**SEMANA 1**

ANÁLISIS Y DISEÑO DE ARQUITECTURA EMPRESARIAL

**ANÁLISIS**:

Examen detallado de algo tangible o intangible, para conocer sus características o cualidades, o su estado, y extraer conclusiones; las que se realizan separando o considerando por separado las partes que la constituyen.

Un **Análisis** es un estudio profundo de un sujeto, objeto o situación con el fin de conocer sus fundamentos, sus bases y motivos de su surgimiento, creación o causas originarias. Un **análisis estructural** comprende el área externa del problema, en la que se establecen los parámetros y condiciones que serán sujetas a un estudio más específico, se denotan y delimitan las variables que deben ser objeto de estudio intenso y se comienza el **análisis exhaustivo** del asunto de la tesis.

Un análisis consiste en identificar los componentes de un todo, separarlos y examinarlos para lograr acceder a sus principios más elementales

**DISEÑO**:

Actividad creativa que tiene por fin proyectar objetos que sean útiles y estéticos.

Un **DISEÑO** es el resultado final de un **proceso**, cuyo objetivo es buscar una solución idónea a cierta problemática particular, pero tratando en lo posible de ser práctico y a la vez estético en lo que se hace. Para poder llevar a cabo un buen diseño es necesario la aplicación de distintos **métodos y técnicas** de modo tal que pueda quedar plasmado bien sea en bosquejos, dibujos, bocetos o esquemas lo que se quiere lograr para así poder llegar a su producción y de este modo lograr la apariencia más idónea y emblemática posible.

Un diseño se trata básicamente de las diferentes formas que puede tomar un objeto, tomando en cuenta que el mismo debe contar con armonía visual, sin perder de vista las funciones que debe cumplir el mismo.

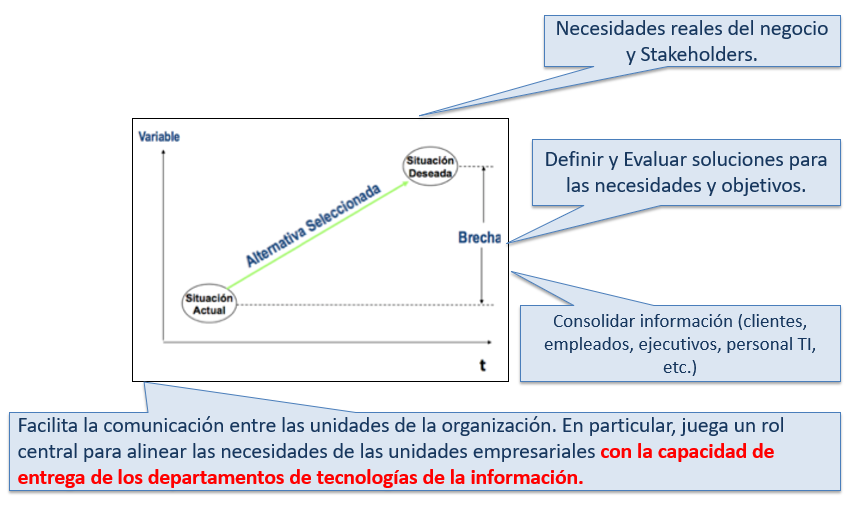
A quien **diseña** se le conoce como **diseñador**, es un individuo que básicamente debe contar con ciertas características, como lo son poseer el don de la imaginación y creatividad y la vez debe tener conocimiento experto de lo que está haciendo, aunado a las habilidades técnicas necesarias para así poder llevar a cabo las investigaciones pertinentes sobre el diseño que desee realizar. Hay especialistas en distintos campos del diseño, por ejemplo diseñadores de moda **(vestidos y atuendos)**, diseñadores tecnológicos **(computadoras y teléfonos)**, diseñadores web (estructura y forma de las páginas webs) y muchos más.

**ANALISIS DE REQUERIMIENTOS**

El Análisis de Negocio es el conjunto de tareas y técnicas utilizadas con el fin de entender la estructura, políticas, y operaciones de una organización, y para recomendar soluciones que permitan a la organización alcanzar sus metas.

internos y externos.

* Implica entender cómo funciona la organización y definir las capacidades que requiere para proporcionar los productos y servicios.
* Incluye desde la definición de metas de la organización, objetivos específicos (qué deben estar alineados o conectados) y la definición de cursos de acción
* Incluye la definición de la interacción de las diferentes unidades organizativas y stakeholders internos y externos





**Un requerimiento** puede ser no declarado, implícito en o derivado de otros requerimientos o directamente declarado y administrado. Uno de los objetivos clave del Análisis de Negocio es asegurar que los requerimientos sean visibles y entendidos por todos los *stakeholders*.

**REQUERIMIENTO**

1. Una condición o capacidad requerida por un *stakeholder* para resolver **un problema** o alcanzar un objetivo.

2. Una condición o capacidad que debe ser cubierta o estar contenida en una solución o componente de la solución para cumplir con un contrato, estándar, especificación, o cualquier otro documento formal.  
3. Una representación documentada de una condición o capacidad como en (1) o (2).

**EL PROBLEMA**

**Incidente**:

Es un evento que se presenta de manera recurrente en un determinado período de tiempo (usualmente conocido). Ejemplo: Todos los fines de mes, el sistema contable presenta demoras en el tiempo de procesamiento del cierre mensual.

**Problema de Investigación:**

Es todo aquello que se convierte en objeto de reflexión y sobre lo cual se percibe la necesidad de conocer, estudiar y solucionar (1).

Ejemplo: Demora en el procesamiento del cierre mensual.

**DEFINICION DE OBJETIVOS**

**A close up of a sign

Description generated with very high confidence**

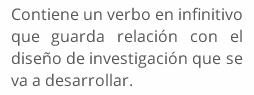
**Tipos de objetivos**

**A screenshot of a cell phone

Description generated with very high confidence**

**REDACCIÓN DE OBJETIVOS**

**A picture containing screenshot

Description generated with high confidenceA close up of a piece of paper

Description generated with very high confidence**

**Requerimientos del negocio**.

* Definiciones de alto nivel de las metas, objetivos o necesidades de la empresa.
* Describen las razones del por qué un proyecto ha sido iniciado, los objetivos que el proyecto debe alcanzar y las métricas que serán utilizadas para medir su éxito.
* Los requerimientos del negocio describen necesidades de la organización como un todo y no de grupos de usuarios o *stakeholders* dentro de ella.
* Son desarrollados y definidos a través del *Análisis Empresarial*

**Requerimientos de los stakeholders.**

* Definiciones de las necesidades de un stakeholder en particular o una clase de stakeholders.
* Describen las necesidades que tiene un stakeholder dado y cómo ese stakeholder interactuará con una solución.
* Los requerimientos de los stakeholders sirven como puente entre los requerimientos del negocio y las diversas clases de requerimientos de la solución.
* Son desarrollados y definidos a través del Análisis de los requerimientos.

**Requerimientos de la solución**.

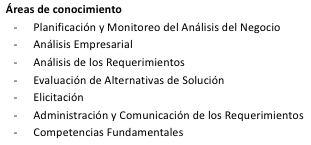
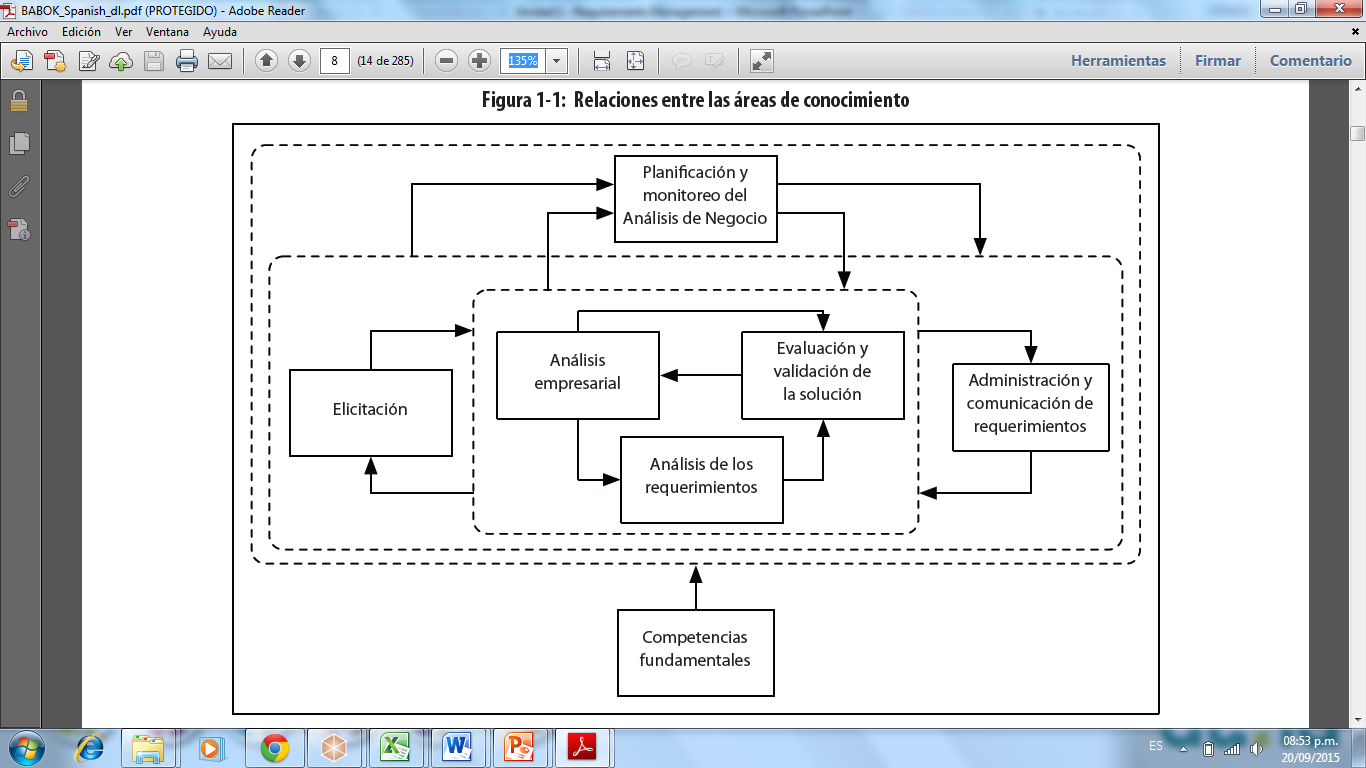
* Describen las características de una solución que satisface los requerimientos del negocio y requerimientos de los *stakeholders*.

**Requerimientos funcionales.** Describen el comportamiento e información que la solución administrará. Describen las capacidades que el sistema deberá poder llevar a cabo.

**Requerimientos no-funcionales**. Condiciones que no están directamente relacionadas con el comportamiento o funcionalidad de la solución sino **del ambiente**. Conocidos también como requerimientos de calidad o suplementarios. Estos pueden incluir requerimientos relacionados con capacidad, velocidad, seguridad, disponibilidad y, la arquitectura de la informacion y la presentacion de la interfaz del usuario.

**Requerimientos de transición.**

* Estos describen las capacidades que la solución debe tener con el fin de facilitar la transición del estado actual de la empresa hacia el estado futuro deseado, pero que no serán requeridos una vez que la transición se haya llevado a cabo.
* Se diferencian de otros tipos de requerimientos porque son siempre de naturaleza temporal y porque no podrán ser desarrollados hasta que ambas, la solución existente y la nueva sean definidas. Habitualmente cubren la conversión de datos de sistemas existentes, brechas en las habilidades que deben ser atendidas y otros cambios relacionados para alcanzar el estado futuro deseado.
* Son desarrollados y definidos a través de la *Evaluación y validación de la solución.*



1. Determina qué actividades son necesarias con el fin de llevar a cabo el esfuerzo. (Identificación de stakehodlers, como evaluar el progreso del trabajo)
2. Describe cómo el analista de negocio identifica una necesidad de negocio, refina y aclara la definición de esa necesidad y define el alcance de una solución.
3. Describe como el analista de negocio prioriza y desarrolla los requerimientos de los stakehodlers y de la solución con el fin de habilitar al equipo del proyecto implementar la solución.
4. Evalúa las soluciones propuestas para determinar cuál solución se ajusta mejor a la necesidad del negocio, identifica brechas y deficiencias en las soluciones, y determina soluciones provisionales.
5. Trabajan con los stakeholders para identificar y entender sus necesidades y preocupaciones, y entender el medio ambiente en el que trabajan.
6. Describe cómo los Analistas de Negocio administran conflictos, asuntos y cambios con el fin de asegurar que los stakeholders y el equipo del proyecto mantengan el acuerdo respecto al alcance de la solución.

**MODELAMIENTO DE ARQUITECTURA DE NEGOCIO (*REPASAR PPT 1*)**

**ANÁLISIS DE PROCESOS CON ENFOQUE SOA**

BPMN CON ENFOQUE SOA

* El proceso de negocio debe estar automatizado a través de servicios SOA.
  + Identificar todas las actividades del proceso.
  + Descomponerlas en términos de servicios para automatizar el proceso.

¿QUÉ ES UN SERVICIO?

* Unidad lógica, que es el resultado de la fase: “análisis orientado a servicios”.
* Agrupan lógicas que el proceso debe llevar a cabo.
* Tienen características diferenciadoras
  + reusable
  + que se pueda descubrir,
  + Escalable,
  + Autónomo,
  + Estandarizado.

ORIENTACIÓN A SERVICIOS

* Es un paradigma, una manera de pensar, una forma de diseñar activos de TI.
* Se definen 08 principios de diseño, los que garantizan que cada servicio tenga características diferenciadoras.
* El resultado de aplicar estos 08 principios de diseño a una unidad lógica (ej. Proceso, requerimiento, historia de usuario), se podrán diseñar servicios SOA.
* El “análisis orientado a servicios” es una fase de un método para identificar, clasificar, agrupar y componer servicios.

**TIPOS DE LÓGICA**

Permiten definir y clasificar servicios

* **NEGOCIO**

Lógica que expresa “CAPACIDADES DE NEGOCIO”, a través de diferentes artefactos.

* + BPM
  + BPMN
  + Taxonomías
  + Ontologías
  + Modelos de datos
  + Información, etc.
  + Ejemplo: Capacidades de un banco: cuenta bancaria, cuenta de ahorros, un débito, un crédito, un proceso de solicitud de crédito, etc.
* **UTILIDAD**

Lógica que “SOPORTA Y PROCESA”, a través de recursos tecnológicos, las “CAPACIDADES DE NEGOCIO”. Es esa lógica que es tecnológica, de aplicación, que está a bajo nivel.

* + Lógica de acceso a datos.
  + Lógica de mensajería.
  + Lógica de seguridad.
  + Lógica de LOG.
  + Lógica de email, etc.
  + Ejemplo: Del proceso de solicitud de crédito, tendrá que haber alguna actividad que diga “generar solicitud” o “generar desembolso” desde el punto de vista del negocio. Pero estas capacidades, requieren de ciertas lógicas de utilidad, ya a un nivel tecnológico, que sea quien lleve a cabo ese registro a nivel de sistema de información que genera el desembolso.

**TIPOS DE LÓGICA**

Permiten definir y clasificar servicios

* **AGNÓSTICA**

Lógica “GENÉRICA”, con alto potencial de ser “RE USADA” por varias capacidades de negocio; las que pueden ser otros servicios u otros procesos.

* + Ejemplo: Si hablamos de un cliente en una organización, este servicio de cliente tendrá lógica que me permita “CONSULTAR” su información. Esta ”CONSULTA” es genérica, porque se puede usar para diferentes capacidades de negocio; es decir, puede haber un proceso que esté consultando la información de un cliente para generar un desembolso, otro proceso que puede consultar al cliente para generarle una factura, otro proceso consulta para genera una orden de compra, etc.

El beneficio directo es el retorno de inversión y la agilidad organizacional que brinda SOA. Esto soe logra identificando y clasificando la lógica, lo suficientemente genérica, como para que pueda ser re usada por otros procesos. Con esto evitamos la redundancia de aplicaciones e información.

Un proyecto exitoso SOA, va más allá de la implementación de un ESB (Enterprise Service Bus)

* **NO AGNÓSTICA**

Lógica con funcionalidad “ESPECÍFICA”; dada su naturaleza, no tiene potencial de ser re usable.

* + Ejemplo: Generar una OC, generar una solicitud de crédito, generar una factura.

El fin no es exponer “WEB SERVICES”, sino identificar servicios de negocio.

Se recomienda tener claros los tipos de lógica, a la hora de descomponer los procesos.

***Se toma un proceso, se descompone en unidades lógicas, se agrupan en “tipos de lógica” (negocio, utilidad, agnóstica o no agnóstica).***

***Sobre la base de esta clasificación por tipo de lógica, se podrán definir diferentes “MODELOS DE SERVICIO”.***

**MODELOS DE SERVICIO**

Los “tipos de lógica” me permiten definir los ”MODELOS DE SERVICIO”. Son una clasificación específica de los servicios.

* **TAREA**

Servicios con contextos funcionales de negocio “NO AGNÓSTICOS” (específicos).

* Ejemplo: generación de una OC, generación de una factura, una solicitud de crédito, la generación de un desembolso, etc.
* **ENTIDAD**

Servicios con contextos funcionales de negocio “AGNÓSTICOS” (generales). Cuando un servicio tiene esta funcionalidad, podemos afirmar que es un servicio “MULTI PROPÓSITO”.

* Ejemplo: La consulta al cliente es un servicio multipropósito, porque me permite llevar a cabo un propósito de generar una OC, una factura, una solicitud de crédito, etc. Por lo tanto, existen varios ”SERVICIOS DE TAREA” consumiendo “SERVICIOS DE ENTIDAD”.
* **UTILIDAD**

Servicios con contextos funcionales de ”UTILIDAD” y son “AGNÓSTICOS” (general y muy re usables, multi propósito). Su lógica está más asociada a la “UTILIDAD”.

* Ejemplos:
* Un servicio de persistencia que se encargue de ir a la base de datos y consultar la información.
* Un servicio de seguridad de información.
* Un servicio de exportación a PDF.
* Un servicio de auditoría

Es toda lógica que está a “BAJO NIVEL” de aplicación.

**CAPAS DE SERVICIO**

Son “VISTAS DE ARQUITECTURA” mostrando estas ”CAPAS DE SERVICIO” con responsabilidades específicas. Básicamente es Ingeniería de SW, sobre la base de la separación de responsabilidades, donde las diferentes capas tienen lógica funcional con responsabilidades específicas y no específicas.

Agrupan diferentes tipos de servicio dentro de un “Inventario de Servicios.”

Ejemplo:

TAREA: Generar una OC, y esta se lleva a cabo consumiendo diferentes servicios (tanto de entidad como de utilidad). Van a existir diferentes servicios de TAREA, es decir, con lógica específica, re usando diferentes servicios agnósticos como los de entidad y utilidad.

**INVENTARIO DE SERVICIOS**

Colección de Servicios que descomponemos y nos permite saber dónde están y cómo los podemos re usar para otros procesos de negocio.

Se define en la etapa de Análisis, de la ejecución iterativa del análisis orientado a servicios.

**ANÁLISIS ORIENTADO A SERVICIOS**

Primera fase del ciclo de vida de los servicios e inicia con pasos preparatorios, recolectando informaciónpara ejecutar el modelamiento de servicios donde se identifican, clasifican, agrupan y componen servicios candidatos por medio de la descomposición de los procesos de negocio.

**MODELAMIENTO DE SERVICIOS**

1. **Descomponer el proceso de negocio.**
2. **Filtrar lógica no apta para ser encapsulada en un servicio.**
3. **Identificar servicios agnósticos candidatos.**
4. **Identificar lógica específica de proceso.**
5. **Aplicar el paradigma de orientación de servicios .**
6. **Identificar composiciones de servicios candidatos.**
7. Analizar los requerimientos de procesamiento.
8. Identificar capacidades de servicio de utilidad candidatas.
9. Definir servicios de utilidad candidatos.
10. Aplicar el paradigma de orientación a servicios.
11. Revisar composiciones de servicios candidatos.

Revisar el agrupamiento de las capacidades candidatas

**Completar con PPT1 pag 54**

**SEMANA 2**

**Evolución y Madurez en la AE**

**Evolución**

DESAFÍOS PARA CONSTRUIR UNA ARQUITECTURA EMPRESARIAL

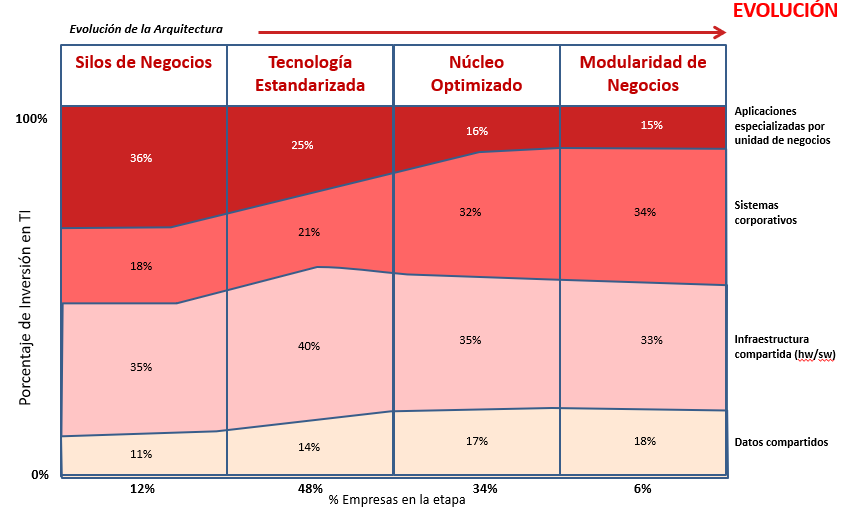
* Cambios frecuentes en nuevas tecnologías de información que se consolidan, se crean, etc. que amplían las posibilidades
* Los procesos construidos por la empresa se vuelven obstáculos en la evolución (dificultad, costumbre, reaprendizaje, etc.)
* La organización no puede parar, debe cambiar sus procesos y tecnologías subyacentes mientras sigue operando

<http://www.youtube.com/watch?v=L2zqTYgcpfg>

* Las organizaciones suelen seguir una evolución predecible en la mejora de su arquitectura empresarial

ETAPAS DE LA EVOLUCIÓN DE LA ARQUITECTURA EMPRESARIAL

* Silos de negocios
  + Las organizaciones maximizan las necesidades de negocios o funcionales de unidades individuales
* Tecnología estandarizada
  + Busca eficiencias en TI a través de la estandarización de tecnología, centralizando la gestión tecnológica
* Núcleo optimizado
  + Incrementa la estandarización de datos y procesos basándose en el Modelo Operacional
* Modularidad del negocio
  + Se definen “módulos de negocios”, que son grupos de funciones desacopladas que permiten recomponer procesos, logrando estandarización y particularización configurable o paramétrica



**SILOS DE NEGOCIO**

* La organización enfoca sus inversiones en TI en dar soluciones a problemas particulares y diferentes de las unidades de negocios
* Podría haber servicios compartidos como centros de datos, pero sólo para posteriormente aprovechar esto de manera separada
* Existen pocos o ningún estándar de TI
* Las inversiones de TI se basan en la reducción de costos por unidad de negocios
* Cada unidad de negocio compra aplicaciones de manera individual para satisfacer necesidades propias
* Los sistemas satisfacen al 100% las necesidades de cada unidad de negocios
* Esto crea aplicaciones que no pueden interactuar entre sí
* Modificar aplicaciones, incluso cambios pequeños, puede ser demorado, costoso y riesgoso
* Esta etapa obstruye la integración y estandarización de procesos de negocios

**TECNOLOGÍA ESTANDARIZADA**

* Las inversiones en TI son a nivel de la organización
* Las organizaciones en esta etapa intentan reducir las tecnologías o plataformas que tienen
* Menos plataformas significan menos opciones de infraestructura, aplicaciones, proveedores, etc.
* El énfasis de la unidad de negocios cambia de innovación a costos y preservación de estándares
* Es clave una gestión centralizada de los estándares
* La estandarización reduce riesgos y costos de servicios compartidos (soporte, mantenimiento, compras) y da confiabilidad, predictibilidad, seguridad y optimización del tiempo
* Las unidades de negocios buscan hallar la mejor solución tecnológica de acuerdo a los estándares existentes: La mejor opción funcional podría ser rechazada si no cumple los estándares
* Muchas organizaciones en esta etapa promueven estandarización y compartición de información, usualmente implementando un Datawarehouse; sin embargo, los datos transaccionales siguen separados

**NÚCLEO OPTIMIZADO**

* Las organizaciones se mueven de una vista particularizada de datos y aplicaciones a una vista integral (corporativa)
* TI minimiza la redundancia de datos, dando acceso democrático a la información operacional o transaccional
* El uso de interfaces estándares se incrementa
* Se empieza a estandarizar procesos de negocios
* Cambios básicos a los procesos de negocios o información, suele ser más difícil, pero construir nuevos productos y servicios en la tecnología ya existente suele ser más rápido

**MODULARIDAD DE NEGOCIOS**

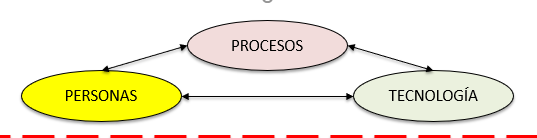
* Permite agilidad en el negocio, a través del uso de módulos reusables
* Es necesario tener documentados los procesos del negocio y hacer análisis profundos para identificar esos módulos
* Las unidades de negocios pueden recomponer sus procesos basados en los módulos reusables o usar interfaces de usuarias dinámicas que permiten usar la variedad funcional disponible
* La estandarización se mantiene y se extiende a interfaces y procesos
* Las organizaciones en esta etapa deben descubrir las oportunidades estratégicas que deben modularizar y permitir reusar.



**MODELO DE CAPACIDADES**

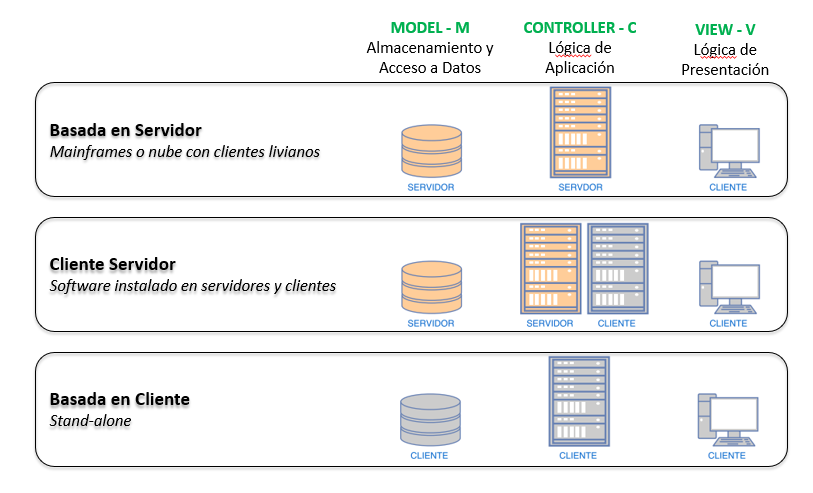
CAPACIDAD EMPRESARIAL

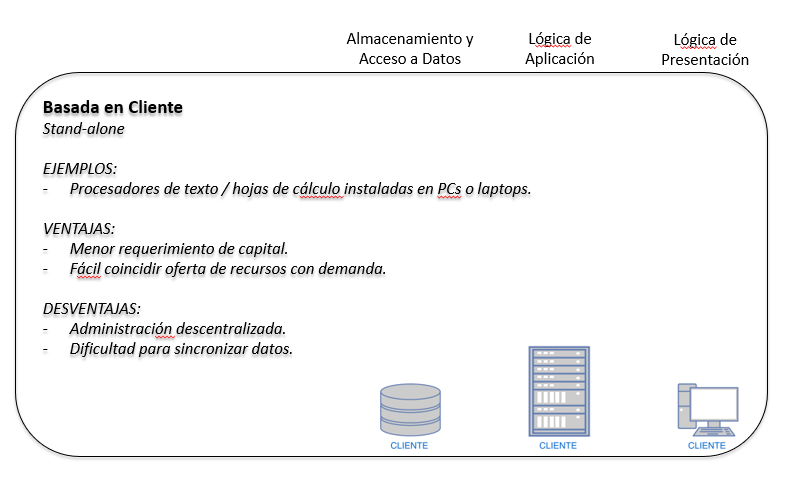
* Fortalezas y debilidades que tiene una empresa para desarrikkar su actuación competitiva. Fuente: wikieconomía
* Está conformado por personas, procesos y tecnología; independientes o por todas o cualquiera de estas combinaciones, la que le permite a la empresa tomar ventaja competitiva. Fuente: TOGAF

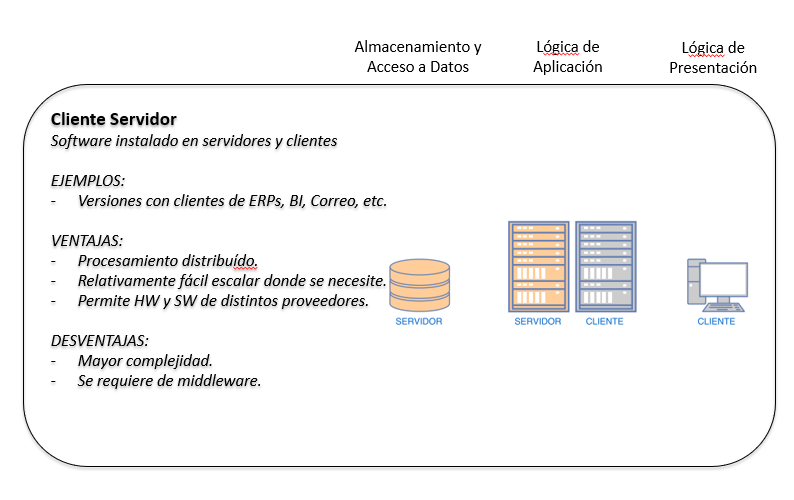


**ARQUITECTURA DE APLICACIONES**

**3 modelos generales**







MIDDLEWARE - *El pegamento que une las partes*

Elemento de software que conecta a 2 aplicaciones de sw independientes

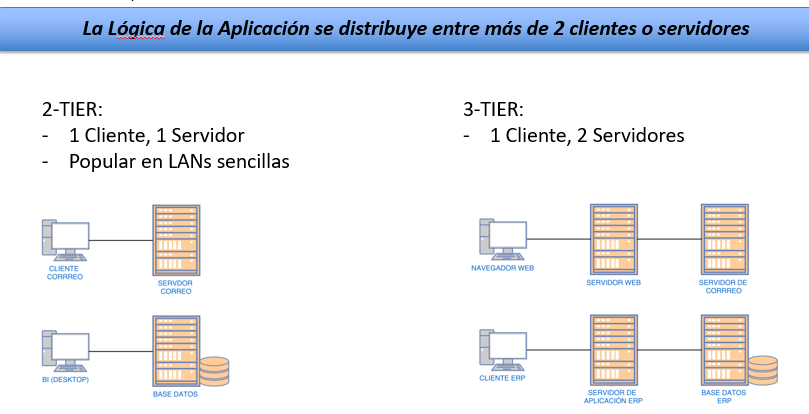
* Método estándar para intercambiar mensajes.
* Aísla cambios en un elemento de la aplicación de otros (ej. Usuario pasa de Windows a Linux y se conecta al mismo servidor, sin ajustes en ese servidor).

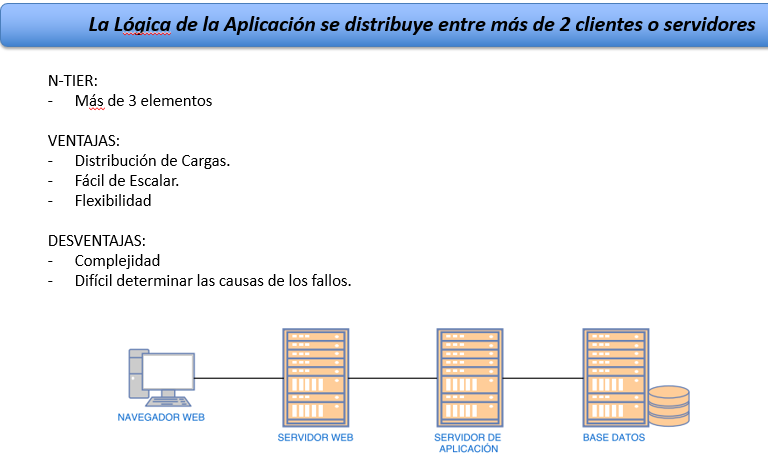
EJEMPLOS DE INTERFACES

* Distributed Computing Environment (DCE).
* Common Object Request Broker Architecture (CORBA).
* Open database connectivity (ODBC).

Java database connectivity (JDBC).

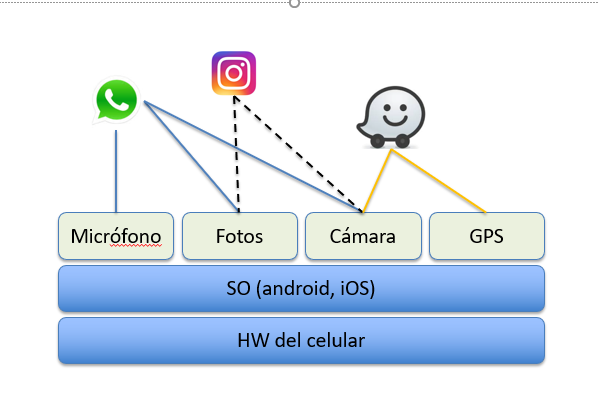
ARQUITECTURAS MULTINIVEL





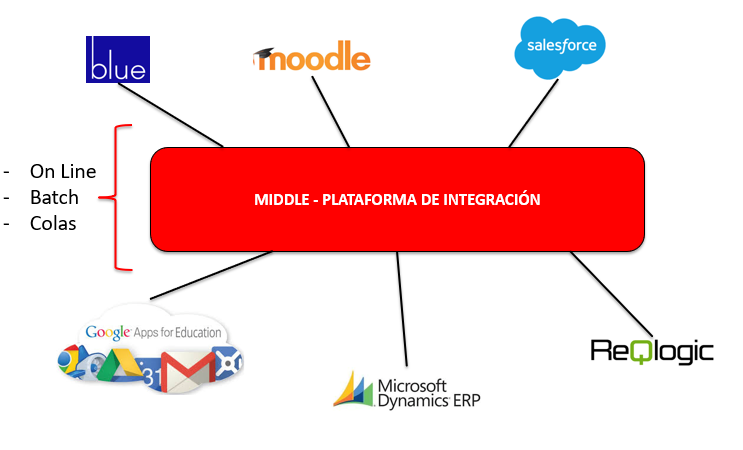
**APIs - *Application Program Interface***

Interfaz expuesta u ofertada por una aplicación de software para que otra se comunica con ella.



**PLATAFORMA DE INTEGRACIÓN**

Paquetes de Software diseñados para integrar sistemas.



**SOFTWARE STACK**

Todo el software necesario para que corra.

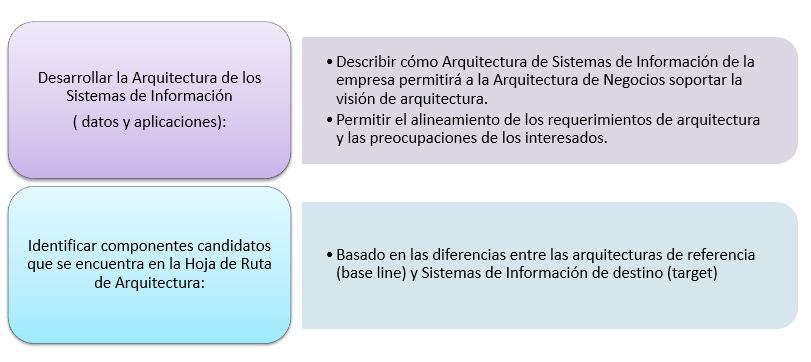
**VIRTUALIZACIÓN**

Servidores dentro de Servidores

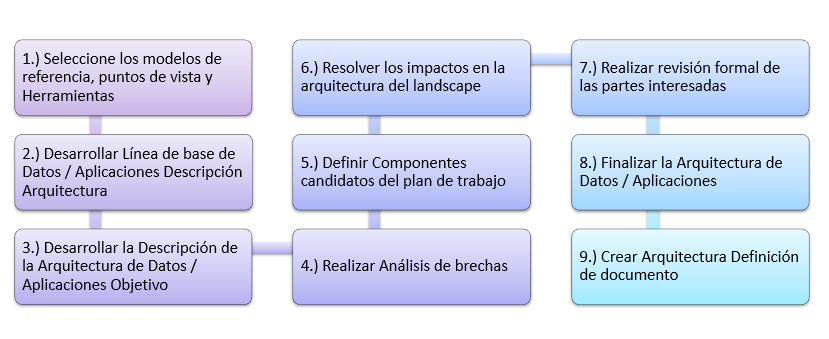
**CONTENEDORES**

Empaquemos solo lo estrictamente necesario

**ENFOQUE DE LA ARQUITECTURA EMPRESARIAL**



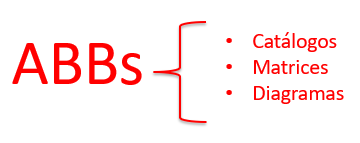
**Proceso: Arquitectura de Aplicaciones o Infrmación / Datos**



**Bloques de Construcción de la Arquitectura (architecture building blocks - ABBs)**

* Se definen o seleccionan como el resultado de la aplicación del ADM.

Características:

* Capturas de los Requisitos de Arquitectura
  + Negocios
  + Datos
  + Requisitos de aplicación
  + Requisitos de tecnología
* Dirigen y orientan el desarrollo de los SBBs

Deben incluir:

Building Blocks - BBs

* Funcionalidades Principales: requerimientos funcionales.
* Atributos: requerimientos no funcionales
* Semántica: metadatos semánticos, describen el contenido, el significado y la relación de los datos.
* Capacidad de Seguridad: qué y cómo debe cumplir.
* Capacidad de Administración: qué y cómo debe cumplir.

Interfaces

* Interoperatibilidad: con otros BBs
* Dependencias entre BBs: describe bloques de construcción dependientes con la funcionalidad requerida y las interfaces de usuario nombrados.

Mapeos

* Mapeo de Entidades de Negocio / Organización: El propósito es tener una referencia cruzada de los bloques de construcción con las entidades de negocio / organización.
* Mapeo de Políticas de Negocio / Organización: El propósito es tener una referencia cruzada de los bloques de construcción con las políticas de negocio / organización.
* Se refieren a la serie continua de soluciones; y puede ser, o bien adquiridas o desarrolladas.

Características:

* Definir qué productos y componentes serán implementadas por las funcionalidades.
* Definir la implementación.
* Cumplir con los requisitos de negocio
* Ser un producto o un proveedor.

Deben incluir:

Building Blocks - BBs

* Funcionalidades Específicas: requerimientos funcionales.
* Atributos: seguridad, manejabilidad, capacidad de ubicación, escalabilidad, etc. (requerimientos no funcionales).
* Rendimiento esperado
* Capacidad requerida de configuración
* Motivadores de diseño y restricciones, incluyendo la arquitectura física.

Interfaces

* Información General
* Interoperatibilidad: y la relación con otros BBs.
* Dependencia entre BBs: Describir bloques de construcción dependientes con la funcionalidad requerida y las interfaces de usuario con nombre.
* SBBs requeridas: SBB utilizado con la funcionalidad y los nombres de las interfaces utilizadas y requeridas.

Mapeos

* Mapeo de SBBs: describir el mapeo de los SBB a la topología de TI y políticas operativas.
* Relaciones entre SBBs y ABBs: describir las relaciones entre los SBB y ABB.
* **Completar con PPT2 pag 52**

**SEMANA 3**

**ARQUITECTURA DE DATOS**

**DATO:**

Es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Los datos describen hechos empíricos, sucesos y entidades. Es un valor o referente que recibe el computador por diferentes medios, los datos representan la información que el programador manipula en la construcción de una solución o en el desarrollo de un algoritmo. Los datos pueden consistir en números, estadísticas o proposiciones descriptivas.

**INFORMACIÓN:**

Los datos convenientemente agrupados, estructurados e interpretados se consideran que son la base de la información humanamente relevante que se pueden utilizar en la toma de decisiones, la reducción de la incertidumbre o la realización de cálculos.

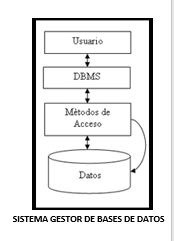
**BASE DE DATOS:**

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos.

**SISTEMA GESTOR DE BASE DE DATOS:**

Permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

**CLASIFICACIÓN DE BASES DE DATOS**:

**ESTÁTICAS**

Únicamente de lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para BI.

**DINÁMICAS**

Son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y edición de datos, además de las operaciones fundamentales de consulta.

**MODELOS DE BASES DE DATOS:**

**TRANSACCIONALES**

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

**MODELOS DE BASES DE DATOS:**

**RELACIONALES**

Este es el modelo utilizado en la actualidad para representar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar F. Cood​ de los laboratorios IBM, y no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones".

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

**MULTIDIMENSIONALES**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de CUBOS OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean aprender.

**ORIENTADA A OBJETCOS**

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación- Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herecia- Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo- Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

**DOCUMENTALES**

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes, sirven para almacenar grandes volúmenes de información de antecedentes históricos. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

**DEDUCTIVAS**

Un sistema de base de datos deductiva, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. Las bases de datos deductivas son también llamadas bases de datos lógicas, a raíz de que se basa en lógica matemática.

Este tipo de base de datos surge debido a las limitaciones de la Base de Datos Relacional de responder a consultas recursivas y de deducir relaciones indirectas de los datos almacenados en la base de datos.

**DISTRIBUIDAS**

La base de datos y el software SGBD pueden estar distribuidos en múltiples sitios conectados por una red. Hay de dos tipos:

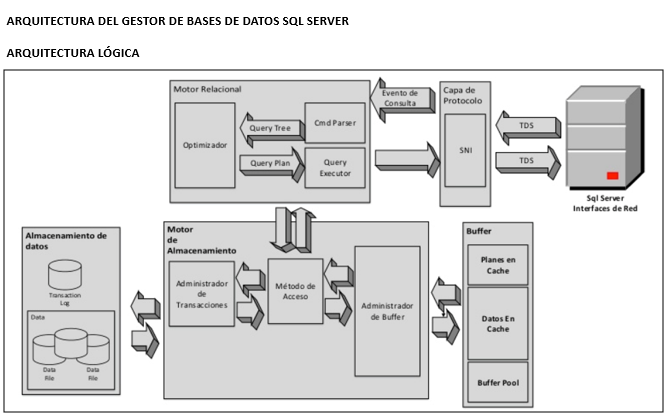
1. Distribuidos homogéneos: utilizan el mismo SGBD en múltiples sitios.

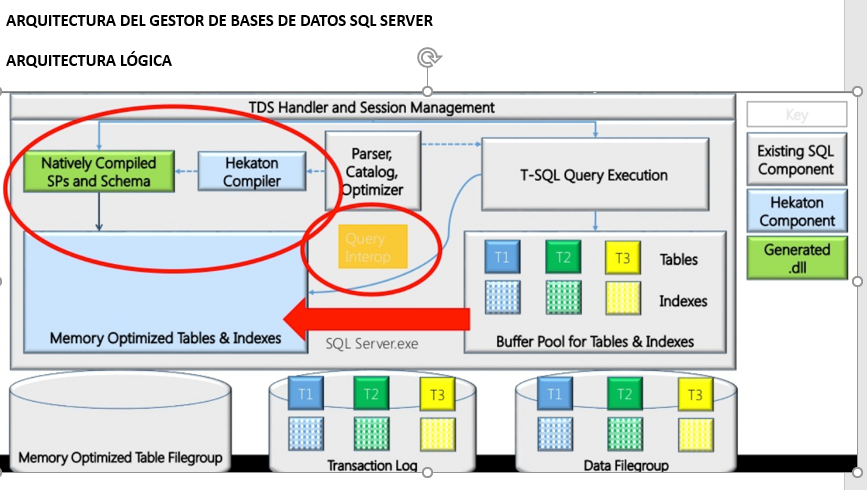
2. Distribuidos heterogéneos: Da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado de autonomía local y tienen acceso a varias bases de datos autónomas preexistentes almacenados en los SGBD, muchos de estos emplean una arquitectura cliente-servidor.

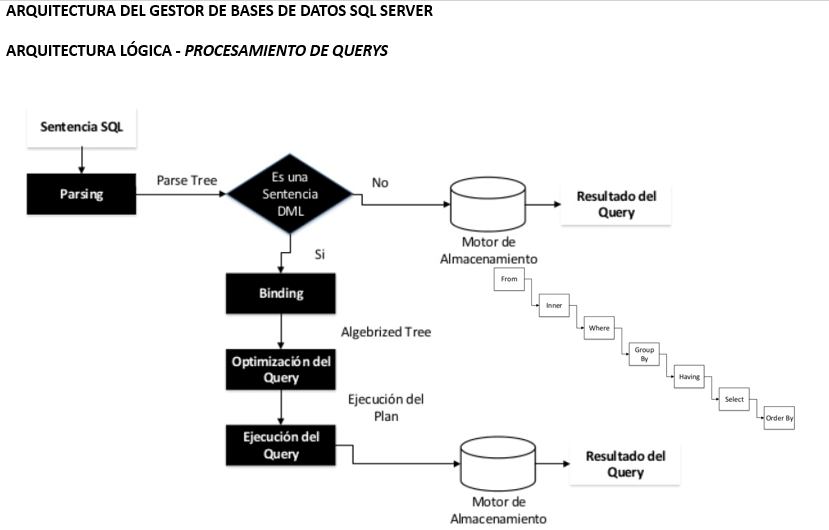
Estas surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etc.

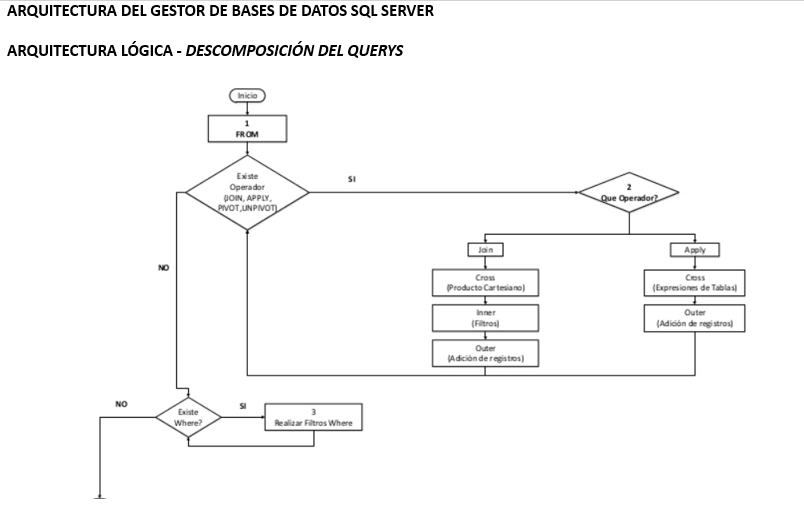
**CONSULTAS A LAS BASES DE DATOS**

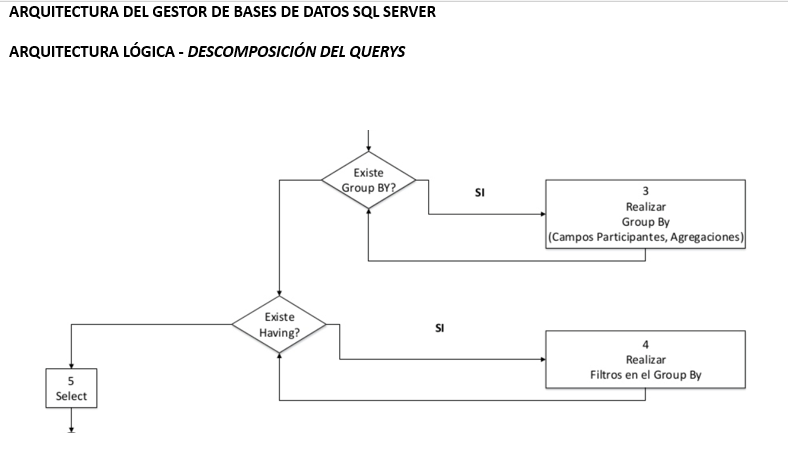
Una consulta es el método para acceder a la información en las bases de datos. Con las consultas se puede modificar, borrar, mostrar y agregar datos en una base de datos, también pueden utilizarse como origen de registro para formularios. Para esto se utiliza un lenguaje de consulta, denominado: SQL.

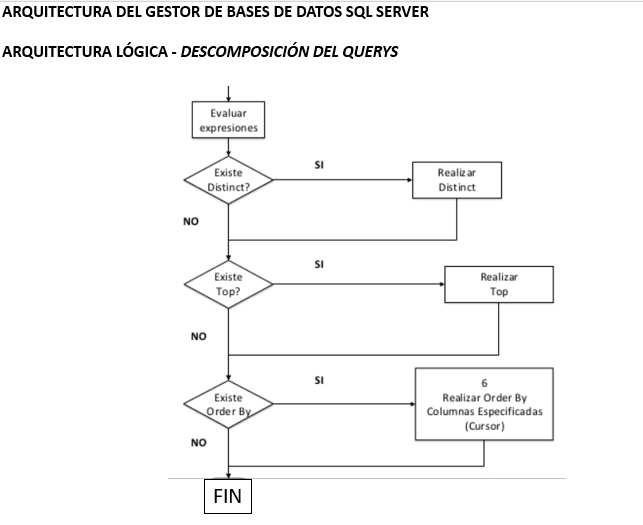


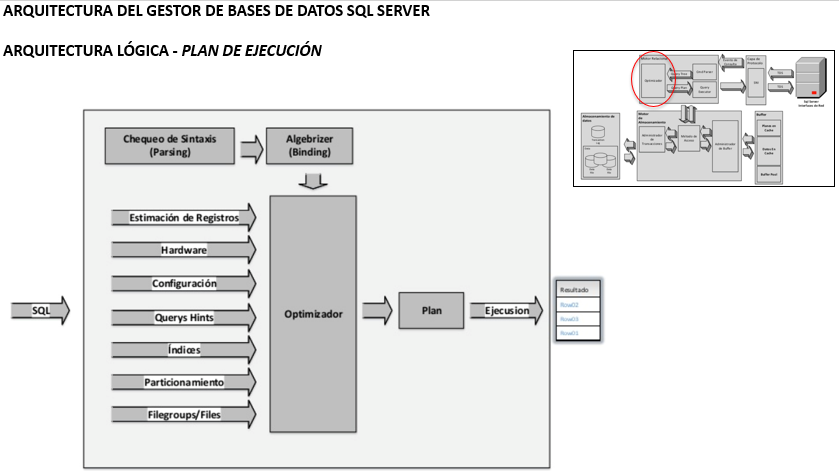


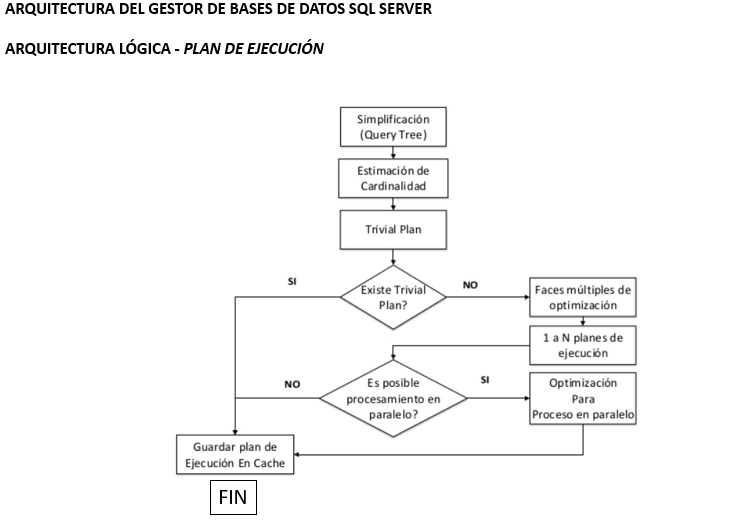


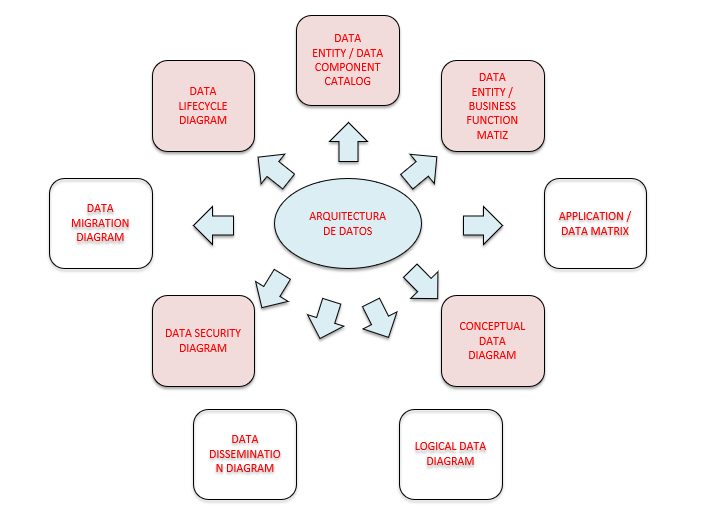










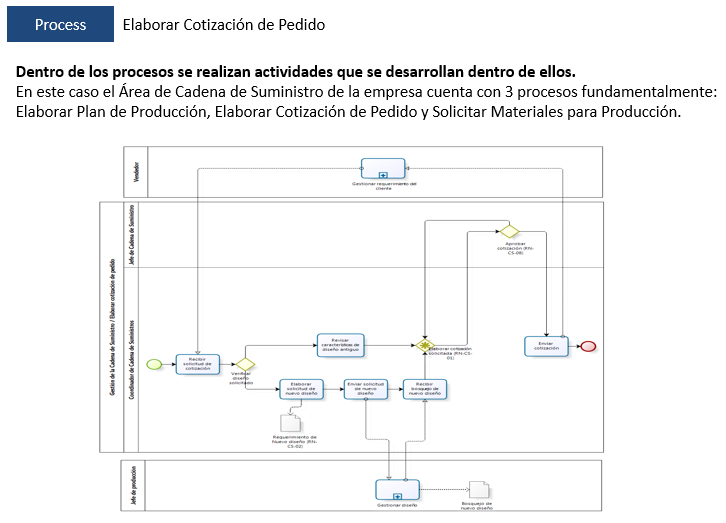


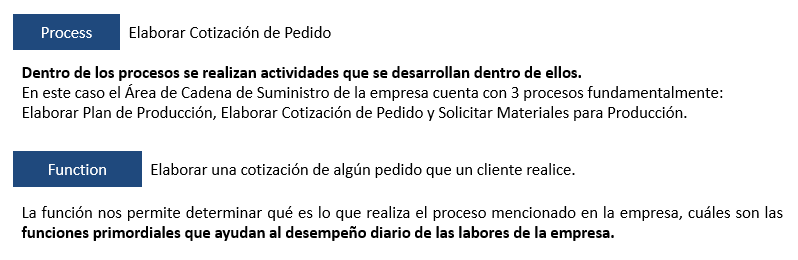
**REPASAR PPT 3 pag 29**

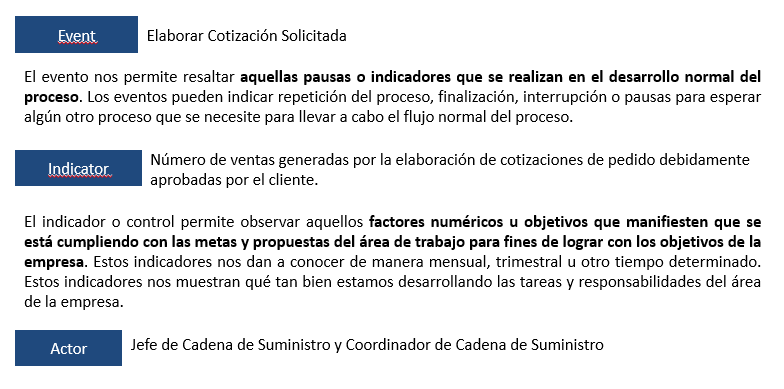
**Metamodelo**

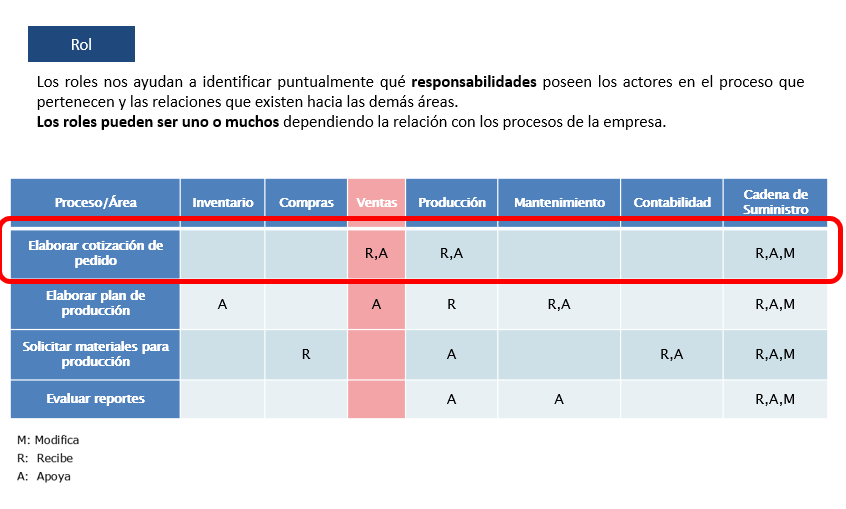
El metamodelo de un proceso relaciona todos los artefactos con la finalidad de lograr una correcta arquitectura empresarial, alineada a la visión, misión y objetivos del negocio. Asimismo, ayuda a aclarar el proceso que se ha desarrollado para poder implementar el Producto de software X de la empresa ABC.

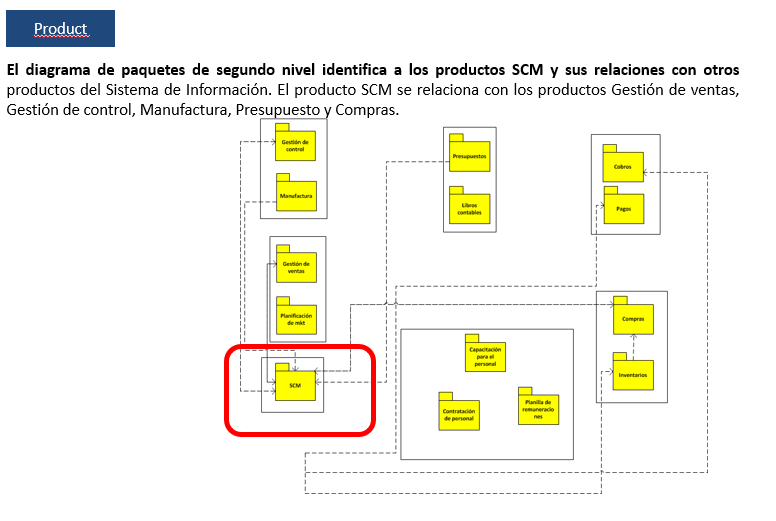


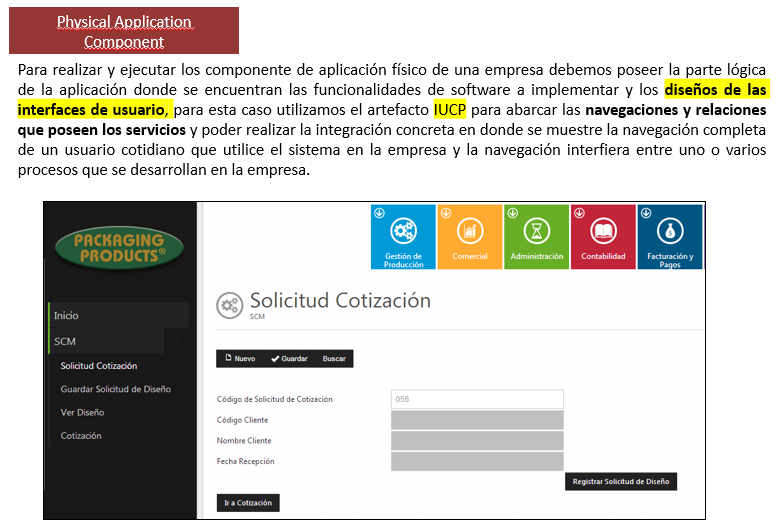


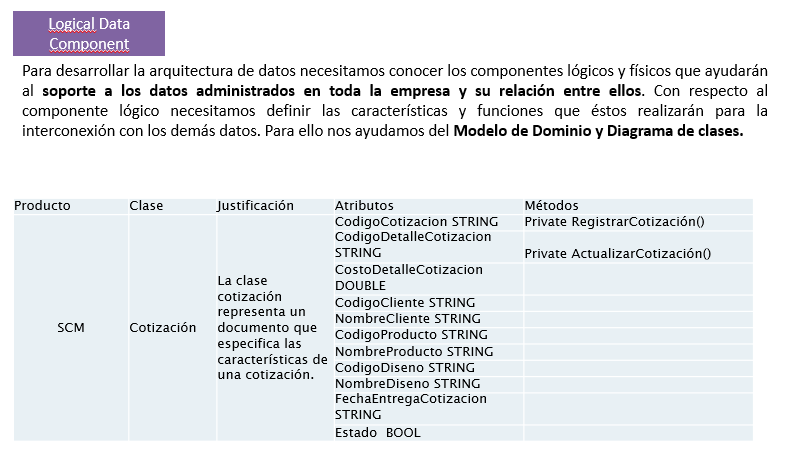


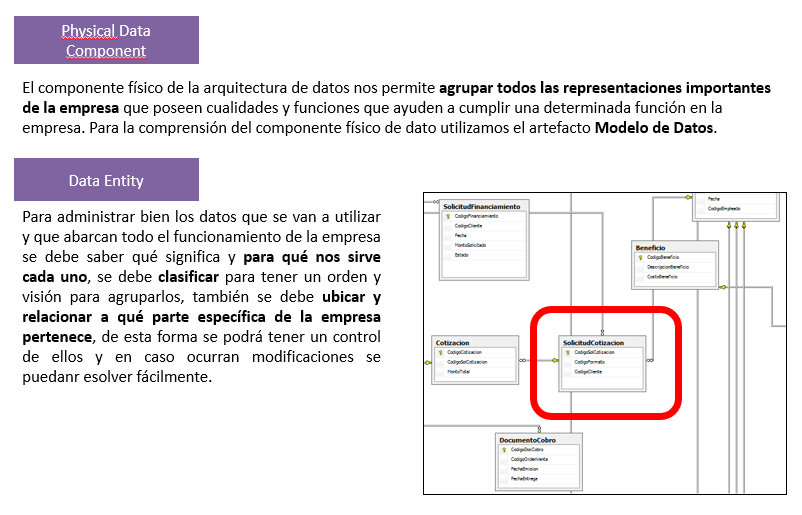


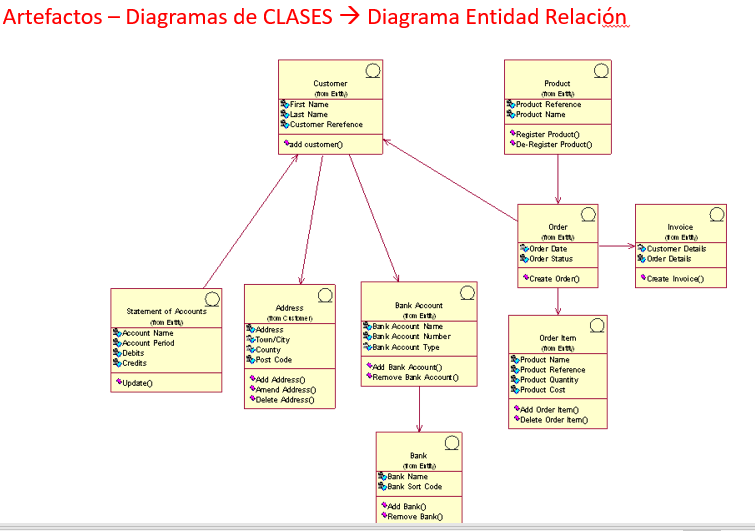








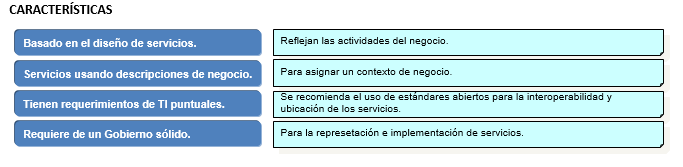


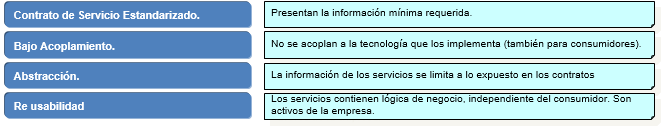
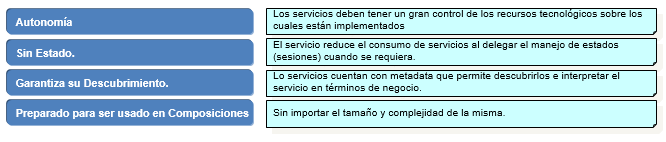


REPASAR ppt 3 pag 42

**SOA (SOA en el contexto de la AE)**

Es un estilo de arquitectura de TI que se apoya en la orientación a servicios. La orientación a servicios es una forma de pensar en servicios, su construcción y sus resultados. Un servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico.

 Proceso de Desarrollo 🡪 Análisis orientado a Servicios -> Diseño Orientado a Servicios

**Principios:**

**Terminología**

ORQUESTACIÓN 🡪 Secuenciar los servicios y proveer la lógica adicional para procesar datos. No incluye la presentación de los datos. Coordinación / Dirige servicios.

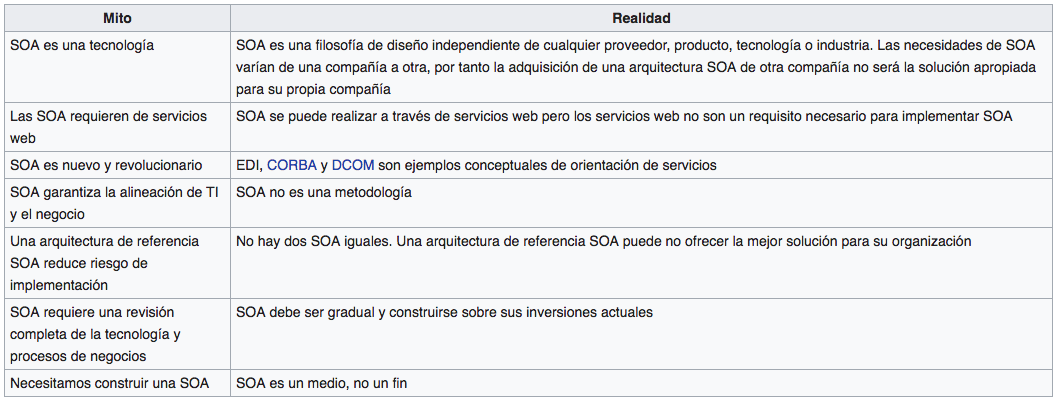
SIN ESTADO 🡪 En una SOA los servicios no son dependientes de la condición de ningún otro servicio.

PROVEEDOR 🡪 La función que brinda un servicio en respuesta a una llamada o petición desde un consumidor.

CONSUMIDOR 🡪 La función que consume el resultado del servicio provisto por un proveedor.

SOA Y SERVICIOS WEB

* SOA -> Todas las funciones están definidas como servicios independientes con interfaces invocables que pueden ser llamados en secuencias bien definidas para formar los procesos de negocio. Además en SOA la clave está en la interfaz, puesto que define los parámetros requeridos y la naturaleza del resultado. Esto significa que define la naturaleza del servicio y no la tecnología utilizada.
* Web service (WS) -> engloba varias tecnologías, incluyendo XML, SOAP, WSDL, UDDI, etc., los cuales permiten construir soluciones de programación para mensajes específicos y para problemas de integración de aplicaciones



**Topología de un serivicio 🡪** Transporte -> Como llegar al servicio (medio, protocolos, sincronía) / Tecnológicamente -> Ethernet + TCP

Interfaces Expuestas

* Son el contrato al servicio, definen como usarlo, y que brinda.
* Deben ser independientes de la plataforma

Pueden ser:

* Primarias: Las que sirven para el propósito principal del servicio
* Secundarias: Las que sirven para hacer consultas del estado del procesamiento del servicio, auxiliares, etc.

Seguridad

* Autenticación: ¿Quién es el que solicita al servicio?
* Autorización y control de acceso/Contenido filtrado: ¿Qué servicios puede utilizar?
* Single sign-on: ¿El solicitante ya fue identificado?
* Privacidad/Confiabilidad/Integridad: ¿El mensaje del servicio no podrá ser interceptado o visto?
* No repudio: ¿Se debe probar que un solicitante ha iniciado, participado o completado el uso de un servicio?

Lógica de Negocios

* Define cómo funciona el servicio, cómo responde a las solicitudes que le realizan
* El servicio es autocontenido; mantiene su estado
* Es una de las claves del reuso, dado que en esta capa se definen los detalle de implementación que le permitirá al servicio responder a diversos escenarios
* Consideraciones estándares básicas de reuso:
  + Multicanal
  + Multicliente
  + Multimoneda
  + Multiempresa

Interfases Solisitantes 🡪 Contiene las invocaciones a interfases que complementan la funcionalidad del servicio: Obtienen información, ejecuta otros servicios.

Información

* Configuración y parametrización: Mantiene la información que requiere el servicio para configurar su operación o parametrizar las reglas internas de negocio, fijar restricciones, etc.
* Log de servicios solicitados: Almacena un log de las invocaciones realizadas al servicio. Debe estar estructurado de una forma que pueda ser explotada por el negocio o por unidades de control y seguimiento
* Información interna: Permite almacenar información que es requerida para el proceso de la lógica de negocios
* El servicio no debería guardar información relacionado a entidades de negocio con una persistencia sin límites temporales; esto es ámbito de una aplicación o de un almacenamiento de datos dedicado

Operaciones

Define mecanismos que permiten la carga, recuperación y consultas del estado del servicio -> El servicio puede ser invocado y cargado dinámicamente, en tiempo de ejecución.

Tipos de Servicios

* **Lógica Pura (Pure Logic)**

Contienen proceso de lógica de negocios propio. Calculador de cuotas de crédito efectivo.

* **Motores (Engine)**

Realizan el trabajo en lotes, asíncronamente y/o procesan información previa a cumplir su propósito. Motor de pagos masivos.

* **Orquestadores (Orchestration)**

Realizan su objetivo invocando servicios externos, en cierta secuencia, típicamente asegurando la integridad (propiedades ACID). Pago a cuenta.

* **Catálogos (Catalogue)**

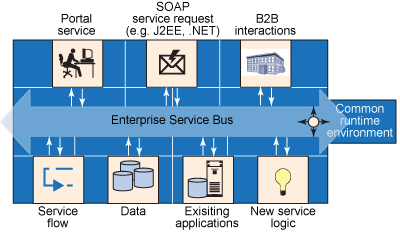
Proveen información, puntual o en bloques. Catálogo de tarifas.

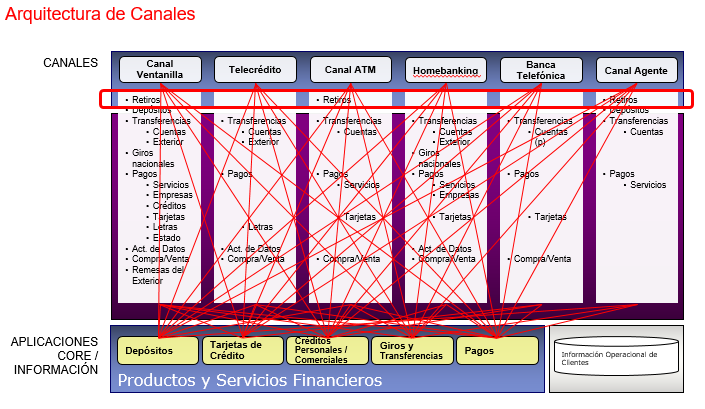
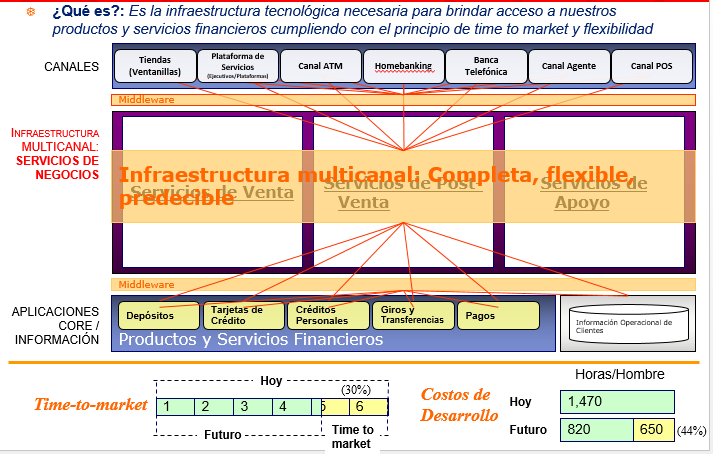
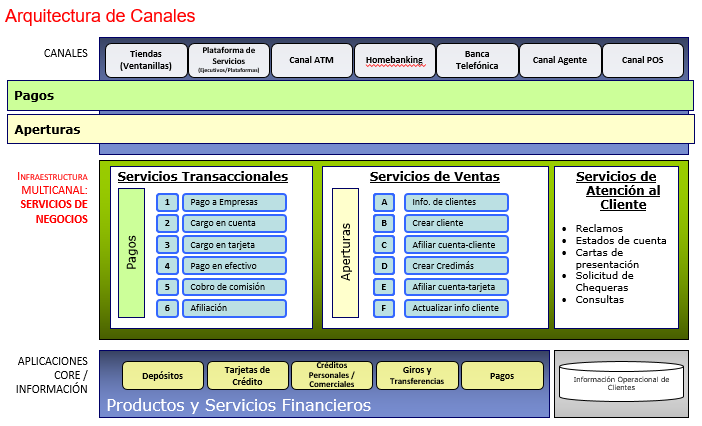
* **Fachada (Wrapper)**

Proveen sólo una máscara a servicios disponibles en otro componente de software típicamente legacy. Usualmente se usan para simplificar el acceso al legacy o darle una interfase diferente.

* **Mixtos**

Un servicio que combine características de dos o más de los tipos acá indicados.

**SOA (Arquitectura orientada a Servicios)**

 **Después**

**Antes**

**SOA (Evolución hacia SOA)**

* SOA trata sobre agilidad del negocio. Significa preparar la infraestructura IT para requerimientos de negocio desconocidos
* El negocio conduce a los servicios, los servicios a la tecnología. Los servicios son la capa de abstracción entre negocios y tecnología
* El meta-requerimiento es: agilidad del negocio. Desde la infraestructura hasta la interfase y lógica de negocios
* La metáfora de la construcción de la ciudad es incompleta. Es mejor considerar la idea de un organismo vivo
* Por ello, el trabajo del arquitecto nunca acaba

**SPA (Service Paradigm Adoption) 🡪 ¿Por qué?**

* La orientación a servicios presenta un mundo ideal con todos los recursos claramente particionados y representados en términos de servicios.
* Cada elemento o servicio, poseerá las cualidades propias de dicha representación: Débil acoplamiento, encapsulación, abstracción, reutilización, autonomía, etc…
* El Paradigma de Servicios debe ser adoptado por la arquitectura empresarial para adecuarse a dicho concepto.
* Resultados: flexibilidad, adaptabilidad y agilidad.

**SOE (Services Oriented Enterprise) ¿Con quién? ¿Qué?**

* SOE nos conduce a un cambio organizacional, nos da la base para construir una arquitectura SOA.
* Todavía no sabemos cómo vamos a realizar esos cambios y que debemos llevar a cabo; pero debemos ir identificando posibles soluciones
* Debemos identificar estrategias de aproximación (Top-Down o Bottom-Up); empezar a tener claro cuáles van a ser los servicios, como categorizarlos.

**SOA (Serivce Oriented Architecture) 🡪 ¿Cómo?**

* SOA nos presenta la manera de hacer tangible esa visión adoptada con el Paradigma de Orientación a Servicios (SPA).
* Debemos dar prioridad a los objetivos de esta visión arquitectural: Independencia de Tecnologías, Independencia del Ciclo de Vida, Débil Acoplamiento y en general cualquier propiedad que se le pueda atribuir a los servicios.

**Componentes de SOA**

* Servicios: Entidades lógicas - Contratos definidos por una o más interfaces públicas.
* Service provider: Entidad de software que implementa una especificación de servicio.
* Service consumer : Entidad de software que llama a un service provider. Tradicionalmente se lo llama “cliente”. Puede ser una aplicación final u otro servicio.
* Service broker: Tipo específico de service  
  provider que puede pasar requerimientos  
  de servicios a otros service providers.

**SOC (Serivces Oriented Computing) 🡪 ¿Con qué?**

* La granularidad de un servicio puede variar y además éste puede estar disponible desde una sola máquina o de forma distribuida.
* Los Servicios Web proporcionan una forma eficiente de representación del Paradigma inicial (SPA) por medio de su representación (WSDL), su interacción (SOAP), su descubrimiento y registro (via UDDI) e incluso su reutilización (Coreografía de Servicios como BPEL).

**STP (Services Transition Plan) 🡪 ¿Cuándo?**

* Generalmente la aplicación de los anteriores conceptos requiere de un plan de adquisición de los mismos.
* Deben estructurarse las distintas fases necesarias, la transición entre cada una de ellas, los hitos necesarios e incluso los planes de contingencia que se consideren oportunos dependiendo de los niveles de criticidad.

**Microservicios**

Los microservicios son una arquitectura y un enfoque sobre la escritura de software en el que las aplicaciones se dividen en componentes más pequeños e independientes entre sí. A diferencia de un enfoque tradicional y monolítico sobre las aplicaciones, en el que todo se crea en una única pieza, los microservicios están separados y funcionan conjuntamente para llevar a cabo las mismas tareas. Cada uno de estos componentes, o procesos, son los microservicios. Este enfoque sobre el desarrollo de software valora la granularidad por ser liviana y la capacidad de compartir un proceso similar en varias aplicaciones.[[1]](#footnote-1)

Los microservicios son un enfoque arquitectónico para crear aplicaciones. Lo que diferencia a una arquitectura de microservicios de enfoques monolíticos más tradicionales es cómo descompone una aplicación en sus funciones principales. Cada función se denomina servicio y se puede construir e implementar de forma independiente, lo que significa que los servicios individuales pueden funcionar (y fallar) sin afectar negativamente a los demás.

Piense en su última visita a un minorista en línea. Es posible que haya utilizado la barra de búsqueda del sitio para buscar productos. Esa búsqueda representa un servicio. Tal vez también vio recomendaciones para productos relacionados: recomendaciones extraídas de una base de datos de preferencias de compradores. Eso también es un servicio. ¿Agregaste un artículo a un carrito en línea? Lo has adivinado, otro servicio.

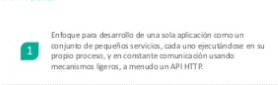
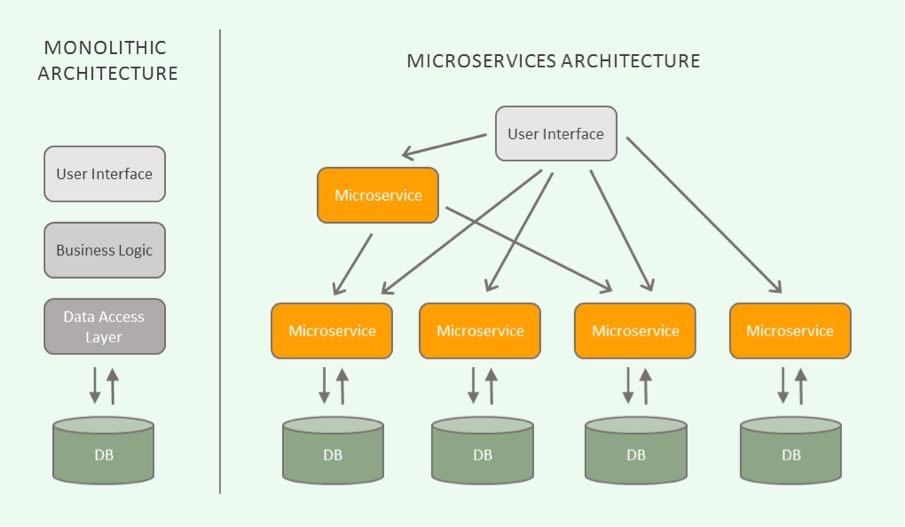
Por lo tanto, un microservicio es una función central de una aplicación y se ejecuta de forma independiente de otros servicios, pero una arquitectura de microservicios es más que un simple acoplamiento de las funciones centrales de una aplicación: se trata de reestructurar equipos de desarrollo y comunicación interservicios de una manera que prepare para fallas inevitables, escalabilidad futura y nueva integración de funciones.[[2]](#footnote-2)

Diferencias entre SOA y microservicios

El estilo de arquitectura de Microservicios es una evolución de SOA, ya que tienen bastante en común, pero con algunas diferencias:

En SOA, los servicios se exponen a través de esquemas y contratos, en cambio en Microservicios, los servicios se exponen generalmente mediante RESTful API.

En SOA el concepto de “orquestación” de servicios está muy presente, ya que un ESB permite construir servicios de negocio complejos mediante la coordinación de múltiples servicios individuales. En cambio, en Microservicios, en lugar de orquestación, se prefiere a la “coreografía” de servicios, esto es, cada servicio sabe cómo actuar en cada momento, e interactúa con otros, en lugar de tener a alguien que los coordine.

En Microservicios, según la definición de Martin Fowler y James Lewis, “Estos servicios son…desplegados de manera independiente a través de una maquinaria de despliegue completamente automatizada“. No sucede lo mismo con SOA, ya que generalmente los servicios son deplegados manualmente; y tal vez, esta sea una de las principales diferencias entre SOA y Microservicios que trae otros impactos importantes que veremos más adelante.

**Ejemplos de microservicios**

Multitud de webs que sirven aplicaciones a gran escala han decidido invertir en la evolución hacia los microservicios en vistas de un futuro donde el mantenimiento y escalabilidad de sus productos es mucho más simple, efectivo y rápido.

* Netflix: Esta plataforma tiene una arquitectura generalizada que desde hace ya un par de años se pasó a los microservicios para el funcionamiento de sus productos. A diario recibe una media de mil millones de llamadas a sus diferentes servicios y es capaz de adaptarse a más de 800 tipos de dispositivos mediante su API de streaming de vídeo, la cual para ofrecer un servicio más estable, por cada solicitud que le pedimos, ésta realiza cinco solicitudes a diferentes servidores para no perder nunca la continuidad de la transmisión.
* Amazon: No soporta tantos dispositivos como Netflix, pero tampoco es que sea fundamental para cubrir su sector. Migró hace tres años a la arquitectura de microservicios siendo una de las primeras grandes compañías que la implementaban en producción.[[3]](#footnote-3)

1. Cfr. Redhat 2018 [↑](#footnote-ref-1)
2. Cfr. Redhat 2 2018 [↑](#footnote-ref-2)
3. Cfr. OpenWebinars 2016 [↑](#footnote-ref-3)