + recurso. Ejecución de una actividad en un workflow.
* Ciclo de vida:
  + Inactiva
  + Habilitada
  + En ejecución
  + Suspendido
  + Finalizado

**Ítem de trabajo**

* Representa el trabajo a ser realizado por un participante dentro de una actividad dentro de una instancia.
* Representa que la actividad está lista para ser ejecutada por un participante.
* Actividad + instancia. Paso lógico que puede ser ejecutado en un workflow.
* Ciclo de vida de un ítem de trabajo:
  + Creado
  + Ofrecido a un único recurso o múltiples
  + Asignar a un único recurso.
  + Comenzado
  + Suspendido
  + Interrumpido
  + Finalizado

**Lista de trabajos**

* Lista de ítems de trabajos asociados a un participante.

**Recursos**

* Llevan a cabo las actividades.
  + Sw o hw
  + Persona:
    - Rol
    - Grupo
* Modelos de recursos: definidos deparados de los modelos de workflow.

**Creación de la instancia de actividad**

* Asignar un ítem de trabajo a un recurso.
* La máquina de workflow tiene que decidir el orden en que se crean los ítems de trabajo y cómo asignarlos a un recurso para ejecutar una actividad.

**Herramientas de definición de procesos**

* Para el modelado y especificación de procesos
* Para el análisis de procesos
* Para la definición de recursos

**Servicio de ejecución de proceso**

* Compuesto de una o varias máquinas de procesos
* Interpreta especificaciones de procesos
* Crea y gestiona instancias
* Mantiene datos de control de las instancias

**Aplicaciones clientes del BPMS**

* Manejador de la lista de trabajo:
  + Presenta los ítems de trabajo
  + Provee propiedades relevantes a un ítem
  + Ordena y selecciona los ítems
  + Provee información del estado de la máquina de proceso
  + Soporta el inicio de una actividad en un caso
  + Reporta la terminación de una actividad

**Herramientas de administración y control**

* Agrega, modifica y elimina personas
* Inspecciona el estado de las instancias
* Manipula el estado de las instancias ante problemas
* Agrega y elimina especificaciones de proceso
* Reconfigura el BPMS
* Colecciona datos históricos y presenta indicadores de performance.

**Workflow nets**

* Perspectiva:
  + Funcional: describe las actividades del proceso.
  + Comportamiento: describe el flujo de control de las actividades.
* Es una red de Petri.
  + Grafo dirigido con nodos: places y transitions.
  + Place contienen tokens
  + Tokens: objeto dinámico.
  + Una transition se dispara cuando hay al menos un token en cada uno de sus places de entrada. Cada token removido del place de entrada, pasa a uno de salida.
* Describe el ciclo de vida de un caso o instancia de proceso.
* Tiene un place de inicio y uno de fin.
* Ejecución:
  + Habilitación de una transition: ítem de trabajo
  + Disparo de una transition: instancia de actividad
* Tipos de disparos:
  + Recurso
  + Evento externo
  + Evento de tiempo

**Análisis de modelos de procesos**

Se requieren técnicas y modelos formales para el análisis. Ej.: Redes de Petri.

* Verificación
  + Bien formado, libre de errores.
  + Errores comunes en WN:
    - Tareas sin condiciones de E/S
    - Tareas muertas: nunca pueden ser realizadas
    - Bloqueo de un caso antes que el place de fin sea alcanzado.
    - Caso en un ciclo infinito.
    - Tareas que aún pueden ser disparadas luego de que se alcance el fin.
    - Tokens que permanecen en la definición del proceso luego de que el caso ha sido completado.
  + Solidez en una WN:
    - Para cada aso siempre es posible alcanzar el estado final.
    - Para un caso, si el place de fin tiene token, todos los place restantes están vacíos.
    - Debería ser posible ejecutar cualquier actividad siguiendo diferentes rutas.
* Validación
  + Se comporta como se espera.
  + Se requiere simulación
* Análisis de performance
  + Satisface los requerimientos de tiempo de procesamiento, niveles de servicio y utilización de recursos.
  + Técnicas usadas para el análisis en WN:
    - Teoría de colas
    - Simulación

**Modelado de procesos de negocio** (BPMN)

* Énfasis en los aspectos de negocio (diseño, gestión, monitoreo).
* Independiente de la tecnología de implementación.
* Independiente de metodologías de rediseño de procesos.

**Tipos de procesos**

* Privado o interno de una organización
  + Ejecutable: con detalles de implementación
  + No ejecutable: sin
* Público o abstracto: interacciones entre un proceso privado con otras organizaciones.

**Interacciones entre organizaciones**

* Coreografía: describe comportamiento esperado entre organizaciones.
* Colaboración: describe interacciones entre dos o más organizaciones.
* Conversación: describe escenarios de intercambio de mensajes entre organizaciones.

**Tipos de diagramas**

* De proceso
* De colaboración
* De conversación
* De coreografía

**BPMN – Elementos básicos**

* Objetos de flujo: definen el comportamiento de los procesos de negocio.
  + Actividades
    - Tarea
      * Atributo para cualquier tarea: StartQuantity
      * Marcadores
        + Loop

LoopCondition

LoopCounter

* + - * + Múltiple instancia

IsSecuencial

LoopCardinality

CompletitionCondition

Behavior: sincronización de las instancias

None: pasan todos los tokens.

One: pasa el primer token que termina, los otros no.

All: pasa un token cuando terminan todas.

LoopCounter

* + - * + Compensación
      * Servicio
      * Envío
      * Recepción
      * Usuario
    - Subproceso
      * Embebido
      * Reusable
      * De evento
      * Marcadores: loop, múltiple instancia, compensación, ad-hoc
    - Call activity (actividad de llamada)
  + Eventos
    - Inicio
      * Ninguno
      * Mensaje
      * Tiempo
      * Condicional
      * Señal
      * Múltiple
      * Múltiple paralell
      * Para subproceso: los anteriores y escalation, error, compensación
    - Intermedio:
      * Ninguno
      * Mensaje
      * Tiempo
      * Escalation
      * Compensación
      * Condicional
      * Link
      * Señal
      * Múltiple
    - Fin
      * Ninguno
      * Mensaje
      * Error
      * Escalation
      * Cancelar
      * Compensación
      * Señal
      * Interrupción
      * Múltiple
  + Gateways
    - Exclusive Gateway (XOR)
    - Inclusive Gateway (OR)
    - Inclusive Gateway merge
    - Paralell Gateway (AND)
    - Event-based Gateway
    - Complex Gateway
* Objetos de conexión: manera de conectar objetos de flujo y artefactos.
  + Flujo de secuencia
  + Flujo de mensaje
  + Asociación
  + Asociación de dato
* Datos: información consumida o producida por las actividades.
  + Objetos de dato
  + Entradas de dato
  + Salidas de dato
  + Almacenes de dato
  + Propiedades
* Swimlanes: agrupan elementos de modelado.
  + Pool
  + Lane
* Artefactos: no tienen efecto. Aclaran el modelo.
  + Grupo
  + Anotación
* Proceso: grafo de objetos de flujo y de conexión que definen la semántica de ejecución del proceso.
* Token: define la semántica del flujo de secuencia a través de los objetos de flujo. Representa la ejecución de una instancia de proceso. No se pasa a través del flujo de mensajes ni por asociaciones de datos.
  + Una actividad será instanciada cuando arribe al menos un token desde uno de sus flujos de secuencia de entrada.
  + Cuando finaliza una instancia, un token será generado por cada flujo de secuencia de salida.

**Elementos adicionales**

* Manejo de excepciones: eventos intermedios asociados a una actividad que la pueden interrumpir.
* Compensación: para deshacer la acción de una actividad previa que fue realizada y finalizada en forma exitosa pero que sus resultados y efectos no son más deseados y requieren ser revertidos. Para compensar una actividad debe estar terminada.
  + Manejador de compensaciones: tiene los pasos necesarios para revertir la actividad.
* Subproceso transaccional: comportamiento expresado a través de un protocolo de transacción.

**Patrones de modelado de procesos**

* Es la descripción de un problema recurrente y su solución.
* Se usa como muestra para medir una magnitud o para repetirla.
* Provee independencia de la tecnología de implementación.
* Características de una solución para ser considerada patrón:
  + Efectividad: Resuelve problemas similares en ocasiones anteriores.
  + Reusabilidad: es aplicable a diferentes problemas en distintas circunstancias.

**Patrones de modelado de procesos**

* Patrones de flujo de control: capturan la forma en que los tiepos de flujo de control son representados y usados en procesos.
* Patrones de datos: capturan la forma en que los datos son representados y usados en procesos.
* Patrones de recursos: capturan la forma en que los recursos son representados y usados en procesos.

**Patrones de flujo de control**

* Consiste de una descripción del flujo de control capturado, posibles sinónimos, ejemplos de escenarios concretos y problemas de implementación de soluciones.
* Categorías de patrones:
  + Básicos
    - **Secuencia**
    - **Paralell Split**
    - **Sincronización**
    - **Exclusive choice**
    - **Simple merge**
  + Bifurcación y sincronización avanzados
    - Multi-choice
    - Sincronización merge
    - Multi-merge
    - Discriminator
  + Estructurales
    - **Arbitrary cycles**
    - **Terminación implícita**
  + Involucran múltiples instancias
    - Múltiple instancia sin sincronización
    - Múltiple instancia with a priori desing time knowledge
    - Múltiple instancia with a priori runtime knowledge
    - **Múltiple instancia without a priori runtime knowledge**
  + Basados en estados
    - **Deferred choice**
    - Interleaved paralell routing
    - Milestone
  + Cancelación
    - Cancel activity
    - Cancel case

**Procesos de negocio en entornos Business to Business**

* E-commerce: compra y venta realizadas por personas.
* E-business: compra y venta realizadas por empresas.
* Business to Consumer: transacciones comerciales entre empresas y consumidores.
* Business to Business (B2B): transacciones comerciales entre empresas.

**Colaboraciones B2B**

* Se establecen relaciones de negocio.
* Se define un acuerdo de colaboración: período de la relación, productos a intercambiar, metas de negocio comunes.
* Hay comunicación bidireccional.
* Los procesos privados son integrados con los de su socio 🡪 Definición de procesos colaborativos.
* Se toman decisiones en forma conjunta.
* Propósitos:
  + Alcanzar metas comunes.
  + Mejorar beneficios y disminuir costos.
* Se definen modelos de colaboración 🡪 reglas generales a ser aplicadas y procesos a llevar a cabo.

**Proceso de negocio colaborativo**

* Define el intercambio de información y la coordinación de actividades entre las empresas.
* Para alcanzar una meta de negocio común.
* Se define y ejecuta conjuntamente.
* Procesos colaborativos como procesos abstractos:
  + Cada empresa define la lógica privada para el soporte de intercambio de mensajes 🡪 procesos privados
  + Un proceso colaborativo es implementado a través de los procesos privados de cada parte.
  + Los sistemas de las empresas ejecutan los procesos de integración que contienen la lógica de coordinación (intercambio de mensajes) y la lógica privada (llamada a aplicaciones internas).

**Colaboraciones B2B**

* Nivel tecnológico
  + Soporte para la interoperabilidad entre los sistemas B2B.
  + Capas de interoperabilidad:
    - Capa de comunicación
    - Capa de contenido
    - Capa de procesos de negocio
  + Cada capa tiene un estándar para el intercambio de mensajes entre los sistemas.
* Nivel de negocio

**Sistemas de información B2B**

* Sistemas e-marketplaces
  + Intercambio a través de un hub
  + Gestión centralizada de los procesos.
* Sistemas peer-to-peer
  + Intercambio directo.
  + Gestión descentralizada de los procesos.

**Tipos de procesos de negocios**

* Colaborativos
  + Vista global de la colaboración
  + No ejecutable
* De interfaz
  + Describe el comportamiento público de un participante en el proceso colaborativo.
  + No ejecutable
* De integración
  + Refina el proceso se interfaz con comportamientos privados.
  + Incluye las actividades para procesar la información intercambiada, transformar los datos e invocar sistemas internos.
  + Ejecutable con un BPMS
  + Es ejecutado por el sistema de GPN del participante.

**Arquitecturas orientadas a servicios (SOA)**

**Servicio**

* SW autónomo e independiente de la tecnología y de otros servicios.
* Provee un conjunto de funciones u operaciones.
* Ejecutan funciones de negocio 🡪 responder solicitudes, ejecutar procesos de negocio.
* Exponen las funcionalidades de las organizaciones en un contrato.
* Computación orientada a servicios: usa los servicios como bloques fundamentales para construir un sistema.
* Un servicio incluye:
  + Contrato
    - Base para la comunicación entre servicios
    - Describe el comportamiento del servicio y los términos para el intercambio de mensajes.
    - Propósito: soportar nuevos requerimientos de automatización minimizando el desarrollo y priorizando la reutilización.
    - Para consumir un servicio, se accede mediante el contrato 🡪 independencia
    - Incluye:
      * Interfaz de servicio
        + Define el conjunto de operaciones que provee el servicio.
        + Puede definir el formato de los mensajes.
      * Esquema de datos de intercambio
      * Calidad del servicio
        + Características de calidad. No incluyen detalles de acceso o interfaz.

Detalles de autorización, seguridad, disponibilidad.

* + - Evitar la transformación de los datos en el intercambio de mensajes.
    - Ofrece una estandarización de la representación de los datos a ser utilizados por todos los servicios.
    - Permite centralizar las definiciones de calidad de servicio.
  + Implementación del servicio
    - Lógica de negocio que soporta el servicio implementada en una tecnología específica.
      * Servicios web, .Net.
  + Detalles de acceso
    - Ubicación: dirección de red donde está el servicio.
    - Binding: protocolo de transporte a usar para la comunicación.
* Tipos de servicio
  + Simple
    - La lógica interna es implementada accediendo a un sistema de información local.
  + Compuesto
    - La lógica de negocio implica operaciones de otros servicios.
    - Combinación de servicios simples y compuestos.

**Comunicación entre servicios**

* Intercambian mensajes
* Una operación puede tener mensajes de entrada y/o salida.
* Patrones de interacción:
  + One-way
    - Único mensaje
    - El cliente invoca al servicio sin obtener respuesta
    - Interacción asíncrona
  + Request-reponse
    - Dos mensajes
    - El cliente invoca al servicio y obtiene una respuesta
    - Interacción sincrónica
  + Solicit-reponse
    - Dos mensajes
    - El servicio envía una notificación al cliente y recibe una respuesta
    - Interacción sincrónica
  + Notification
    - Único mensaje
    - El servicio envía un mensaje
    - Interacción asincrónica

**Principios de orientación a servicios**

* Deben compartir el contrato
  + Definir y compartir un contrato para la interacción entre servicios.
* Deben ser débilmente acoplados
  + La lógica de los servicios tiene que ser lo más independiente posible de los demás servicios.
* Deben ser abstractos
  + Ocultar los detalles de implementación e infraestructura tecnológica de ejecución.
* Deben ser reusables
* Deben ser autónomos
  + Autonomía: grado de libertad para controlar y tomar sus propias decisiones.
  + Mayor control posee un programa 🡪 mayor autonomía.
  + Cada servicio puede tener su propio entorno de ejecución.
* Sin estados
  + No deberían requerir mantener información sobre solicitudes de sus funciones cuando son implementados.
  + No deberían ser dependientes del estado de otros servicios.
* Debe poder ser descubierto
  + Para ser utilizado.

Interrelación entre los principios

* El contrato proporciona independencia y asegura la reusabilidad.
* El bajo acoplamiento minimiza las dependencias consiguiendo la reusabilidad.
* La composición permite la reusabilidad.
* La autonomía ofrece independencia del entorno de ejecución y asegura la reusabilidad.
* Que no se mantengan estados ofrece más oportunidades de reusabilidad.
* El descubrimiento evita duplicaciones y aporta a la reusabilidad.

**SOA – Arquitectura orientada a servicios**

* Forma de organizar y diseñar un sistema para proveer servicios a aplicaciones de usuarios o a otros servicios a través de interfaces publicadas que pueden ser encontradas.
* Independiente de la tecnología.
* Tipos de componentes
  + Proveedor de servicios (service provider)
    - Provee funcionalidades de aplicaciones y negocio como servicios.
    - Publica servicios.
    - Implementa el servicio.
  + Solicitante de servicios (service requestor)
    - Tiene un requerimiento que puede ser cumplido por un servicio.
    - Busca e invoca servicios.
  + Registro de servicios (service registry)
    - Provee un repositorio de servicios donde se pueden publicar y encontrar servicios.
* Operaciones
  + Creación
  + Descripción
  + Publicación
  + Descubrimiento
  + Invocación
  + Eliminación
* Beneficios
  + Encapsulación y modularización de componentes de SW autónomos, heterogéneos y débilmente acoplados.
    - Los servicios son definidos, desarrollados y gestionados de manera independiente.
  + Reutilización de servicios.
  + Incremento de la interoperabilidad.
  + Alineación entre los procesos de negocio y los sistemas.
    - Facilita la integración de las aplicaciones para la automatización de los procesos.
  + Alineación de la solución tecnológica con la solución de negocio
    - El servicio encapsula una funcionalidad de la empresa.
  + Agilidad ante cambios organizacionales
    - Reuso de los servicios.
    - Reemplazo de los servicios.
    - Cambio en la lógica y no en el contrato.

**Clasificación de los servicios**

Según el tipo de lógica que encapsulan, el potencial reuso de la lógica y cómo está relacionada la lógica con los dominios existentes dentro de una empresa.

* Servicios de utilidad
  + Provee funciones e capacidades independientes de la lógica de negocio.
  + Altamente reusable.
* Servicios de entidad
  + Define su funcionalidad sobre una o más entidades de negocio.
  + Independiente de la lógica de procesos de negocio.
  + Altamente reusable.
* Servicios de tareas
  + Provee funcionalidades directamente asociadas a las tareas de negocio.
  + Generalmente tiene un alcance funcional a una entidad de negocio.
  + Independiente de los procesos de negocio.
  + Puede ser reusado.
* Servicios de integración o composición
  + Expresa la lógica de un proceso de negocio a través del cual se invoca a otros servicios.
  + Poca probabilidad de reuso.

**Orden para identificar servicios**

* Dos enfoques
  + Top-down
  + Bottom-up
* Se recomienda una mezcla de ambos:

1. Servicios compuestos
2. Servicios de tarea
3. Servicios de aplicaciones existentes
4. Resto de servicios

**Tipos de granularidad**

* Granulación de servicios
  + Alcance funcional del servicio.
  + Cantidad de lógica funcional que puede encapsular el servicio.
* Granulación de funciones
  + Alcance funcional de una función específica de un servicio.
  + Granularidad fina 🡪 menos trabajo
* Granulación de datos
  + Cantidad de datos que una función de un servicio necesita intercambiar para llevar a cabo su función.
  + Granularidad amplia: mensajes centrados en documentos (conjuntos enteros de información)
  + Granularidad fina: comunicación RPC (parámetros de datos más pequeños)
* Granulación de restricciones
  + Nivel de detalle para validar los parámetros de una función.

**Acoplamiento de servicios**

* Los servicios deberían poder evolucionar con un mínimo impacto en sus relacionados.
* Reducir el número de conexiones entre servicios.
* Un bajo nivel de acoplamiento evita problemas de redundancia y duplicación de servicios.
* Definir primero el contrato y luego la lógica de implementación

**Cohesión de servicios**

* Grado de relación funcional que tienen las funciones de un servicio.
* Declarar diferentes interfaces, no todo en una.
* Una alta cohesión permite:
  + Aumento de la claridad y comprensión de las funciones
  + Simplificación del mantenimiento.
  + Granularidad de servicio en niveles razonables.
  + Bajo acoplamiento

**Middleware**

* Facilita y gestiona el intercambio de información entre aplicaciones a través de plataformas heterogéneas.
* Facilita el manejo y programación de aplicaciones distribuidas. Simplifica el trabajo de los programadores de crear conexiones y sincronizaciones.
* Soporta la integración de aplicaciones bajo un servicio de interfaz común.
* Middleware como abstracción de programación
  + Oculta la complejidad de construir un sistema distribuido.
  + Provee abstracciones de programación de alto nivel para invocar aplicaciones distribuidas.
* Middleware como infraestructura de sistemas distribuidos
  + Provee herramientas para implementar las abstracciones de programación en tiempo de diseño.
  + Interpreta estas abstracciones en tiempo de ejecución.
  + Oculta la heterogeneidad de los sistemas.
* Tipos de middleware
  + RPC (remote procedure call)
    - Soporta llamadas a procedimientos remotos en forma transparente y uniforme.
    - Modelo de comunicación síncrono.
    - Como abstracción de programación:
      * Oculta los detalles de comunicación detrás de una llamada.
  + TP Monitors
  + Object Brokers
  + MOM (message oriented middleware)
    - Modelo de comunicación asincrónico.
    - Usa colas de mensajes.

**Servicios web**

* Son aplicaciones de negocio modulares con interfaces basadas en estándares, orientadas a internet y abiertas.
* Soporta interacciones directas con otros agentes de sw utilizando protocolos de intercambio basados en internet.
* Expone la funcionalidad de un sistema y la hace disponible a través de tecnologías web estándares.
* Funcionan como adaptadores: encapsulan aplicaciones proveyendo una única interfaz y un acceso web a la misma.

**Tecnologías estándar de servicios web**

* SOAP
  + Protocolo de comunicación para servicios web.
  + Para el intercambio de información entre entidades pares descentralizadas.
  + Especifica
    - Formato de mensaje para comunicaciones one-way y empaquetamiento.
    - Convenciones sobre cómo realizar llamadas RCP a través de mensajes SOAP.
    - Conjunto de reglas para procesar un mensaje SOAP.
    - Descripción de cómo debería ser transportado el mensaje a través de internet.
  + No especifica qué hacer con el mensaje.
* WSDL
  + Lenguaje para la descripción de los contratos o interfaces de los servicios web.
  + Separa la información del protocolo de transporte y de la dirección del servicio.
* UDDI
  + Registro de servicios para servicios web.
  + Soporta la construcción de repositorios de servicios.
  + Soporta la definición de un servicio de directorio y nombres para servicios web.
  + Define estructuras de datos y apis para publicar servicios y consultarlos.

**Diseño de servicios web**

* Interfaces
  + Indican las operaciones que pueden ser invocadas de un servicio.
  + Especifica la dirección del servicio y el protocolo de transporte.
  + Estándar usado: WSDL.
* Protocolo de negocio o conversación
  + Orden en que pueden invocarse las operaciones de un servicio.
  + Estándar: BPEL

**Enterprise Serivce Bus – ESB**

* Middleware de integración de aplicaciones basado en mensajes y estándares abiertos para implementar, desplegar y gestionar aplicaciones basadas en SOA.
* Gestiona la comunicación entre servicios web.
* Basado en interacciones orientadas a eventos y de mensajes asincrónicos entre aplicaciones.
* Posibilita interoperabilidad entre aplicaciones con adaptadores e interfaces.
* Facilita la escalabilidad de las conexiones entre aplicaciones.
* Cuando un servicio web es desplegado en el ESB, puede ser usado en cualquier aplicación o servicio conectado al ESB.
* La invocación de servicios no requiere conocer su ubicación exacta.