

Guía de referencia de administración de recursos (no oficial)

Fecha	Autor	Comentario	Versión
12-06-2015	Santiago Perez Torre con colaboración de Martin Montenegro y Matías Sas	Inicial	1.0
18-06-2015	Santiago Perez Torre	Arreglo los links que no estaban	1.0.1
18-06-2015	Nacho Kiman	Pequeñas correcciones	1.0.2
18-07-2015	Nicolás Monzón	Correcciones	1.0.3
15-07-2015	Nicolás Arias	Correcciones y agregados en Arquitectura de Software, Atributos de Calidad y Bases de Datos.	1.1
21-07-2015	Santiago Perez Torre	Agregados en Virtualización	1.2
21-07-2015	Santiago Perez Torre	Pequeños cambios en Cloud Computing	1.2.1
10-09-2015	Santiago Perez Torre	Agrego temas del segundo parcial y cambios de formato	2.0.0
05-11-2015	jonybuzz	Mejoras en gestión de riesgos y verificaciones de auditoría	2.1
29-11-2015	Santiago Perez Torre	Pequeñas correcciones y mejoras	2.1.1
16-12-2015	Santiago Perez Torre	Mejoro la idea de gestión de recursos humanos	2.1.2
17-02-2016	Santiago Perez Torre	Mejoras generales en plan de contingencia,	2.1.3

		atributos de calidad, estimaciones, amortización, plan de negocio y proyecto	
02-03-2016	Santiago Perez Torre	Mejoro la idea de VPN	2.2
02-03-2016	Santiago Perez Torre	Mejoro idea de Message Broker Pattern	2.2.1
05/03/2016	Santiago Perez Torre	Algunas cositas en Atributos de calidad	2.2.2
09/07/2016	Santiago Perez Torre	<p>Agrego ejemplo de MEP.</p> <p>Mejoro ideas de PV/EV/AC.</p> <p>Agrego paravirtualización y full virtualización y mejoro hipervisor.</p> <p>Agrego ideas de económico y financiero en impuestos.</p> <p>Mejoro leasing y agrego ejemplo.</p> <p>Mejora en pasos de auditoría.</p> <p>Mejoro Capex y Opex con ejemplo.</p> <p>Mejoro RTO y RPO.</p> <p>Cambios en virtualización.</p> <p>Reordenamiento de contenido y cambio de estilos.</p>	3.0.0-A
11/07/2016	Santiago Perez Torre	Agrego Datacenter	3.0.0-B
11/07/2016	Santiago Perez Torre	<p>Corrijo virtualización.</p> <p>Agrego de Datacenter a Paas.</p> <p>Mejoro Broker Pattern.</p> <p>Mejoro DAS, SAN,</p>	3.0.0-C

		NAS. Mejoro comparación de Mainframe con Supercomputadora. Mejora al Teorema CAP	
13/07/2016	Santiago Perez Torre		3.0.0

Aclaraciones

Este documento tiene como objetivo ser una guía de referencia o servir como otro punto de vista para los temas que se evalúan en la materia. No necesariamente están incluidos todos los temas que evalúan en la materia. No necesariamente los temas están bien explicados en este apunte. No necesariamente las explicaciones de este apunte coinciden con las explicaciones que brinda la cátedra.

Índice

- [Aclaraciones](#)
- [Índice](#)
- [Benchmark](#)
- [Direct Attach Storage \(DAS\)](#)
- [Storage Área Network \(SAN\)](#)
- [Network Attached Storage \(NAS\)](#)
- [Comparaciones entre dispositivos](#)
 - [Comparación entre DAS y SAN](#)
 - [Comparación entre SAN y NAS](#)
 - [Comparación entre DAS y NAS](#)
- [Conclusión de dispositivos de almacenamiento](#)
- [Tape drives](#)
- [Plan de contingencia](#)
 - [Tipos de planes](#)
 - [Planes de protección de datos](#)
 - [Plan de respaldo / protección](#)
 - [Plan de emergencia](#)
 - [Plan de recuperación](#)
- [Pasos](#)
 - [Análisis de riesgos/amenazas](#)
 - [Determinación del plan de contingencia](#)
 - [Revisión y actualización](#)
- [Recovery Time Objective \(RTO\)](#)
- [Recovery Point Objective \(RPO\)](#)
- [Ejemplo RTO y RPO](#)
- [Infraestructura IT](#)
- [Datacenter](#)
 - [Concepto](#)
 - [Decisiones para implementación](#)
 - [Planeación estratégica](#)
 - [Planeación de recursos](#)
 - [Planeación operativa](#)
 - [Planeación de instalaciones](#)
 - [Planeación de personal](#)
- [Estándar TIA](#)

- [Impacto de la caída del datacenter](#)
- [Mainframe](#)
 - [Concepto](#)
 - [Características](#)
- [Supercomputadora](#)
 - [Concepto](#)
 - [Características](#)
- [Comparación entre mainframe y supercomputadora](#)
- [Servidor hardware](#)
 - [Concepto](#)
 - [Ventajas](#)
 - [Comparación de tipos \(torre/rack/blade\)](#)
- [Clúster](#)
 - [Concepto como algo lógico](#)
 - [Concepto como algo físico](#)
 - [Concepto según ADR](#)
 - [Ventajas](#)
 - [Desventaja](#)
 - [De alta disponibilidad\(HA-C\)](#)
 - [De alto rendimiento \(HP-C\)](#)
 - [De balanceo de cargas \(LB-C\)](#)
 - [Middleware de un clúster](#)
- [Grid computing](#)
 - [Concepto como algo lógico](#)
 - [Concepto como algo físico](#)
 - [Ventajas](#)
 - [Desventajas](#)
 - [Conflictos](#)
 - [Middleware de un grid](#)
- [Cloud computing](#)
 - [Concepto como algo lógico](#)
 - [Concepto como algo físico](#)
 - [Concepto](#)
 - [Ventajas](#)
 - [Desventajas](#)
 - [Infrastructure as a Service \(IaaS\)](#)
 - [Platform as a Service \(PaaS\)](#)
 - [Software as a Service \(SaaS\)](#)
 - [De Datacenters a PaaS](#)
- [Comparación entre clúster, grid computing y cloud computing](#)
- [Conclusión](#)
- [Comparación entre opex y capex](#)
 - [Opex](#)
 - [Capex](#)
 - [Conclusión](#)
- [Software base](#)
 - [Características al momento de elección](#)
- [Virtualización](#)

- [Virtualización de servidores](#)
- [Virtualización de acceso](#)
- [Virtualización de aplicaciones](#)
- [Virtualización de procesamiento](#)
- [Virtualización de almacenamiento](#)
- [Virtualización de red](#)
- [Paravirtualización vs Full virtualización](#)
- [Base de datos Relacionales](#)
 - [Características al momento de la elección de base de datos relacional](#)
- [Bases de datos NoSQL](#)
 - [Diferencias con las relaciones](#)
 - [Características de DB NoSQL](#)
 - [Teorema de CAP](#)
- [Estimaciones](#)
 - [Medida](#)
 - [Métrica](#)
 - [Indicador](#)
 - [Puntos de función](#)
 - [Concepto](#)
 - [Entradas externas](#)
 - [Salidas externas](#)
 - [Consultas externas](#)
 - [Interfaces externas](#)
 - [Archivos lógicos internos](#)
- [Juicio experto](#)
- [Wideband](#)
- [Estimación análoga \(TOP-DOWN\)](#)
- [Estimación detallada \(BOTTOM-UP\)](#)
- [Estimación paramétrica](#)
- [Estimación por tres valores \(PERT\)](#)
- [Arquitectura de software](#)
 - [Etapas](#)
 - [Ventajas](#)
 - [Patrones \(ppt vieja\)](#)
 - [Monocapa](#)
 - [Cliente-Servidor](#)
 - [Multicapa](#)
 - [Patrones](#)
 - [Concepto](#)
 - [Layered Pattern](#)
 - [Message Broker Pattern](#)
 - [MVC Pattern](#)
 - [Client-Server Pattern](#)
 - [Service-Oriented Architecture Pattern](#)
 - [MapReduce Pattern](#)
- [Atributos de calidad](#)
 - [Adaptabilidad](#)
 - [Disponibilidad](#)

- [Escalabilidad](#)
- [Interoperabilidad](#)
- [Performance](#)
- [Seguridad](#)
- [Usabilidad](#)
- [Conclusión](#)
- [Telefonía IP \(VoIP\)](#)
- [VPN](#)
- [Recursos humanos](#)
 - [Gestión de recursos humanos](#)
- [Liderazgo](#)
- [Cambio](#)
 - [Niveles](#)
 - [Gestión del cambio](#)
 - [Fuerzas impulsoras](#)
 - [Fuerzas restrictivas](#)
- [Negociación](#)
- [Plan de negocio](#)
- [Impuestos](#)
 - [Gravar](#)
 - [Económicos vs Financieros](#)
 - [Directos](#)
 - [IVA](#)
 - [IDCB \(al cheque\)](#)
 - [IIBB](#)
 - [Indirectos](#)
 - [Ganancias](#)
 - [Ganancia mínima presunta](#)
- [Amortización](#)
- [Leasing](#)
- [Abastecimiento](#)
 - [Fases del abastecimiento](#)
- [Método de evaluación de propuestas \(MEP\)](#)
 - [Pasos](#)
 - [Punto de ponderación](#)
- [Proyecto](#)
 - [Fases de un proyecto](#)
 - [Fallos de un proyecto](#)
 - [Gestión de proyectos](#)
- [Gestión de valor ganado \(EVM\)](#)
- [Riesgo](#)
 - [Gestión del riesgo](#)
 - [Tratamiento de riesgos](#)
 - [Problema](#)
- [Indicadores de proyectos](#)
 - [VAN: valor actual neto](#)
 - [TIR: tasa interna de retorno](#)
 - [PRI: período de recuperación de la inversión](#)

[ROI: costo beneficiario](#)

[Auditoría](#)

[Verificaciones a los sistemas](#)

[Objetivos de la auditoría](#)

[Auditoría de sistemas](#)

[Mecanismos de control a evaluar](#)

[Pasos de una auditoría](#)

[Controles de la auditoría a los sistemas](#)

[Áreas que revisa la auditoría de sistemas](#)

[Controles internos](#)

Benchmark

Es un **proceso** -seguidilla de pasos- que **compara** elementos -de la misma familia- en base a características predefinidas y **arroja resultados** -que se pueden comparar porque están en la misma unidad- que ayudan en la **toma de decisiones**.

El benchmark es un proceso que tiene una entrada y una salida como todos los procesos. Su entrada es un producto a “benchmarkear” y su salida es un valor. Ese valor se calcula a partir del análisis de ciertas características predefinidas (por ejemplo, si quiero hacer un benchmark de una pc, las características podrían ser velocidad de procesamiento, velocidad de acceso a disco, entre otras).

Si yo quiero comparar varios productos “de la misma familia” (es decir, comparar pcs, comparar discos, comparar parlantes), puedo aplicarles a todos los productos el benchmark. Este te va a dar un valor por cada uno de ellos que vas a poder usar para poder compararlos ya que “están en las mismas unidades”.

¿Para qué sirve? El principal uso es para comparar productos y saber cuál es mejor. Pero hay otros usos. Uno de ellos es obtener una relación de costos/beneficio. ¿Cómo? Si al valor que te da el benchmark, lo dividís por lo que sale el producto, te va a terminar quedando “cuanto valor” te da ese producto por unidad monetaria. Por ejemplo, si evaluamos placas de video, podríamos obtener un valor que sea cuantos fps te da la tarjeta por cada peso que pones. Por último, también puedes usar el benchmark para saber si el producto te sirve o no, es decir, por ahí no te sirve que un producto no cumpla con determinada característica o no alcance cierto valor.

Algunas de las **características** son:

- Sistemático (que sigue un proceso, metódico)
- Está en continua evaluación y corrección (los fallos que cometí hoy al realizar el benchmark, son las mejoras del próximo).
- Devuelve un resultado objetivo (ya que es el resultado de un algoritmo o proceso)
- Te dan un resultado de mejor relación costo/beneficio.

Las **etapas** para la creación y uso de benchmark es:

1. Determinar el elemento de estudio (definir el qué, las características a analizar y las opciones en el mercado)
2. Preparar el entorno (generar el entorno de prueba, lo más parecido a la realidad en donde va a funcionar ese producto. Por ejemplo, si quieres hacer un benchmark para un disco para tu pc, la realidad va a ser tu pc. En cambio, si es para la pc de tu primo, la realidad va a ser la de tu primo)

3. Realizar el benchmark (correr el algoritmo, tomo muestras parciales y finales)
4. Analizar los resultados (comparar las muestras y generar reportes/informes)
5. Analizar, proponer y realizar correcciones al algoritmo (para recalibrarlo o calibrarlo mejor)

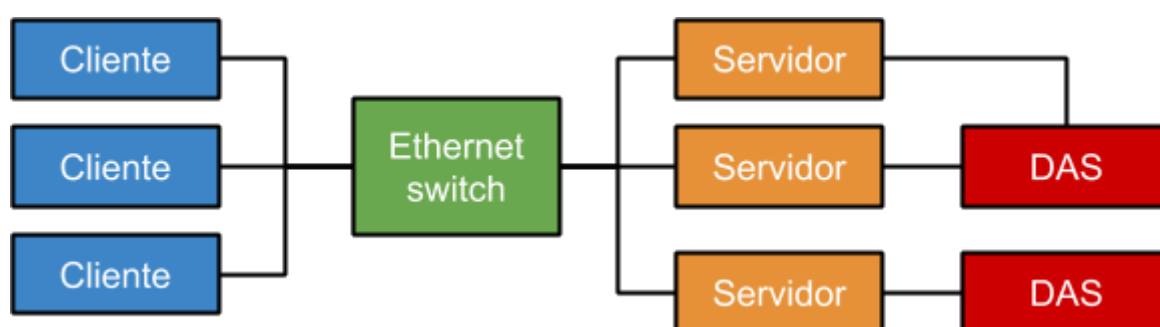
Direct Attach Storage (DAS)

Sistema de almacenamiento en el cual el dispositivo de almacenamiento se conecta físicamente y de forma directa al servidor.

Si bien la idea es la misma que un disco duro o externo -un DAS personal-, en las empresas no termina siendo así por las siguientes razones (puede haber más): poco espacio de discos y poca posibilidad de gestión.

Lo que usan las empresas son soluciones mejor preparadas. Es decir, los DAS son como cajas con compartimentos. En cada compartimiento pones un disco común (el mismo que pones en tu PC pero con mucha capacidad). Lo que te provee esta caja, es la conectividad de todos esos discos, y además, la posibilidad de gestionarlos eficaz y eficientemente. ¿Que quiero decir con esto? Básicamente que te da una interfaz para crear RAIDs de forma sencilla, para definir políticas en caso de falla, para realizar particiones, etc.

También tene en cuenta que la transmisión de los datos es importante, por eso se usan protocolos e interfaces -es decir, los conectores- particulares como: SCSI, Serial Attached SCSI (SAS), Serial ATA (SATA) o GigaEthernet



Para que te des una idea de ejemplos de DAS fijate lo siguiente:

- <http://www.dell.com/us/business/p/direct-attached-storage>
- <http://shop.lenovo.com/us/en/systems/storage/das/lenovo-e1024/>

Las **ventajas** pueden ser:

- Buen a alto rendimiento en transmisión (dependen del tipo de dispositivo y las interfaces que uses)

- Relativo bajo costo (dependen del tipo de dispositivo o soluciones empresariales que uses)
- Simplicidad (es conectar los componentes y definir políticas de gestión)
- Facilidad de mantenimiento (ya que no incorporas redes como en las demás opciones)
- Pueden soportar virtualización (dependen del tipo de dispositivo que uses. Esto aumenta la eficiencia por las ventajas que aporta la virtualización)

Las **desventajas** pueden ser:

- No se puede compartir la información de forma sencilla (entre servidores que no están conectados a través del DAS)
- Solo sirve para servidores con los que está conectado físicamente y de forma directa
- Mayor complejidad para conectar una PC a varios DAS

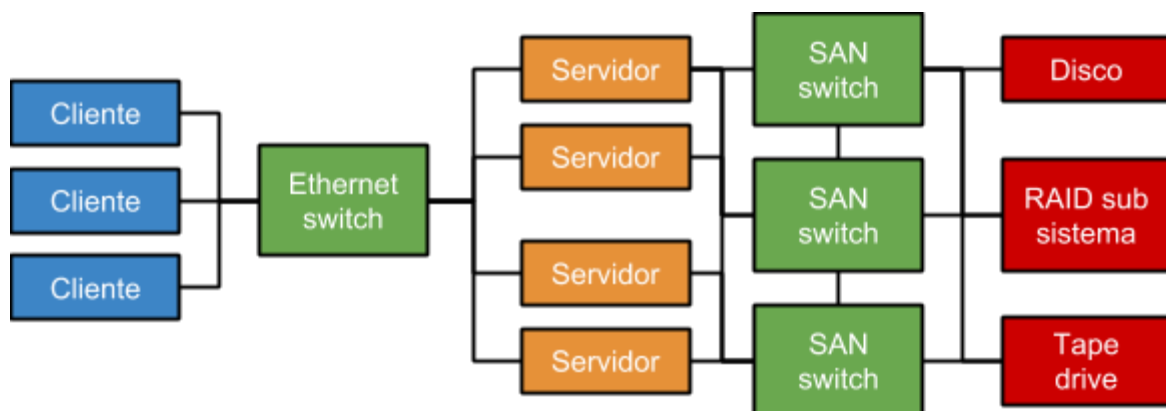
Fuentes:

- <http://es.slideshare.net/alex.avila1976/almacenamiento-das-direct-attached-storage>
- <http://whatis.techtarget.com/reference/Fast-Guide-to-Storage-Technologies>
- http://es.wikipedia.org/wiki/Direct_Attached_Storage
- http://www.webopedia.com/TERM/D/direct_attached_storage.html

Storage Área Network (SAN)

Sistema de almacenamiento en donde el dispositivo se conecta a una red SAN (de alta velocidad y de protocolos propios, es decir, no sobre internet) que es compartida con los servidores.

Sistema de almacenamiento en el cual existe un pool de dispositivos de almacenamiento que son accedidos por múltiples servers a través de una intranet de alta velocidad.



Las **ventajas** pueden ser:

- Buen a alto rendimiento (depende de la configuración que le demos a la red y los componentes que usemos)
- Mayor flexibilidad para agregar nuevos servidores y dispositivos SAN.
- Puede transferir información entre dispositivos de almacenamiento sin intervención de los servidores.
- Al estar todo en la misma red, la administración suele ser más sencilla.

Las **desventajas** pueden ser:

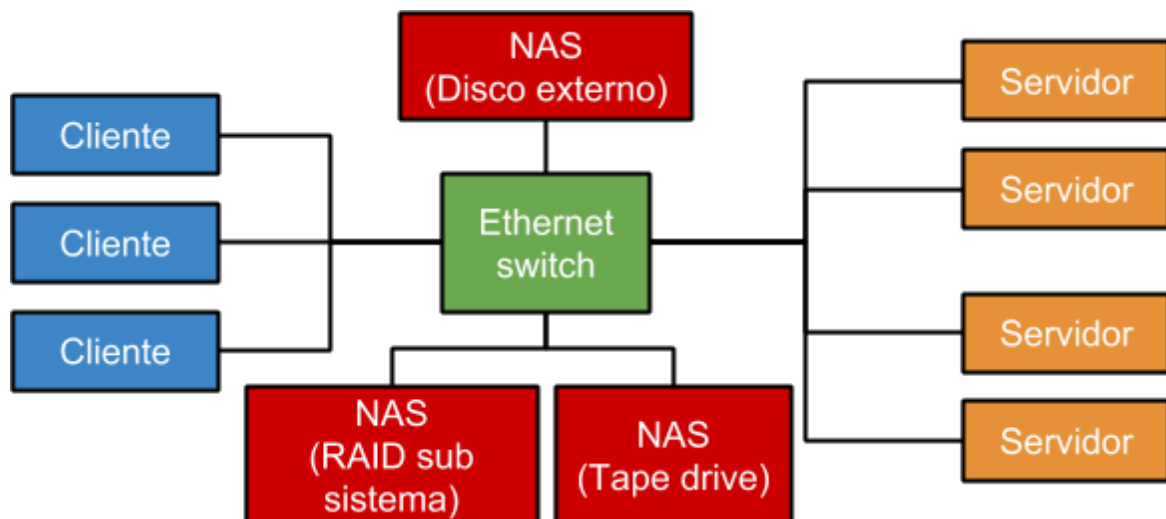
- Costoso (el hardware al ser muy específico, termina siendo muy caro. O las soluciones empresariales también terminan costando un montón, básicamente porque la solución SAN es compleja)
- Complejidad (tenes que entenderlo muy bien para poder desarrollarlo. O comprar soluciones de empresas que se dediquen a esto)

Fuentes:

- <http://searchstorage.techtarget.com/definition/storage-area-network-SAN>
- <http://www.webopedia.com/TERM/S/SAN.html>
- http://www.snia.org/education/storage_networking_primer/san/what_san

Network Attached Storage (NAS)

Sistema de almacenamiento en donde el dispositivo de almacenamiento se conecta a una red (LAN, WAN con protocolos HTTP, FTP, NFS) que es compartida con los clientes y los servidores.



Las **ventajas** pueden ser:

- Muchos clientes pueden acceder a los mismos archivos
- Muchos servidores pueden acceder a los mismos archivos
- Relativo bajo costo (dependen del tipo de dispositivo que uses)
- Relativa facilidad de gestión (ya que te olvidas de la parte de redes porque es una infraestructura que ya tenes armada)
- Mayor flexibilidad (te permite agregar servidores y NAS con facilidad)

Las **desventajas** pueden ser:

- Buen a bajo rendimiento (depende mucho del tipo de red que utilices y la velocidad a la que puedan correr los datos, la cantidad de clientes y servidores conectados, el dispositivo NAS. Ojo acá pasa un poco lo mismo que DAS, depende mucho de los componentes que utilices)

Fuentes:

- <http://searchstorage.techtarget.com/definition/network-attached-storage>

Comparaciones entre dispositivos

Comparación entre DAS y SAN

Ventajas de DAS

- Menos costoso

- Menos complicado (porque requiere de una gran planeación por temas de escalabilidad)

Ventajas de SAN

- Más flexibilidad (deja agregar más dispositivos de almacenamientos)
- Mejor disponibilidad (porque hay mayor cantidad de dispositivos de almacenamiento, se puede aplicar mejores políticas de recovery and restore)
- Mejor performance (depende mucho del tipo de conexión que uses)
- Mejor uso de los dispositivos de almacenamiento (ya que todos usan todos, tiende a haber una mejor utilización de todos los recursos)
- Poder compartir información (antes si el servidor A quería acceder a la info del servidor B, éste último debía copiar su info en el servidor A. Con SAN, el servidor A puede acceder a la info del servidor B vía la red SAN)

Comparación entre SAN y NAS

Ventajas de SAN

- Maneja los pedidos a nivel bloque
- Generalmente más performantes (porque están diseñados para correr sobre Fiber Channel)
- El servidor maneja el file system (ve al dispositivo como un disco)
- Manejan protocolos propios (no dependen de la red de Internet)
- Alta eficiencia

Ventajas de NAS

- Maneja los pedidos a nivel archivo
- Mejor adaptabilidad (puede correr sobre Ethernet, Fiber Channel)
- El dispositivo maneja el File System (el servidor ve al dispositivo como un File System)
- Manejan protocolos públicos (que ya están desarrollados y testeados y son compatibles con muchos componentes)
- Poca eficiencia

Fuentes:

- http://www.snia.org/education/storage_networking_primer/san/closing

Comparación entre DAS y NAS

Ventajas de DAS

- Mejor performance (porque tiene una conexión física directa)

Ventajas de NAS

- No requiere conexión física directa a la PC (el dispositivo puede estar conectado a un switch o router y la PC puede acceder a él a través de la red)
- No requiere que cada dispositivo sea administrado por separado
- Facilidad para compartir archivos (porque los dispositivos no están relacionados con un solo servidor, sino que están conectados a la red y pueden ser accedidos por todos los que estén conectados a ella. En cambio, al DAS solo pueden acceder los que están conectados física y directamente a él)

Fuentes:

- http://www.snia.org/education/storage_networking_primer/san/closing

NAS	iSCSI	SAN
+ Bajo Costo + Tarjeta NIC + LAN existente - Rendimiento Bajo - Poco Eficiente o CIFS/NFS/HTTP/FTP - No permite Boot-from-NAS	+ Bajo Costo o Tarjeta NIC o TOE* + LAN existente o Rendimiento Medio o Eficiencia Media + Permite Boot-from- iSCSI	- Costo Alto - Tarjeta HBA - SAN Switches + Rendimiento Alto + Muy Eficiente + Muy Escalable + Permite Boot-from-SAN

Conclusión de dispositivos de almacenamiento

Al principio, los dispositivos de almacenamiento soportaban lecturas y escrituras. Con el tiempo fueron evolucionando y fueron agregando cosas como redundancia (RAID) o mejores file systems. En un punto se dieron cuenta de que podrían incorporar procesadores para optimizar la lectura y escritura. De esta manera, pasaron a ser servidores diseñados para almacenar información de forma eficiente. Al agregar los procesadores, habilitaron al dispositivo de almacenamiento a soportar virtualización, con todos los beneficios que esta tecnología conlleva -ver Virtualización-.

Existen tres formas de conectar dispositivos de almacenamiento con servidores. La primera es conectarlo físicamente al servidor (DAS), no dejándote compartir la información entre los servidores. La segunda es conectarlo a una red LAN en donde también estén conectados los servidores (NAS). La tercera armar una red especial de alta velocidad entre los servidores y los dispositivos de almacenamientos. Cada arquitectura tiene sus beneficios y costos, por lo que hay que saber elegir la mejor para cada situación.

Tape drives

Dispositivo de almacenamiento que lee y escribe datos en una cinta magnética para almacenamiento offline de archivos.

La tecnología está desarrollada por una asociación que se encuentra en funcionamiento. Se llama Linear Tape-Open (LTO) y está compuesta por IBM, HP, Quantum y Seagate. La última generación LTO fue la 7 y fue lanzada en Diciembre de 2015.

Los tape drives no se usan como discos externos, sino que se los usan en drives -que son “cajas” donde podés ir metiendo tape drives-.

En general, se utilizan como “offline back-up”, es decir, guardas cosas que tenes que guardar pero que no vas a acceder en mucho tiempo. Para lo otro, se utilizan soluciones con discos, ya que son más performantes.

Ventajas (comparado con discos):

- Menor costo por byte o mayor capacidad por igual precio
- Mayor resistencia al paso del tiempo en iguales condiciones (en general, como todo)
- Mayor simpleza (pueden fallar menos cosas)

Desventajas (comparado con discos):

- Menor performance en tiempos de búsqueda (aunque se desarrolló un file system especialmente diseñado para los LTO, los tiempos de búsqueda son mayores que los discos)
- Mayor atención en el mantenimiento (ya que no le debería entrar polvo ni nada. Igual esto es relativo, porque se puede tener un buen proceso de mantenimiento, por lo que no sería necesario una buena atención)
- Menor performance en recovery (tarda más tiempo en volver a copiar los archivos al sistema que los discos)

Para los que creen que la tecnología es obsoleta, les muestro algunos links de productos que hoy se venden:

- <http://www.quantum.com/products/tapelibraries/scalari6000/index.aspx> que puede llevar a los 180PB de datos
- <http://www.quantum.com/products/autoloaders/superloader3/index.aspx> que llega a los 240TB

Tipo	Características
Manuales	<ul style="list-style-type: none"> ● 1 cinta ● 1 cabezal lectura/escritura ● Operación manual
Semiautomáticos	<ul style="list-style-type: none"> ● 8 o 9 cintas ● 1 cabezal lectura/escritura ● Operación automática
Automáticos	<ul style="list-style-type: none"> ● Varios slots para cintas ● Varios cabezales de lectura/escritura ● Operación automática
Virtuales	<ul style="list-style-type: none"> ● Hace lo mismo que los automáticos, pero en vez de guardar en tapes guarda en disco ● Emula varios slots para cintas ● Emula varios cabezales de lectura/escritura ● Operación automática ● Mejor performance para realizar back-ups ya que guarda en disco y no en tape ● Mejor eficiencia para almacenar información (menor espacio por tecnologías de compresión)

Fuentes:

- <http://www.ironmountain.com/Knowledge-Center/Reference-Library/View-by-Document-Type/General-Articles/D/Disk-vs-Tape-Which-Ones-Game-for-the-Long-Term.aspx>
- <http://serverfault.com/questions/550763/what-are-the-advantages-of-tape-drives>

Plan de contingencia

Es un **plan** para reducir los efectos negativos de las situaciones de emergencia.

Es un **plan** que propone una serie de **procedimientos alternativos** a los procedimientos normales (es decir, cuando sucede un problema, lo que se hace habitualmente se deja de hacer) que buscan

garantizar la continuidad en el funcionamiento de la empresa o volver al funcionamiento normal rápidamente, ante **situaciones de emergencia o eventualidades** que alteren el funcionamiento normal.

Algunas de las **características** son:

- Dinámico (debe estar en continua actualización por mejoras o cambios en las condiciones iniciales)
- Controlado (debe estar en continuo control para determinar si en cada momento el plan va a seguir siendo efectivo)
- Preventivo (porque trata de prevenir la situación)
- Predictivo (porque está suponiendo que va a pasar algo)
- Estratégico (porque tiene una importancia decisiva)
- Operativo (porque se realiza, se lleva a cabo)

Tipos de planes

Planes de protección de datos

Entre las medidas que se pueden tomar están:

- **Respaldo en una cinta de manera externa e interna:** con costos bajos y posibilidad de transporte, pero con un alto tiempo de recupero (ver comparación con discos en Tape Drive)
- **Respaldo en disco de manera externa e interna:** más costoso y con menor posibilidad de transporte, pero con rápido acceso y facilidad de integración (ver comparación cinta en Tape Drive)
- **Respaldo en sitio externo:** mayor costo y no se necesita transporte, pero no puedo continuar la operatoria desde ahí
- **Respaldo en sitio externo como contingencia:** mucho mayores costos, pero con la posibilidad de continuar operando ante cualquier contingencia en el datacenter que falló.

Plan de respaldo / protección

Acciones preventivas para que no me afecte la amenaza.

Plan de emergencia

Acciones que se toman cuando la amenaza se materializó.

Plan de recuperación

Acciones que se toman después de que sucede la materialización de la amenaza para volver al estado anterior.

Pasos

Análisis de riesgos/amenazas

Se analizan todos los riesgos/amenazas que afectan el proceso de producción de empresa. En el análisis se tienen en cuenta cosas como la probabilidad y el impacto que podrían llegar a causar.

Por ejemplo, un riesgo podría ser un desastre natural, un incendio, pérdida de energía eléctrica, desperfecto técnico en equipos de hardware o software, etc.

Determinación del plan de contingencia

Sólo se debería hacer un plan de contingencia si el impacto se tiene que controlar y restaurar en caso de que ocurra, ya que el desarrollo del plan conlleva recursos monetarios y humanos.

Se determinan las medidas técnicas, humanas y organizativas que se llevarán a cabo ante la eventual transformación del riesgo en un problema. Se tienen en cuenta variables como RPO y RTO.

Por ejemplo, se determinan cosas como los materiales necesarios, qué personas llevarán a cabo qué tareas, qué materiales tendrán esas personas a cargo para realizar esas tareas, cuáles son los procedimientos ante el eventual desastre, quiénes son los responsables de cada procedimiento. Además, se analiza la posibilidad de mudarse a otra locación en caso de desastres, se definen las políticas de replicación de datos y la telecomunicación con la segunda locación.

Revisión y actualización

Los planes de contingencia deben ser evaluados y testeados periódicamente para mantenerlos actualizados ante nuevas amenazas y nuevas condiciones externas.

Recovery Time Objective (RTO)

Tiempo que pasará la infraestructura para estar disponible y operativa de nuevo. Incluye la detección, recuperación de datos y restauración.

Tiempo máximo que puede estar el proceso en estado no operativo sin tener un impacto negativo en la continuidad del negocio por causa de la eventualidad. Incluye la detención, el arreglo, la restauración, el testeo y la comunicación a los usuarios que todo vuelve a estar bien.

Es decir, a vos como administrador de recursos te van a decir: “mira no podemos estar más de 5 horas sin sistemas después de un desastre”. Entonces vos después vas a tener que pensar planes y diseñar la infraestructura para cumplir con esa limitación.

Recovery Point Objective (RPO)

Datos que está dispuesta a resignar la empresa ante un desastre.

Tiempo máximo que puede pasar entre el último back-up y el momento del desastre sin tener un impacto negativo en la cantidad de datos perdidos por causa de la eventualidad.

El RPO va a ser mayor o menor según cuánto te cueste recuperar a vos esa información. Pensá que estás escribiendo un libro en una computadora. Cada tanto vas guardando los cambios porque te querés cuidar ante un posible fallo en la computadora. Pero, ¿cada cuánto? Y un poco vas a decir por lo que te cuesta volver a escribir lo que perdiste desde la última vez que guardaste. Por ejemplo, puedes guardar cada vez que terminas un capítulo (RTO va a ser igual a un capítulo), pero ¿es correcto eso? ¿Te va a ser fácil volver a escribir ese capítulo? Y depende, si los capítulos son cortos, tal vez sea correcto, si son más largos, tal vez no. En este caso vos decidís cuánto estás dispuesto a perder, pero en una empresa, deciden por vos, y a vos te dicen: “mira no podemos perder más de medio capítulo” y vos tenés que diseñar y planear en consecuencia.

Ejemplo RTO y RPO

Toda la información que no fue backupeada es información no usable. Si ocurre un desastre o falla, toda la información que no fue backupeada se considera perdida. Entonces, toda la información perdida es la resta entre la información desde que empezó el desastre o falla menos la información del último backup. Este resultado es el que hay que tener en cuenta para calcular la frecuencia de los back-ups. Si yo digo que no puedo perder más de 1GB de información -mi RPO- y produzco 1MB por minuto, entonces mi frecuencia va a ser igual a $1\text{GB} / 1\text{MB/s} = 1024$ minutos. Por lo tanto, voy a tener que hacer un back-up cada 1024 minutos o menos.

Fuentes:

- <http://definicion.de/plan-de-contingencia/>
- http://en.wikipedia.org/wiki/Recovery_point_objective
- http://en.wikipedia.org/wiki/Recovery_time_objective
- <http://whatis.techtarget.com/definition/recovery-point-objective-RPO>
- <https://www.linkedin.com/pulse/difference-between-rto-rpo-mohammed-pmp-itol-expert-prince2>

Infraestructura IT

Son el **conjunto de plataformas y servicios** que necesitan las empresas para desarrollar su negocio (gestionar el servicio a los clientes, gestionar el trabajo con distribuidores y proveedores y gestionar los sistemas internos de la empresa).

Las **plataformas y servicios** son:

- **Plataforma de hardware** (no solo computadoras, todo componente físico como routers, cables, cámaras, aires acondicionados, etc)

- **Plataforma y servicios de software** (sistemas operativos y software empresarial que proporcionen el entorno para la mejor gestión de los recursos y comunicación con clientes)
- **Plataforma y servicios de telecomunicaciones** (que proporcionen conectividad con cliente, proveedores y interno)
- **Plataforma y servicios de gestión de datos** (que proporcionen herramientas para el almacenamiento y el análisis de datos)
- **Servicios de consultoría**
- **Servicios de instalación y mantenimiento** (de componentes físico como lógicos)
- **Servicios de integración** (que proporcionen la coordinación necesaria para que los proyectos de infraestructura se lleven a cabo con la menor fricción posible)
- **Servicios de capacitación** (que proporcionen la enseñanza tanto para los empleados como para los gerentes en el uso de los sistemas para un uso eficiente)
- **Servicios de planeación e investigación** (que proporcionen ideas de mejora a la infraestructura actual)

Fuentes:

- <http://ti-1.wikispaces.com/2.0+Infraestructura+TI>

Datacenter

Concepto

Conjunto de recursos físicos, lógicos y humanos que son necesarios para *organizar* -ordenar para lograr un fin-, *realizar* -llevar a cabo- y *controlar* -advertir desvíos- las actividades informáticas de una empresa.

Esto nos quiere decir que el datacenter no son solamente las computadoras que vemos en las fotos, sino eso más todos los demás recursos necesarios para que funcione de forma perfecta.

- **Apoyar la labor administrativa** para hacerla más segura, fluida y simple
- **Centralizar y procesar datos** de forma automática
- **Satisfacer la necesidad de información** para la toma de decisiones dentro de una organización.
- **Presta servicios** (producción, control de operaciones, captura de datos) a áreas dentro de una organización.
- **Operación**
 - Ejecutar los procesos según los calendarios preestablecidos

- Revisar los resultados de los procesos
- Realizar acciones correctivas a los procesos
- Mantenimiento
 - Realizar back-ups
 - Llevar un registro de fallas, problemas, soluciones, acciones realizadas, etc.
 - Vigilar que los sistemas se mantengan funcionando siempre
 - Accionar si los sistemas dejan de funcionar
- Seguridad
 - Aplicar normas de seguridad y control

Un datacenter se divide en áreas (operaciones, desarrollo, implementación, soporte técnico).

Un datacenter es la culminación de la sistematización de la empresa.

Un datacenter predica la buena administración.

Decisiones para implementación

¿Qué cosas hay que tener en cuenta antes de implementar un datacenter, si es que alguna vez lo haces?

Planeación estratégica

Elegir entre un datacenter local y propio vs uno en la nube.

Tener en cuenta las ventajas y desventaja que aporta la nube para este punto.

Planeación de recursos

¿Qué beneficios tiene? ¿Qué costo tiene?

Listar los beneficios -tangibles/intangibles/fijos/variables- y los costos -operativos/instalaciones/personal/otros- que aporta el datacenter.

No hay ejemplos pero la idea es que el “precio” de tus beneficios sea mayor al “precio” de tus costos.

Planeación operativa

¿Cómo va a funcionar? ¿Qué servicios va a prestar?

Determinar los procesos de uso para apoyar la labor administrativa de la empresa y los planes de contingencia (prevención, detección, recupero, de datos, de hardware).

Tener en cuenta todo lo que hay en el apartado de *Planes de Contingencia* para este punto.

Planeación de instalaciones

EREA: Espacio, Refrigeración, Energía y Ancho de banda. Se debe contar con lo suficiente de cada punto.

Planeación de personal

¿Quiénes? ¿Cuántos? ¿Para hacer qué?

Estándar TIA

Este estándar -guías, recomendaciones, directrices- lo que te quiere dar es una clasificación de datacenters según su disponibilidad. Básicamente, los clasifica en cuatro categorías: TIER I, II, III y IV. La mejor es la IV que tiene un porcentaje de disponibilidad de 99,995% (52,56 minutos) y la peor es la I con 99,67% (28,82 horas).

En cada uno de estos niveles define ciertas características que tiene que tener tu datacenter para entrar dentro de esa categoría. Es decir, si vos cumplís esas características, “te aseguras” que tu datacenter va a tener cierta disponibilidad.

Impacto de la caída del datacenter

Lo que hay que saber es que el impacto es costoso (según la ppt es US\$ 7.900 por minuto en promedio, con un promedio de tiempo de 86 minutos. Hace la cuenta). No conviene nada que se caiga el datacenter.

Hay cuatro variables que afectan al impacto -esos US\$ 7.900- y son:

- **Costo por la inactividad de los empleados:** calculamos cuánto salen por minuto y lo multiplicamos por los minutos que estuvo caído el datacenter.
- **Costo por la pérdida de operaciones:** calculamos la suma de los beneficios que nos hubieran generado las transacciones que no se llevaron a cabo en el tiempo que estuvo caído el datacenter.
- **Costo por incumplimiento en contratos:** calculamos cuánto es la penalización que tenemos que pagar por interrumpirles el sistema a otras empresas.
- **Costo por la pérdida de confianza en la marca:** totalmente subjetivo y estimativo, calculas en términos monetarios cuánto te salió que tu producto no diera servicio.

Te voy a decir cuatro causas que generan caídas en los datacenters:

- Fallo en las baterías/sistema/equipo/capacidad UPS
- Fallo en equipo IT
- Ataque cibernético
- Fugas de agua
- Errores humanos o accidentes (te doy una más)

Mainframe

Concepto

Computadora pensada para el procesamiento de grandes cantidades de datos.

Características

- Alta disponibilidad (soportan grandes cantidades de usuarios concurrentes y gracias a su ingeniería interna y redundancia de componentes, nunca se podría caer)
- Alta confiabilidad (la información que brinda es acertada y no contiene errores)
- Alta capacidad de procesamiento de transacciones
- Costosa (costos del mantenimiento del lugar, costos de servicio técnico, costos de consumo de electricidad, costos de implementación)
- Incompatibilidad entre plataformas (están pensadas con protocolos privados, por lo que la compatibilidad es nula)
- Gran almacenamiento interno (pensada para guardar y manejar grandes cantidades de datos)
- Permite virtualización (que implica reducción de costos de mantenimiento de equipos personales y mayor escalabilidad ya que hay una independencia física)
- Permite ser reparado sin afectar el servicio (por su ingeniería, es poco probable encontrar un componente que origine la caída de todo el sistema)

Supercomputadora

Concepto

Computadora pensada para el procesamiento de cálculos complejos.

Características

- Alta disponibilidad (gracias a su ingeniería interna, nunca se podría caer)
- Alta performance (debido a su característica de paralelismo de tareas y gran cantidad de procesadores, puede resolver cálculos complejos rápidamente)

- Costosa (costos de mantenimiento del lugar, costos de servicio técnico, costos de consumo de electricidad, consumo de agua para enfriamiento, costos de implementación)
- Dificultad de uso (requieren de algoritmos muy bien pensados)

Comparación entre mainframe y supercomputadora

	Mainframe	Supercomputadora
Cantidad de procesadores	Varias docenas	Miles
Dificultad de uso	Sencillo porque se programa secuencialmente	Complicado porque hay que explotar el paralelismo
Objetivo	Procesamiento de grandes cantidad de transacciones	Procesamiento de cálculos complicados y de gran cantidad de datos
Área	Empresas -bancarias, generalmente- y organismos administrativos -por ejemplo la afip-	Ciencia, desarrollos industriales y ejército
Unidad de medición	MIPS (micro instrucciones por segundo)	FLOPS (operaciones de coma flotante por segundo)

Las supercomputadoras y los mainframes son dos “tecnologías” que llevan a la computación a sus límites. Ambas son máquinas poderosísimas y grandes, y generalmente están metidas en una habitación y son de color negro.

La gran diferencia es en su propósito. Cada una fue diseñada para lograr un objetivo particular y ser la mejor en hacerlo.

Las supercomputadoras fueron diseñadas para resolver problemas donde la limitación es la velocidad de cálculo, es decir, para solucionar problemas complicados. Mientras tanto, los mainframes fueron diseñados para resolver problemas donde la limitación es la gran cantidad de transacciones, es decir, poder manejar muchas transacciones sin perder ninguna.

Las supercomputadoras se usan para correr grandes simulaciones o analizar grandes cantidades de datos en busca de nuevos modelos o tendencias. Por eso se usan mucho en la ciencia.

Los mainframes se usan como soporte para la cantidad de transacciones que puede producir una empresa, por ejemplo, visa con tarjetas de créditos. Por eso se usan mucho en las empresas.

<http://aspg.com/mainframes-vs-supercomputers/#.V4WPzGh96UI>

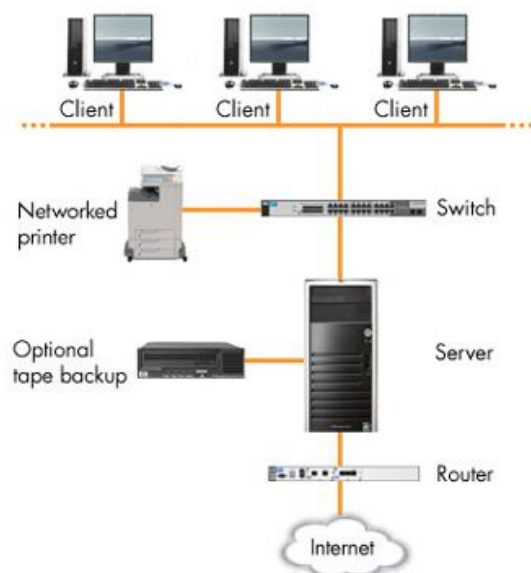
Servidor hardware

Concepto

Computadora pensada para ser más veloz que las computadoras medias y para suministrar información a los dispositivos que estén conectados a él.

Ventajas

- Autenticación centralizada (poder manejar un directorio de usuarios de todos los que se conectan y asociarles permisos)
- Información centralizada (sin el servidor, la información debería estar distribuida en varias computadoras, teniendo problemas de versionado)
- Información compartida (todos los usuarios tienen acceso a un directorio en donde intercambiar información)
- Aplicaciones centralizadas (aplicaciones centrales en la empresa pueden ejecutarse desde el servidor)
- Tener un lugar para acceder de forma remota (los empleados que trabajan de la casa se pueden conectar a este servidor para obtener la información que necesitan)
- Mayor seguridad ante contingencias/desastres (los servidores vienen integrados con sistemas de backup para proteger la información ante la mayoría de los problemas)
- Poder implementar virtualización



Comparación de tipos (torre/rack/blade)

Torre	Rack	Blade
-------	------	-------

Forma	Similar a una computadora	Similar a un organizador de CDs	Similar a un organizador de CDs
Ventaja	Disponen de todo lo necesario para operar. Versiones baratas No necesitan de un lugar especial	Reducen espacio Buena escalabilidad Componentes centralizados (servidor, almacenamiento, cable de red, switches)	Reducen espacio y consumo (todo integrado en el contenedor) Eliminan complejidad de cableado (build-in) Pueden cambiarse en caliente Mejor performance
Desventaja	Complejidad en el cableado. Requieren mucho espacio. (esto pasa cuando tenes muchos servidores torres)	Complejidad en el cableado (pero más limpia que torre) Necesitan de un lugar especial	Disponen de cpu, memoria y buses. (Los gabinetes disponen de fuentes de alimentación, ventiladores, almacenamiento) Problemas para escalar Más caros Necesitan de un lugar especial
Recomendacion	Pequeña/Mediana empresa sin experiencia	Pequeña/Mediana empresa con experiencia y necesidades de un centro de datos centralizado	Medianas/Grandes empresas con experiencia y necesidades de un centro de datos centralizado y especializado

Fuentes:

- <http://www.dell.com/learn/uk/en/ukbsdt1/sb360/best-server-uk>
- <http://www.techradar.com/news/world-of-tech/roundup/blade-rack-or-tower-which-server-is-best-for-you-1133161>
- <http://www.techrepublic.com/blog/the-enterprise-cloud/the-pros-and-cons-of-tower-rack-and-blade-servers/>
- http://es.wikipedia.org/wiki/Servidor_blade

- <http://www1.la.dell.com/content/topics/segtopic.aspx/es/dell-server-basics-buy-guide?c=cl&l=es&cs=clbsdt1>

Clúster

Concepto como algo lógico

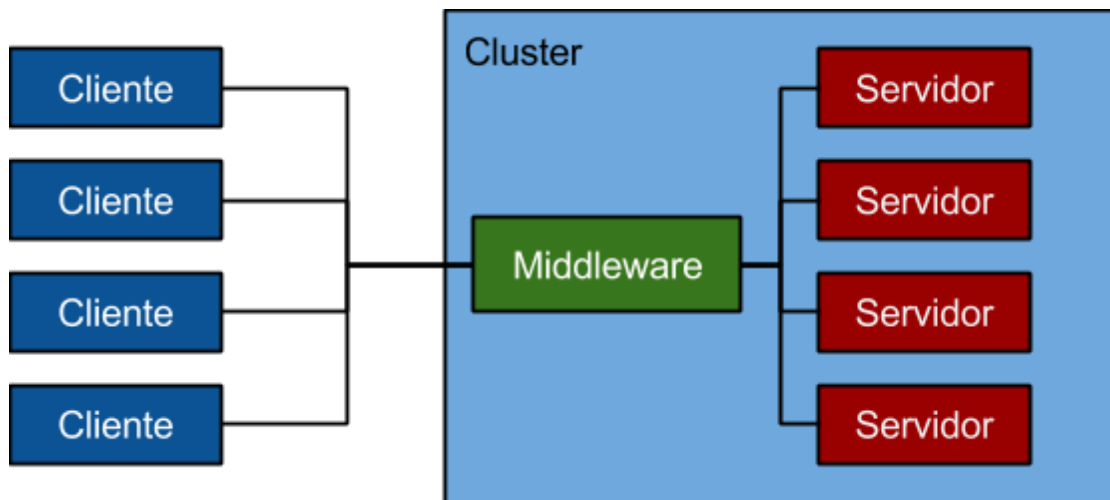
Sistema de computación distribuido centralizado geográficamente que presta recursos (CPU y almacenamiento) para resolver problemas de gran escala.

Concepto como algo físico

Grupo de computadoras (no necesariamente iguales) centralizadas en una habitación y conectadas por una red de alta velocidad que se comportan como si fueran una sola computadora y tienen por objetivo prestar recursos (CPU y almacenamiento).

Concepto según ADR

Grupo de computadoras conectadas por una red de Internet que se comportan como si fueran una sola y tienen por objetivo prestar recursos (CPU y almacenamiento). No necesariamente tienen que estar ubicadas en un mismo lugar geográfico. No necesariamente tienen que estar conectadas por una red de alta velocidad. Ejemplo: todos prestamos recursos de nuestras computadoras para buscar el número primo más grande. Por lo tanto, todas las computadoras que prestaron recursos conforman el clúster.



Ventajas

- Alta disponibilidad (solo en clústers HA-C) (por la ingeniería del cluster de procesamiento distribuido, si se cae un nodo, el clúster es capaz de levantar la tarea en otro nodo)
- Alta escalabilidad (puedo agregar servidores fácilmente)
- Alta eficiencia en recursos (ya que aprovecho la virtualización)
- Alto rendimiento (solo en clústers HP-C) (por la ingeniería del clúster de procesamiento distribuido)
- Menos costoso que un mainframe (y con capacidad de procesamiento de grandes cantidades de datos importante, solo HA-C)
- Menos costoso que una supercomputadora (y con capacidad de cálculo importante, solo HP-C)

Desventaja

- No puede trabajar con software heterogéneo

De alta disponibilidad(HA-C)

La característica principal es que son capaces de arrancar los servicios de un nodo que se cayó/desconectó en otro nodo, de modo tal que el sistema nunca deje de funcionar. Cuando el nodo vuelve a estar operativo, se le migra el servicio de nuevo, sin intervención ni conocimiento del usuario. Es decir, alta disponibilidad.

Otra posibilidad es que los servicios dejen de funcionar. En ese caso, el software del clúster también puede actuar para levantar el servicio en otro nodo.

De alto rendimiento (HP-C)

La característica principal es que son capaces de realizar cálculos complejos en poco tiempo. Para eso necesitan que las tareas sean capaces de ser paralelizable.

De balanceo de cargas (LB-C)

La característica principal es que son capaces de rutear la carga de trabajo a otros nodos con eficiencia. Es decir, como conocen cual es la carga de trabajo de cada nodo y sus características, tienen la información necesaria para hacer el mejor ruteo.

Un poco también garantizan o se encargan de la alta disponibilidad, y además pueden levantar nodos si la carga es alta.

Middleware de un clúster

Software con políticas para las siguientes funciones (puede correr en un servidor aparte)

- Único punto de acceso (para generar la sensación que se trabaja con un único ordenador muy potente)
- Migración de procesos de servidor (congelar, matar, mudar procesos)
- Balanceo de cargas (según capacidad de procesamiento, carga de trabajo actual, mantenimiento de servidores)
- Tolerancia a fallos (poder levantar el proceso en otro nodo, si se cae en uno)
- Seguridad (autenticación de usuarios)
- Redundancia (si escribo, lo mando a un servidor de replicación. Si leo, lo mando a un servidor distribuidor de cargas)

Fuente:

- http://es.wikipedia.org/wiki/Cl%C3%BAster_%28inform%C3%A1tica%29

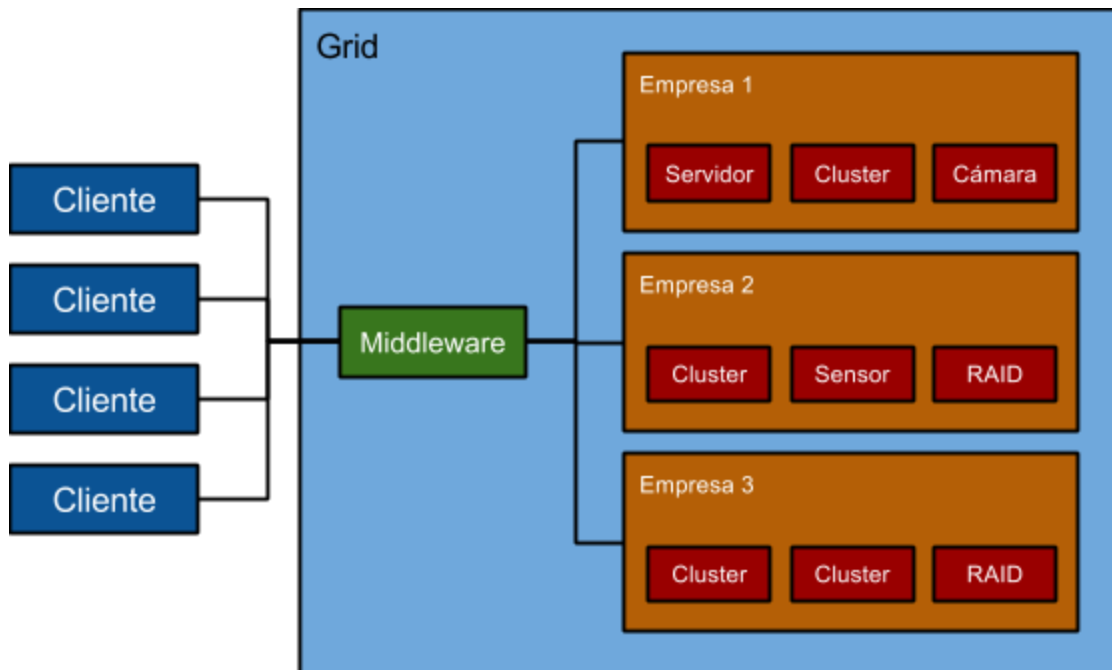
Grid computing

Concepto como algo lógico

Sistema de computación distribuido no centralizado geográficamente que presta recursos (CPU, almacenamiento, herramientas, software) para resolver problemas de gran escala.

Concepto como algo físico

Grupo de computadoras distribuidas por el mundo (en varias empresas en realidad) y conectadas por redes (de alta velocidad, de intranet, de Internet) que se comportan como si fueran una sola computadora y tienen por objetivo prestar recursos (software, cpu, almacenamiento, sensores, hardware periférico).



Ventajas

- Alto rendimiento (por su característica de procesamiento distribuido)
- Alta disponibilidad (por su característica de procesamiento distribuido y respuesta ante fallos)
- Alta escalabilidad (puedo detectar hardware y agregarlo al sistema)
- Puede trabajar con recursos heterogéneos (computadoras diferentes, redes diferentes)

Desventajas

- Comunicación lenta en muchos casos (cuando se usan equipos con conexiones lentas)

Conflictos

- Trabajar con recursos heterogéneos requiere de un software muy bien pensado
- Trabajar con tanto recursos requiere de un software muy bien pensado

Middleware de un grid

Software encargado de gestionar el grid y sus recursos de forma eficiente. Similar al de clúster

Fuentes:

- http://es.wikipedia.org/wiki/Computaci%C3%B3n_grid
- http://en.wikipedia.org/wiki/Grid_computing
- <http://www.ramonmillan.com/tutoriales/gridcomputing.php>

Cloud computing

Concepto como algo lógico

Sistema de computación distribuido no centralizado geográficamente que presta recursos (CPU, almacenamiento, herramientas, software) para resolver problemas de gran escala.

Concepto como algo físico

Grupo de computadoras distribuidas por el mundo y conectadas por redes (de alta velocidad, de intranet, de Internet) que se comportan como si fueran una sola computadora y tienen por objetivo prestar recursos (software, cpu, almacenamiento, sensores, hardware periférico) a clientes de manera on-demand.

Concepto

Tecnología que permite ofrecer y consumir servicios (de cálculo, de almacenamiento, de ejecución de aplicaciones, de acceso a recursos, de herramientas) a través de internet, y en la medida que uno necesite.

Ventajas

- Acceso a los datos desde cualquier computadora con acceso a Internet (es decir, desde computadoras heterogéneas)
- Alta disponibilidad (por su característica de procesamiento distribuido y respuesta ante fallos)
- Alta flexibilidad (puedo aumentar y disminuir la infraestructura on demand)
- Alto rendimiento (por su característica de procesamiento distribuido)

- Consumir antes que construir (los recursos pasan a ser consumidos como la electricidad, en vez de tener generadores propios)
- Menores costos (no hay que invertir en hardware, ni instalación, ni actualización, solamente pagar por lo que se usa)
- Olvidarse de la seguridad (se la delega en el proveedor)

Desventajas

- Información sensible puede verse vulnerada (por estar en servidores ajenos)
- Dependencia de los proveedores de servicios (¿qué pasaría un proveedor quebrara y me deja sin servicio?)
- Seguridad (se depende de la seguridad que provea el proveedor)

Infrastructure as a Service (IaaS)

Ofrecen máquinas virtuales para procesamiento y para almacenamiento y APIs para usarlos (todo corre en los servidores del que presta el servicio). Es decir, se ofrece un disco duro de almacenamiento teóricamente infinito y un cpu de procesamiento teóricamente infinito o todo lo que el dinero pueda comprar. Ejemplos: Amazon Web Services, Windows Azure, VMware vCloud Hybrid Service.

El consumidor tiene control sobre como se configuran los sistemas operativos, los servidores, las conexiones, etc. Pero no sobre las máquinas virtuales, la infraestructura donde corren, etc.

Platform as a Service (PaaS)

Ofrecen herramientas y ambiente para el desarrollo de aplicaciones. Los desarrolladores no tienen que conseguir las licencias ni mantener las herramientas en óptimas condiciones. Ejemplos: Google Apps Engine o Microsoft Azure, Heroku.

El consumidor no tiene control sobre como se configuran los sistemas operativos, los servidores, las conexiones, etc. Aunque si lo define él. Es decir, dice lo que hay que hacer y después el proveedor lo hace.

Software as a Service (SaaS)

Ofrecen software. Se elimina la necesidad de instalar el software en una computadora local y del mantenimiento del hardware y del software para su ejecución. La información y las aplicaciones se guardan en la nube. Ejemplos: Microsoft Office ofrece software de productividad y email, servicios de calendarios, de streaming en video, etc. Otros ejemplos son Google Drive, Dropbox, Gmail o Box.

El consumidor no tiene control ni gestión de nada. Solamente usa la aplicación como se la presenta el proveedor.

De Datacenters a PaaS

Cuando quieras levantar un servidor para tu página web, lo que tenías que hacer es tener un servidor, con switches, routers, dispositivos de almacenamiento, etc y conectarlo a red y levantar tu aplicación. Si esta era muy grande, los recursos físico crecían en cantidad y en costos. Estos podía llamarse Datacenters.

Esta forma tenía ventajas como todas las debilidades de la virtualización -ver Virtualización-. Pero a su vez tenía problemas de costos y mantenimiento.

La virtualización de los servidores o datacenters fue el siguiente paso en el hosting de páginas web. A cambio de resignar algo de rendimiento y agregar complejidad, obtenías todos los beneficios de la virtualización.

Las empresas que poseían grandes datacenters y proveían soluciones de hosting a otras empresas, empezaron a darse cuenta de que también le podían ofrecer los mismos servicios a los usuarios finales o empresas más chicas. Ahí fue cuando nacieron los primeros IaaS. Uno de los grandes impulsores fue Amazon con Amazon Web Services.

Lo que hacías era pedirle a estas empresas que te dieran una o varias máquinas virtuales para vos con ciertas características y vos te encargabas de administrarlas. Además, los proveedores te ofrecían soluciones de disponibilidad y seguridad y un poco todo lo que puedes hacer con la virtualización.

Esto fue un gran cambio para el hosting particular, ya que permitía tener una idea y llevarla a producción mucho más rápido que si tenías que montar tu propio servidor físico.

Sin embargo, ¿qué pasaba? Administrar una máquina virtual no es tarea sencilla y levantar servidores, DNS, balanceadores de carga, monitoreo de aplicaciones tampoco. Por eso surgieron los PaaS.

Lo que hacen básicamente es solucionar esos problemas. Te dicen: “vos dame tu aplicación, decime como correrla que yo la hago correr, te la mantengo, te levanto las instancias que quieras, te levanto las instancias si se caen, etc”. Un proveedor muy conocido es Heroku que utiliza máquinas virtuales de Amazon para correrte la aplicación y gestionarla.

<https://blog.containership.io/iaas-vs-paas-vs-caas-which-cloud-architecture-is-right-for-you-part-1-c7bf3c48c70c>

<https://blog.containership.io/iaas-vs-paas-vs-caas-which-cloud-architecture-is-right-for-you-part-2-a72623d7d001>

Fuentes:

- http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube
- <https://debitoor.es/glosario/definicion-cloud-computing>
- <http://searchcloudcomputing.techtarget.com/definition/cloud-computing>
- <http://www.ticbeat.com/cloud/que-es-cloud-computing-definicion-concepto-para-neofitos/>

Comparación entre clúster, grid computing y cloud computing

	Clúster	Grid Computing	Cloud Computing
Ubicación	Todas las computadoras en una sola habitación (esto es nada más con la definición más formal. Se puede dar el caso que se refiera a un cluster cuando hay muchas pc del mundo procesando para un objetivo)	Distribuidas por el mundo	Distribuidas por el mundo
Dueño/Cliente	Es la misma persona	Varias empresas combinan sus computadoras (y recursos) y la utilizan entre todas	Proveedor ofrece el recurso y un cliente lo alquila

Conclusión

Desde que apareció la arquitectura cliente-servidor se la ha tratado de mejorar con distintas configuraciones de hardware. Primero, se utilizaba una sola computadora como servidor. Los pedidos desde los clientes no eran muchos (no había muchas computadoras conectadas a la red), entonces los podían manejar con facilidad. Con el tiempo, la cantidad de clientes empezó a crecer y los servidores a saturarse. Por ese motivo, se buscaron mejores diseños de servidores y así aparecieron los servidores de tipo rack y blade. No solo mejoraban los tiempos (la performance), sino que además ahorran en costos de mantenimiento. Sin embargo, las redes seguían creciendo (con

el avance de la red de Internet) y los procesamiento eran cada vez más costosos en términos de tiempos. Por ese motivo aparecieron los primeros clusters. Los clusters eran un conjunto de servidores rack o blade conectados por redes de altas velocidades. Pero pasaba algo, no todas las empresas tenían el suficiente dinero ni la suficiente experiencia para tener un cluster, por lo que, empezó a desarrollarse el grid computing. En resumen, el grid computing es una red de recursos que se comparten entre empresas. Por ejemplo, una empresa puede tener un cluster para procesamiento y otra un datacenter que almacene y procese datos. Para finalizar, estas empresas se dieron cuenta que no aprovechaban del todo sus recursos y que podían alquilarlos a otras empresas. En ese momento apareció el cloud computing. El cloud computing es básicamente el alquiler de recursos.

Comparación entre opex y capex

Opex

Operating Expense. Se mide en dinero.

La **suma total de dinero** que se usó para el mantenimiento de los bienes de capital -tanto equipos como limpieza de la oficina-, gastos consumibles -cosas de librería, comidas para empleados- y gastos para el funcionamiento del negocio -publicidad, servicios públicos y privados, honorarios, licencias-.

Los costos impositivos relacionados con estos gastos son deducidos completamente en el período contable -generalmente un mes- que se hacen.

Es decir, podemos ver una relación con el estado de resultado:

+ Ventas
- Costo por ventas
Resultado bruto
- Amortizaciones
+/- Otros gastos
Resultado parcial
- Impuesto a las ganancias o ganancia mínima presunta
+ Amortizaciones

El Opex sería todo lo que le restas al resultado bruto y como se ve, hace más chico al resultado neto, por lo que, hace más chico lo que pagas por ganancias.

Capex

Capital Expenditures. Se mide en dinero.

La **suma total de dinero** que se usó para inversiones de bienes de capital. Es decir, la sumatoria de dinero en inversión o en mejoras de los bienes de capital actuales para mantener funcional el negocio.

Los costos impositivos relacionados con estos gastos de inversión son deducidos -amortizados- a lo largo de los años.

Lo que gastas en comprar una nueva máquina, no lo ves en el estado de resultado -a menos que lo finances y lo imputes como gastos de financiación-. Lo único que ves que afecta al resultado bruto y por consiguiente al impuesto a las ganancias es la amortización.

Conclusión

- Todos los gastos que se realizan cuando empieza el negocio para adquirir bienes de capital son Capex.
- Todos los gastos que se hagan día a día son OPEX.
- Todos los gastos excepcionales que se hagan para mejorar el rendimiento, reducir el OPEX o alargar vida de bienes de capital son Capex.

El impuesto a las ganancias -junto con otros- se hacen sobre un total de beneficio ganado en un año (el tiempo puede ser distinto). El beneficio ganado es la diferencia entre todo lo que ganaste menos todo lo que gastaste. Todo lo que ganaste es la sumatoria de todo el dinero que ganaste por las facturas que emitiste. Todo lo que gastaste son todos los gastos que hiciste que son deducibles. Los gastos deducibles son aquellos que son necesarios para el funcionamiento de tu negocio -o de tu vida-. Todos estos gastos están explícitos por ley. Por lo tanto, cuantos más gastos podamos deducir -restar de las ganancias- por año -o período que estemos teniendo en cuenta-, menor va a ser el impuesto a las ganancias que vamos a tener que pagar.

¿Qué nos conviene más: realizar un Capex (inversión en bien de capital) en una impresora de \$1000 que podemos deducir \$200 por año en 5 años o alquilar una impresora que nos sale \$300 por año de donde podemos deducir todo porque es un OPEX? Aquí es donde entra a jugar la contabilidad y ver cuál es la mejor estrategia a adoptar.

Fuentes:

- <http://es.wikipedia.org/wiki/Opex>
- <http://es.wikipedia.org/wiki/Capex>
- <http://jotaerre.net/2013/10/17/concepto-de-capex-opex/>
- <https://ar.answers.yahoo.com/question/index?qid=20080213204751AA7RvCw>
- <http://www.germanbacca.com/capex-vs-opex-y-cloud/>

Software base

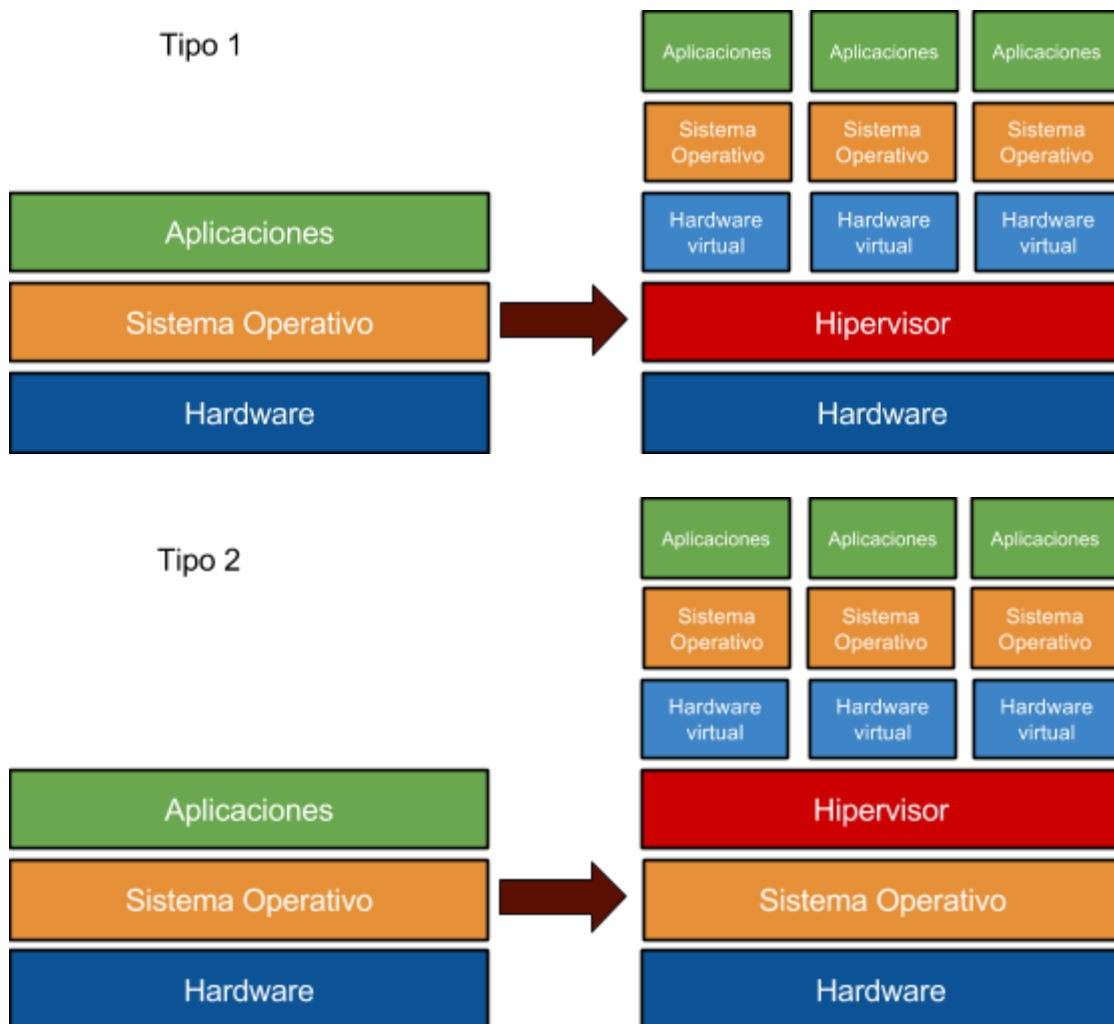
Características al momento de elección

- Aplicaciones que tenés disponibles (no todas las aplicaciones están disponibles para todos los software base. Por ejemplo, Linux no se banca el Age of Empires 2)
- Seguridad (que servicios te trae por defecto, que servicios puedes instalar, cuanta seguridad te dan esos servicios)
- Escalabilidad (qué tan fácil es agregar más memoria o recursos)
- Usabilidad (qué tan fácil es entenderlo y usarlo)
- Estabilidad (el sistema se cae seguido, tiene problemas de deadlock)

Virtualización

Proceso por el cual se crea una versión virtual de una física.

Virtualizar es crear una capa de abstracción de software del hardware, haciendo que en vez de usarse el hardware, se utilice la capa virtual.



Las **ventajas** que aporta son:

- **Mejor utilización de los recursos disponibles** (en vez de ejecutar cinco servicios en cinco servidores diferentes, ejecuto esos cinco servicios en cinco máquinas virtuales diferentes. Esto se puede lograr, ya que un servicio solo se estima que consume nada más que entre el 5-15% de los recursos del servidor. Hay veces que necesitas más que eso y ahí vas a tener que ajustar mejor la disposición de máquinas virtuales)
- **Mejor gestión de recursos disponibles y flexibilidad** (el hipervisor trae incorporado una configuración centralizada de recursos para las máquinas virtuales. Eso quiere decir que puedo distribuir los recursos de hardware disponibles según me convenga más. Si un servicio necesita mucha memoria RAM, le puedo asignar la que necesite y quitársela a otro que tal vez no la necesite tanto. De hecho, puedo hacerlo en runtime)
- **Mayor confiabilidad, disponibilidad y tolerancia a fallos** (por ejemplo, si una VM se cae por cualquier razón, automáticamente el hipervisor levanta otra con el mismo estado que la que se cayó. O por ejemplo, no es necesario cortar el servicio por el reemplazo de alguna pieza de

hardware, ya que solamente se muda la VM a otro hardware. Todo esto hace que el tiempo que el servicio este caído tienda a cero)

- **Mayor seguridad** (ya que hay una administración global centralizada donde se controla todo y se asigna recursos desde un solo lugar. Además, las máquinas virtuales están aisladas unas de las otras, por lo que si una se infecta con un virus, las demás no se enteran)
- **Facilidad para crear, mover, copiar, clonar y eliminar entornos** (se pueden clonar las VM, lo que permite que si quiero levantar un nuevo servidor no tengo que comprar otro servidor (como era antes) sino que levanto una nueva VM en el servidor actual. Se pueden mover las VM de servidor en servidor. Se pueden crear snapshots. Se pueden eliminar y volver a crear)
- **Poder ejecutar varios sistemas operativos en un solo servidor físico.**
- **Reducción del costo, espacio necesario y del consumo** (ya que va a haber menos computadoras. El espacio tiene un costo de mantenimiento, lo mismo que el consumo. Además los hipervisores permiten administrar mejor las aplicaciones de manera que se consuman los menores recursos posibles)

Las **desventajas** que aporta son:

- **Overhead y pérdida de rendimiento** (ya que las aplicaciones no tienen el 100% de la cpu para ellas y tienen que competir para obtener los recursos)
- **Requiere tiempo, experiencia y agrega un capa más donde las cosas pueden fallar**

Virtualización de servidores

Las organizaciones, pre virtualización, podía correr un solo sistema operativo y una sola aplicación en cada servidor. Esto generaba que solamente se usará entre un 5-15% de los recursos disponibles.

Gracias a la virtualización de servidores, se pudo crear hardware virtual que permitió crear servidores virtuales. En cada uno de estos servidores virtuales o máquinas virtuales, se podía instalar cualquier sistema operativo.

Unos de los puntos más importantes es que cada máquina virtual estaba aislada una de la otra y, gracias a la inclusión de ese software, se pueden asignar recursos dinámicamente.

El software encargado de crear la versión virtual en lugar de la física se conoce como Hipervisor.

Existen dos tipos de hipervisores: los de tipo uno y los de tipo dos. Los primeros, se instalan directamente sobre el hardware -como si fueran un sistema operativo-, mientras que los segundos se instalan sobre el sistema operativo que tiene instalado el hardware.

Ejemplos de los de primer tipo son Hyper-V de Microsoft y VMware vSphere de VMware.

Ejemplos de los de segundo tipo son Oracle VirtualBox de Oracle y VM Fusion o VM Workstation de VMware.

Por tener acceso directo al hardware, los de tipo uno son más eficientes y pueden realizar funciones no implementables en los de tipo dos. En cambio, al correr sobre un sistema operativo, los de tipo dos son más flexibles en cuanto a su instalación.

Virtualización de acceso

Abstracción de las interfaces de dispositivos y aplicaciones para que puedan comunicarse fluidamente sin que tengan que conocerse.

Tener un servidor con todas las aplicaciones instaladas y que los usuarios se conecten por una aplicación cliente a ese servidor para usar las aplicaciones.

<http://searchvirtualdesktop.techtarget.com/definition/Citrix-XenApp>

Virtualización de aplicaciones

Encapsulan la aplicación en un contenedor. Ese contenedor puede tener diferentes cosas según la tecnología o aplicación que se use para encapsular.

Una vez encapsulada, el contenedor se puede mover a otro sistema operativo y ejecutarse ahí sin tener que instalar nada y con las mismas configuraciones que tenía al momento de encapsulamiento. Además, al ser un contenedor, no entra en conflictos con aplicaciones existentes.

Ejemplos de virtualizador de aplicaciones son App-V de Microsoft, ThinApp de VMware y AppZero de AppZer

<http://stealthpuppy.com/app-v-faq-2-what-is-microsoft-application-virtualization/>

<http://stealthpuppy.com/app-v-faq-19-does-app-v-replace-citrix-xenapp-or-remote-desktop-services/>

<http://www.vmware.com/ar/products/thinapp/features.html>

<https://www.appzero.com/faq#n213>

Virtualización de procesamiento

Abstracción de los componentes de hardware encargados del procesamiento. Puede ser tomada como similar a virtualización de servidores de tipo uno.

Virtualización de almacenamiento

Abstracción de los componentes de hardware encargados del almacenamiento.

Agrupación de espacio físico de múltiples dispositivos de almacenamiento en red en lo que aparenta ser un único dispositivo, mucho más grande.

Se aumenta la eficiencia y simplicidad para el guardado de datos. Es decir, se delega en esta nueva capa de abstracción la lógica de como dividir un archivo en varios dispositivos de almacenamiento. Por eso se dice que genera simplicidad, porque las aplicaciones ya no tienen que tener esa lógica.

Independencia física. Puedo almacenar un gran archivo en múltiples sistemas con múltiples formas de guardado (puedo guardar en Windows, Linux y un Mainframe)

Creación de particiones virtuales, donde cada una pueda tener su propio file system, sus propias políticas, sus propios permisos.

Por ejemplo, RAID es un ejemplo de varios discos que funcionan como uno.

Virtualización de red

Abstracción de los componentes de hardware encargados de las conexiones de redes (switches, routers, canales, banda ancha)

Por ejemplo, la división lógica de un recurso de red como puede ser la banda ancha. Se divide en varios canales que pueden ser asignados o reasignados a distintos servidores en tiempo real. Cada uno de estos canales pueden tener sus propias políticas, restricciones y capacidad.

Los servidores se conectan a redes lógicas con puertos, switches, load balancers, VPNs lógicas.

Garantiza todas las funcionalidades de un recurso de red más las funciones operacionales (rápido aprovisionamiento, etc.) y la independencia física que ofrece la virtualización.

Paravirtualización vs Full virtualización

La gran diferencia es que en la paravirtualización, el sistema operativo sabe que está siendo virtualizado, por lo que llama al sistema operativo host por cada acción que desee realizar sobre el hardware. En cambio, en la full virtualización, el sistema operativo no es consciente de que está siendo virtualizado, por lo que ejecuta sus acciones en hardware virtualizado por el sistema operativo host. En este caso, el hypervisor se encarga de traducir esas llamadas al sistema en runtime para poder ejecutarlas en el hardware disponible.

Fuentes:

- <http://www.zdnet.com/article/sorting-out-the-different-layers-of-virtualization/>
- http://en.wikipedia.org/wiki/Application_virtualization
- https://www.vmware.com/files/pdf/VMware_paravirtualization.pdf
- <http://www.vmware.com/latam/virtualization/how-it-works>

Base de datos Relacionales

Características al momento de la elección de base de datos relacional

- Réplicas en otras bases de datos y sincronización entre estas réplicas (¿se incluyen estas características? ¿con qué facilidad?)
- Niveles de aislamiento soportados (para permitir la concurrencia y el ACID)
- Tipos de índices soportados
- Tipos de backups soportados y herramientas para el manejo
- Soporte para vistas materializadas (son como nuevas tablas que se crean dinámicamente y se mantienen actualizadas)
- Tipos de optimizaciones de consultas
- Soporte para particionamiento de tablas grandes en tablas más chicas (para aumentar la performance de las consultas y ganar espacio)
- Tipo de seguridad que proveen
- Rendimiento (<cantidad de algo> / segundo)
- Costo (muy importante es el costo por byte almacenado)
- Soporte y capacitación brindado por el proveedor y por entidades externas
- Tamaño estimado del set de datos que se pretende guardar
- Herramientas que se ofrecen para OLAP y Data Mining
- Hardware y SO dónde funcionan
- Facilidad para desarrollar

Bases de datos NoSQL

Son las bases de datos que comparten una misma arquitectura interna particular, diferente a la arquitectura de las bases de datos relacionales.

Diferencias con las relaciones

- No están basadas en tablas, sino que se basan en clave-valor, documentos, grafos, etc
- No necesariamente necesitan un esquema para funcionar
- No necesariamente necesitan de tipos de datos “estructurados”
- No necesariamente garantizan ACID
- No necesariamente soportan SQL como lenguaje de consultas, por lo que no necesariamente tienen JOINS
- No necesariamente soportan relaciones

Características de DB NoSQL

- Es más fácil escalar horizontalmente que las relacionales: en general, para las no relacionales, la tecnología viene “por default” mientras que para las relacionales, la tarea es externa y más compleja, por lo tanto más cara.
- Mejor performance en situaciones específicas
- Mejor performance al manejar grandes cantidades de datos
- Es más fácil para migrar el esquema, ya que muchas no tienen esquemas.
- No aseguran ACID, muchas veces
- Falta de estándares
- Falta de madurez

Teorema de CAP

Es un teorema que dice que un **sistema de cómputo distribuido** solo puede garantizar dos de las tres características fundamentales:

- **Consistencia** todos los request que se reciban, van a tener el mismo resultado si se realizan al mismo tiempo en todos los nodos
- **Disponibilidad:** todas los request que sean recibidos por un nodo activo, tienen que tener un response que no sea un error
- **Tolerancia a particionado:** todos los nodos están activos, pero se perdió la comunicación con por lo menos uno de ellos.

Hay que notar que dice sistema de cómputo distribuido, por lo tanto ojo, puede afectar a las bases de datos relacionales si esta está distribuida.

CP

El sistema ejecuta todas las operaciones de forma consistente, incluso aunque se desconecte un nodo. Lo que no asegura es que devuelva una respuesta ante una petición (ya que las respuesta a consultas no las voy a poder hacer porque no puedo actualizar el otro nodo y necesito estar consistente).

CA

El sistema ejecuta todas las operaciones de forma consistente, y siempre devuelve una respuesta. Lo que no asegura a seguir estando operativo si se cae un nodo (si un nodo se desconecta, el sistema no va a funcionar más ya que no puedo actualizar los datos de forma consistente, por lo que pierdo disponibilidad). Impracticable

AP

El sistema asegura siempre devolver una respuesta, incluso si se desconecta un nodo. Lo que no asegura es que todos los nodos tengan la misma información (yo te acepto tu pedido de escritura pero no lo guardo en los dos nodos, solo en el que le cae. La próxima vez, no vas a recibir lo mismo).

Gran link para entender el problema:

- <http://www.lnds.net/blog/2012/05/dos-de-tres.html>

Explicación de porque no sirve de nada el teorema:

- <https://martin.kleppmann.com/2015/05/11/please-stop-calling-databases-cp-or-ap.html>

Según el siguiente post:

<https://www.quora.com/What-is-the-relation-between-SQL-NoSQL-the-CAP-theorem-and-ACID>

No tiene sentido pensar en una base de datos CA, ya que no tiene sentido práctico que si se cae un nodo, todo el sistema deje de funcionar. Por eso, se sugiere elegir en base a la siguiente pregunta: si se cae un nodo, ¿preferís que la base de datos te responda tal vez con información desactualizada (AP) o que no te responda porque no se pudo actualizar todo los nodos (CP)?

También puedes pensarlo así:

Una empresa quiere ofrecer su producto web a nivel mundial y quiere que en todos los lugares del mundo la performance sea la misma. Para eso decide contratar varios datacenters distribuidos por el mundo.

¿Qué pasa si un datacenter deja de funcionar? ¿Tiene que dejar de funcionar el producto en todo el mundo? Claramente no queremos eso, por lo que no queremos que nos falte la P.

Ahora ¿qué pasa? ¿Queremos que en ese país deje de funcionar el servicio? Y la verdad que no queremos, y pasa eso se adoptan medidas de replicamiento de datos. Y ahí entre el problema entre la C y la A.

Si privilegiamos la C sobre la A, decidimos que todos los datacenters, en todo momento, tengan la misma información. Por tal motivo, decidimos que la sincronización sea sincrónica. ¿Qué genera esto? Que la respuesta general de todas las consultas baje, ya que hay mucho overhead en la replicación de datos. Además, al haber muchas más transacciones, aumentan las chances de que se caigan otros datacenters.

Si privilegiamos la A sobre la C, decidimos que si se cae un datacenter, algunos datos no sean replicados síncronamente, sino asíncrona. Perdemos calidad en el producto, pero ganamos mayor velocidad de respuesta.

¿Por qué decimos que no tiene sentido no tener la P? Supongamos que tiene sentido perder la P y se cae una conexión con un nodo.

Si alguien quiere escribir en la base de datos, tendríamos que escribir en todos los nodos, incluso al que no podemos llegar. Como no podemos llegar a ese nodo, no podemos escribir. Si no podemos escribir, perdemos la igualdad de datos en todos los nodos, perdemos la consistencia.

Si no permitimos las escrituras, cada vez que se haga una escritura, no vamos a poder responder, perdemos la disponibilidad.

Fuente:

- http://es.wikipedia.org/wiki/Teorema_CAP
- http://es.wikipedia.org/wiki/Computaci%C3%B3n_distribuida
- <http://www.rodenas.org/ferdyblog/2011/02/25/el-teorema-de-cap/>
- <http://hackingdistributed.com/2013/03/23/consistency-alphabet-soup/>

Estimaciones

Medida

Indicación cuantitativa de una propiedad de un producto, sistema o proceso.

Por ejemplo: dimensiones de una caja, líneas de código de un sistema, cantidad de tareas de un proceso.

Métrica

Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo de calidad.

Por ejemplo: robustez de un sistema, usabilidad de un componente, eficacia de un proceso.

Existen métricas orientadas al tamaño pero no están muy aceptadas porque dependen de las líneas de código que a su vez dependen mucho del lenguaje de programación elegido y el diseño elegido o de la metodología elegida. Algunos ejemplos son errores/defectos por miles de líneas de código, páginas de documentación por miles de líneas de código, errores/defectos por hora-hombre, etc.

Existen métricas orientadas a la función que evalúan según la funcionalidad, pero como no se puede medir la funcionalidad de forma directa, se utilizan técnicas como los puntos de función. Éstas son independiente de la tecnología y aplicables en cualquier etapa del ciclo de vida del software.

Indicador

Métrica o conjunto de métricas que dan una visión del proceso, proyecto o producto de software.

Por ejemplo:

Ayudan a evaluar el estado (productividad, calidad, etc) de un producto, proceso o proyecto en un instante dado y a detectar áreas problemáticas. Esta ayuda me permite tomar mejores decisiones.

Puntos de función

<http://www.softwaremetrics.com/Function%20Point%20Training%20Booklet%20New.pdf>

Concepto

Método utilizado para medir el tamaño del software a partir de la funcionalidad entregada, independientemente de la tecnología elegida y el ciclo de vida actual. No es confiable para proyectos pequeños y resulta muy arduo para empresas que no tienen experiencia con el método (ya que se requiere de bastante conocimiento para poder usarlo correctamente).

Entradas externas

Cada dato único (formato diferente o con diferente lógica de procesamiento) o salida de control que ingrese al sistema y actualice un archivo lógico interno.

Cualquier modificación que quede registrada en el sistema.

Ejemplos: en una pantalla ABM tenemos 3 entradas, en dos pantallas con los mismo datos a ingresar pero diferente lógica de procesamiento tenemos 2 entradas.

Salidas externas

Cada dato único (formato diferente, con diferente lógica de procesamiento o actualiza a un archivo diferente) o salida de control procedural que sale del sistema y en que se realiza, por lo menos, un cálculo matemático.

Ejemplos: informes, gráfico, facturas, cheques, mensajes a otras aplicaciones, mensajes al usuario, etc.

No ejemplos: mensajes de confirmación.

Consultas externas

Cada combinación única (formato diferente o con diferente lógica de procesamiento) en donde cada entrada online da como resultado una salida online inmediata y no se actualizan archivos lógicos internos.

Si se procesa, no es consulta.

Ejemplos: una aplicación que pregunta por el precio de un productos, pantalla de selección de archivos almacenados (por ejemplo usuarios)

Interfaces externas

Cada grupo lógico de datos o de información de control que se almacena fuera de los límites de la aplicación.

Archivos lógicos internos

Cada grupo lógico de datos o de información de control que se almacena dentro de los límites de la aplicación.

Juicio experto

Una persona, en base a su experiencia y sus creencias, estima cuánto va a durar cierta tarea.

Ventajas:

- Rápida (no necesita mucho análisis)
- Barata (se necesita de una sola persona)

Desventajas:

- Poco precisa (por el poco análisis que se hace y porque no tiene en cuenta las diferencias entre proyectos)
- Necesita personal experimentado

Wideband

Similar a juicio experto, pero participan más personas en la estimación de las tareas.

Estimación análoga (TOP-DOWN)

Es la que está basada en proyectos anteriores similares. Es decir, si cae un proyecto para realizar una página web de administración de clientes y vos ya habías hecho alguna, estimas que más o menos va a tardar lo mismo.

Ventajas:

- Rápida (no necesita mucho análisis)

Desventajas:

- Poco precisa (por el poco análisis que se hace y porque no tiene en cuenta las diferencias entre proyectos)
- Necesita personal experimentado

Estimación detallada (BOTTOM-UP)

Es la que se analizan todos los componentes y se estima a partir de eso.

Ventajas:

- Precisa (por el análisis)

Desventajas:

- Lenta y costosa (necesita mucho análisis, ya que hay que identificar a todos los componentes)
- Necesita de un proyecto definido (ya que si cambian los componentes, cambia la estimación, por eso se necesita de un proyecto estable)

Estimación paramétrica

Definidos por una fórmula en función del esfuerzo.

Estimación por tres valores (PERT)

Es la que analiza el camino crítico y calcula los valores optimistas, pesimistas y normales y hace un promedio.

PERP sería como una mejora de juicio experto y wideband. Es decir, lo que hace además de tener en cuenta los valores que otorgan estas dos últimas técnicas es una “banda de protección”, es decir, valores que considero mínimos y máximos. A partir de eso, obtiene, generalmente, una mejor aproximación. Además, agrega la variable de probabilidad deseada.

$$T_{esperado} = \frac{T_{optimista} + 4 * T_{medio} + T_{pesimista}}{6}$$

$$error\ o\ desvio = \frac{T_{pesimista} - T_{optimista}}{6}$$

$$desvio\ estandar = \sqrt{\sum \left(\frac{T_{pesimista} - T_{optimista}}{6} \right)^2}$$

Fuentes:

- <http://www.sc.ehu.es/jiwdocoj/mmis/fpa.htm>

Arquitectura de software

Es el **diseño de más alto nivel** de las estructuras de un sistema. Es decir, son el **conjunto de decisiones** que se tomaron respecto a los **elementos** que componen un sistema, sus **propiedades**, sus **funcionalidades**, sus **interfaces**, sus **interrelaciones** y sus **relaciones** con los sistemas exteriores que sirve para lograr uno o varios objetivos.

Esta arquitectura (o diseño de alto nivel o conjunto de decisiones) debe cumplir con uno o varios objetivos (requerimientos funcionales y no funcionales) y con una o varias restricciones (limitaciones de recursos, ya sean tecnológicos como pueden ser limitaciones de un lenguaje de programación o monetarios como puede ser el presupuesto del proyecto).

La arquitectura se diseña en etapas tempranas del desarrollo y sirve como guía para la construcción de todo el sistema. No realizar una arquitectura en etapas tempranas nos puede dar como resultado problemas sin solución en etapas futuras. De la misma manera, realizar una mala arquitectura en etapas tempranas nos puede dar como resultados problemas sin solución en etapas futuras. Por estos motivos, es crítico realizar una buena arquitectura en etapas tempranas del desarrollo.

Etapas

1. **Análisis** de requerimientos funcionales y no funcionales, las restricciones y los recursos disponibles. Y, además, análisis del contexto donde va a funcionar el sistema.
2. **Diseño** de la arquitectura (toma de decisiones respecto a elementos, funcionalidad y relaciones teniendo en cuenta los objetivos y limitaciones, por eso la etapa 1)
3. **Comunicación** del diseño de la arquitectura (a todos los participantes en el desarrollo del sistema)
4. **Evaluación y corrección** (es decir, mantenimiento evolutivo y correctivo)

Ventajas

- Reducir la incertidumbre (les da a los interesados o stakeholders (clientes, desarrolladores, Product Managers) una visión de cómo debería ser el sistema. Gracias a eso, los stakeholders pueden tomar mejores decisiones ya que se pueden sustentar en la arquitectura)
- Reutilización (dos sistemas similares pueden tener la misma arquitectura, lo que facilita la realización)
- Mejor comunicación (un diseño de alto nivel permite una comunicación más efectiva con los interesados, ya que se evitan detalles técnicos que no aportan)

- Mejora el manejo de riesgo (ya que sabemos cómo debería ser nuestro sistema, conocemos o por lo menos nos acercamos a saber en dónde podría fallar o dónde podría haber problemas, como por ejemplo detectando los “SPF” (Single Point of Failure) que son fallos que ocasionan un fallo global en el sistema dejándolo inoperante)
- Reducción de costos a futuro (malas decisiones o no decisiones llevan a problemas futuros con impactos negativos en los costos. Por ejemplo, si no se tuvo en cuenta el atributo de escalabilidad, puede llegar a ser difícil agregar nuevos servidores)

Patrones (ppt vieja)

Monocapa

Las aplicaciones monocapa son -casi siempre- aplicaciones de escritorio. Un solo usuario las puede usar al mismo tiempo. No se conectan con ninguna aplicación ni servidor. Guardan su información en el file system del sistema operativo.

Las aplicaciones monocapa, en su arquitectura de software, pueden usar cualquier otro patrón. Es decir, podría llegar a usar el patrón MVC.

Cliente-Servidor

Las aplicaciones cliente-servidor se dividen en dos: un cliente que tiene toda la lógica de funcionamiento y un servidor que es la base de datos. Permite más de un usuario al mismo tiempo. Se pueden conectar las aplicaciones (porque comparten información). Guardan la información en la base de datos a la que están conectadas.

Devuelta, la aplicación cliente puede estar construida con cualquier patrón de arquitectura de software.

Multicapa

Las aplicaciones multicapa se dividen en tres: un cliente que pide la aplicación, un servidor que provee la aplicación y se conecta con una base de datos y una base de datos. Permite más de un usuario al mismo tiempo. Permite que se conecten las aplicaciones. Guardan la información en la bases de datos.

Devuelta, las aplicaciones del cliente (el browser) y el servidor pueden ser construidas con el patrón arquitectónico que se desee.

Patrones

Concepto

Es una solución genérica, probada y exitosa ante un problema determinado en un contexto determinado. Ésta solución se debe ajustar de acuerdo al problema particular que se quiera resolver.

Layered Pattern

Es un patrón de arquitectura que se enfoca en dividir todos los componentes en capas (o módulos) independientes y cohesivos. Éstas capas exponen una interfaz pública con la que permiten la comunicación por parte de otras capas.

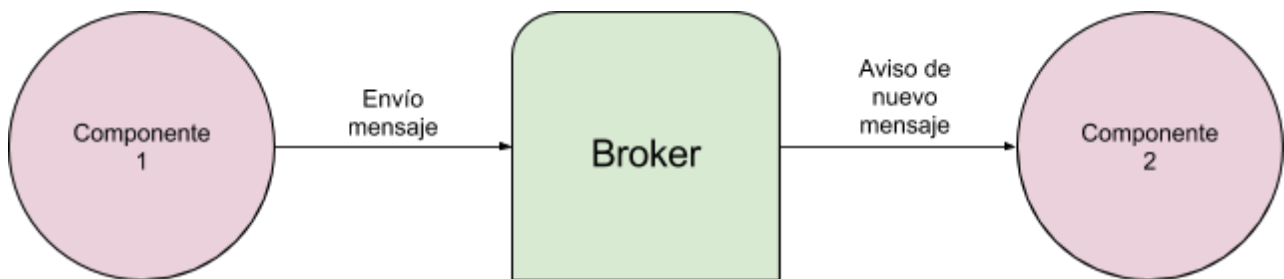
La gran ventaja es que permite a éstas capas evolucionar y adaptarse de forma independiente. Ésto permite:

- Mayor portabilidad (puedo cambiar capas y moverlas a otros proyectos)
- Mayor reutilización (varios proyectos pueden usar las mismas capas)
- Mayor mantenibilidad (al haber un menor acoplamiento entre las capas).

Message Broker Pattern

Es un patrón de arquitectura de sistemas que se enfoca en el envío de mensajes o de eventos. También se lo conoce como *Publish-Subscribe Pattern*.

El patrón propone un sistema de envío de mensajes distribuido y poco acoplado. Un componente envía un mensaje a un destino. Otro componente puede leer el mensaje desde el destino. Ninguno de los dos componentes sabe que el otro existe. De hecho, el segundo puede no estar activo al momento del envío del mensaje o puede leerlo cuando quiera. En lo único que están acoplados es en que los dos tienen que enviar/leer a/desde un destino, y en el formato del mensaje.



El broker es lo que se conoce como una cola o un tópic. Lo que hace es recibir mensajes desde un componente y avisarle a otros componentes que hay nuevos mensajes en la cola. Hay que saber que el segundo componente se tuvo que haber registrado con el broker. Al componente uno se lo conoce como publicador y al componente dos como suscriptor.

Que el envío de mensajes esté poco acoplado da un par de ventajas como:

- Mayor flexibilidad y mantenibilidad (facilidad para agregar y quitar generadores y consumidores de mensajes. Si vos quieres consumir algún evento, lo único que tenes que hacer es registrarte. Si este sistema, le tenías que decir al componente 1 que te mande el mensaje a vos, lo que es mucho más complejo y burocrático.)
- Mayor escalabilidad (ya que la cantidad de generadores y consumidores puede crecer mientras de el hardware para bancarlo)

Pero, además, podemos rescatar otras ventajas:

- Asincronicidad (el consumidor ya no tiene que bloquearse esperando que el generador le envíe el mensaje. Ojo, la asincronicidad se puede lograr sin este sistema igual)
- Seguridad (tenes un lugar para controlar los mensajes/eventos que se generan y poner filtros si es necesario)
- Datos (tenes un lugar donde podes ver todos los mensajes que se envían. Toda esa información que se genera puede ser explotada para mejorar los sistemas o descubrir errores)

<https://docs.oracle.com/javaee/7/tutorial/jms-concepts001.htm#BNCDR>

MVC Pattern

Es un patrón de arquitectura que es una especialización del *Layered Pattern*. Lo que hace es dividir al sistema en tres capas: vistas, modelos y controladores.

La primera es la que maneja la interfaz de usuario (es decir, cómo mostrar la información y cómo pedirla), la segunda es la que maneja la parte lógica del negocio (los objetos de negocio y las reglas, como validaciones y relaciones) y la tercera es la que maneja la comunicación entre ambas capas (acepta los inputs de las vistas, los convierte en objetos que el modelo entiende, los pasa a los modelos, los modelos le responden, los convierte a algo que la vista entienda).

La gran ventaja es que permite cambiar la información del negocio sin afectar la vista y viceversa (igual a medias, depende el framework y cómo programes, depende el grado de acoplamiento). Esto permite:

- Mayor portabilidad (puedo cambiar vistas o modelos o controller y moverlas a otros proyectos)
- Mayor reutilización (varios proyectos pueden usar las mismas vistas o modelos o controladores)
- Mayor mantenibilidad (al haber un menor acoplamiento entre las capas).

Client-Server Pattern

Es un patrón de arquitectura que se enfoca en dividir al sistema en dos capas. La primera son los clientes y la segunda son los servidores. Los clientes solicitan servicios a los servidores que éstos proveen.

La gran ventaja que tiene es que tengo que mantener los servicios en un solo lugar y no en muchos, lo que aumenta la mantenibilidad y reutilización.

La gran desventaja es que puede haber problemas con cuellos de botellas en los servidores.

Service-Oriented Architecture Pattern

Es un patrón orientado a servicios. Tenemos proveedores que proveen servicios y tenemos consumidores que generan servicios de consumición. Ambos se escriben en diferentes lenguajes y con diferentes implementaciones, pero los servicios de consumición deben saber cómo consumir los servicios que proveen los proveedores sin saber cómo están implementados.

La idea es muy parecida a la de *Client-Server Pattern*. Tienes los clientes que son los consumidores y tienes los proveedores que son los proveedores.

MapReduce Pattern

Es un patrón orientado a la consumición y explotación de datos almacenados en base de datos distribuidos. Lo que se hace es dividir la información en varios nodos distribuidos. Cuando se hace una consulta, se va a ejecutar mucho más rápido ya que se va a mapear y reducir por cada nodo, para después hacer una reducción general.

Fuente:

- http://es.wikipedia.org/wiki/Arquitectura_de_software
- http://en.wikipedia.org/wiki/Software_architecture
- http://sg.com.mx/revista/27/arquitectura-software#.VW8Pss9_Oko
- <https://msdn.microsoft.com/es-es/library/dd490886.aspx>

Atributos de calidad

Adaptabilidad

Es un concepto que hace referencia al costo de realizar un cambio en el sistema para adaptarse a un nuevo contexto. Cuanto menor sea el costo, mayor es la adaptabilidad.

Este costo hace referencia al costo monetario, a los costos de los riesgos que el cambio produce y a los costos de trabajo a realizar.

Disponibilidad

Es un concepto que hace referencia a la relación que existe entre el tiempo que un sistema permanece inoperable (o no disponible) y el tiempo total de medición. Cuanto menor sea esta relación, mayor será la disponibilidad.

Por ejemplo, si mi sistema va a estar operable durante 364 días seguidos y un día me lo voy a tomar para realizar mantenimiento, la relación va a ser 1/365.

Se mejora minimizando las interrupciones de servicio (con equipos redundantes), previniendo las posibles fallas del sistema (con ciertos planes de prevención), reduciendo los puntos de fallos del sistema. Es decir, tratando de bajar el numerador de la relación, tratando de evitar ser inoperable.

Escalabilidad

Es un concepto que hace referencia al costo de aumentar el tamaño de un sistema. Cuanto menor sea el costo, mayor es la escalabilidad.

Este costo hace referencia al funcionamiento del sistema, es decir, si cambio tal componente para aumentar el tamaño del sistema, cuánto tiempo de inoperabilidad tengo, cuánto me sale, cuánto tengo que trabajar.

Interoperabilidad

Este concepto hace referencia al costo de conectar dos sistemas o componentes de dos sistemas o componentes en un mismo sistema. Cuanto menor sea el costo, mayor es la interoperabilidad.

Este costo hace referencia a cuánto trabajo tengo que hacer para poder conectarlo. Está íntimamente relacionado con la facilidad de uso de las interfaces.

Performance

Este concepto hace referencia al tiempo que un sistema tarda en responder ante el pedido de uno de sus clientes con la menor cantidad de recursos. Cuanto menor sea el tiempo, mayor va a ser la performance.

Existen otras formas de medir el rendimiento, además del tiempo de respuesta. La mayoría tienen la forma de <cantidad de algo>/segundos. Por ejemplo, si consideras una base de datos, podrías pensar en cantidad de entradas y salidas por segundo. Si consideras una red, podrías pensar en cantidad de bytes transmitidos por segundo.

Seguridad

Este concepto hace referencia a la habilidad de un sistema de prevenir, detectar y eliminar los accesos no autorizados a la misma. También incluye la habilidad de detectar y resistir a los ataques si no se pueden evitar.

Usabilidad

Este concepto hace referencia a la percepción que tiene un usuario de que tan fácil e intuitivo es usar un sistema.

Conclusión

Un punto importante es que estas mediciones que hagamos de los sistemas no nos dicen mucho por si solas. La “papa” está en comparar las mediciones de varios sistemas.

Otro punto importante es que no podemos decir que un sistema es seguro o es mantenible si no decimos contra que lo estamos comparando. Por ejemplo, el sistema A tarda en responder una

consulta un segundo mientras que el sistema B tarda cuatro segundos. En ese caso, el sistema A es más performante respecto al sistema B, ya que si lo comparas con el sistema C que tarda medio segundo, no va a ser más performante.

Es importante remarcar que estos conceptos se oponen en muchos casos. Por ejemplo, si nos concentramos en la interoperabilidad podemos agregar puntos vulnerables a accesos no deseados, los cuales atentan contra el concepto de seguridad.

En el diseño de sistema siempre existe lo que se conoce como “trade-off” que es, justamente, la oposición entre los atributos de calidad. Es por eso, también, que no existen soluciones que apliquen siempre y en todos los casos, sino que, dependen mucho del contexto y de que los atributos que se deseen buscar.

Telefonía IP (VoIP)

Es una forma de comunicación en donde la voz viaja a través de la red de Internet. La voz se digitaliza y se la divide en paquetes que viajan a través de la red de Internet según decida el router el mejor camino. Cuando llegan a destino se ordena y se convierte a voz.

- Permite reducir los costos (ya que solamente tenés que contratar Internet y no Internet + Telefonía)
- Extensibilidad y Flexibilidad para reconfigurar las redes
- Acceso remoto

VPN

La idea es hacer una red sobre otra red.

¿Por qué querés hacer esto? Porque la segunda red física no es confiable por lo que tus datos pueden ser vulnerados. Para evitar eso, se “construye” una red segura virtual por sobre esa red, de manera que tus datos estén protegidos.

¿Cómo se hace esto? La idea es que los paquetes que viajan por la red insegura viajen encriptados y solamente los “endpoints” (el Router que envía el mensaje y el Router que lo recibe) puedan desencriptarlos.

¿Cuándo querés hacer esto? Te voy a dar un ejemplo:

Supongamos que existe una empresa de salud que tiene un sistema de pedidos de médicos. Este sistema tiene varios ambientes: desarrollo, testing, producción. Para el desarrollo del sistema se contrata a una empresa consultora que va a trabajar de forma remota, es decir, va a trabajar desde

sus instalaciones. Los servidores donde está corriendo la aplicación (tanto desarrollo como testing y producción) están en las instalaciones de la empresa de salud y están todos en la misma red. ¿Cómo hace la consultora para tener acceso a los servidores? A mi se me ocurren tres opciones:

- No tiene acceso: es decir, programan un requerimiento y se lo llevan en un pendrive a los responsables de sistema de la empresa de salud.
- Contratan una conexión punto a punto: es decir, le piden a una empresa de telecomunicaciones (por ejemplo fibertel) que “les tire un cable” desde las oficinas de la empresa de salud hasta las de la consultora. Esto permite que la segunda pueda estar en la misma red en la que están los servidores y así acceder. No nos olvidemos que estas conexiones punto a punto no son baratas.
- Implementan una conexión site to site a través de una VPN: es decir, configuran los routers de salidas de ambas oficinas de manera que la consultora pueda conectarse de forma segura a la red de la empresa de salud y así poder acceder a los servidores.

Este es solamente un ejemplo. También podríamos pensar que un desarrollador quiere trabajar desde su casa (homeworking) y tiene que tener acceso a la red de su trabajo. Se podría contratar una línea punto a punto pero saldría caro. En vez de eso, se conecta a la red a través de una VPN.

Recursos humanos

Trabajo que aporta el conjunto de empleados de una organización.

La misma cantidad de personas puede dar más o menos esfuerzo dependiendo de como se lo gestione, por eso es importante una buena gestión.

Gestión de recursos humanos

Proceso encargado de poner en funcionamiento, desarrollar y movilizar a los recursos humanos para que estos alcancen los objetivos de la organización y aporten la mayor cantidad de esfuerzo (es decir, den lo máximo que pueden dar de sí mismos).

Para lograrlo, se define un plan (o una estrategia, como en todos los casos de gestión) que incluye políticas claras, roles/funciones claras, dependencias claras, objetivos claros, comunicación clara, buena motivación, etc. Esta estrategia se piensa dentro del marco de la organización (es decir, que va a tener en cuenta las oportunidades y limitaciones que presenta dicha organización).

Los pasos son:

1. Relevar la organización (para conocer las oportunidades y limitaciones)
2. Planificar (definir las políticas, roles, dependencias, objetivos, etc)

3. Ejecutar (llevar a cabo el plan para que los recursos humanos aporten la mayor cantidad de esfuerzo)
4. Controlar (no solamente controlar que se cumpla el plan, sino tomar medidas que se transformen en métricas que te permitan saber la efectividad de los planes)
5. Retroalimentar (corregir/actualizar los planes para sacarle mayor provecho a los recursos humanos)

La idea es crear un plan (planificar, paso 2) que, teniendo en cuenta las limitaciones y las oportunidades de la organización (relevación, paso 1), maximice el esfuerzo entregado por los recursos humanos. Ese plan, después, tiene que ser llevado a cabo (ejecución, paso 3) y controlado (control, paso 4) para que se cumplan de acuerdo como fue planeado. Como todos los planes, se deben sacar métricas que permitan tener una retroalimentación (retroalimentación, paso 5) para poder realizar mejoras o correcciones.

Liderazgo

Habilidad gerencial (es decir, que necesitan los gerentes) de una persona para **influir en el comportamiento** otras, de manera que estas alcancen sus metas de forma satisfactoria y con entusiasmo.

Liderar implica administrar (asignar recursos y controlar que se haga lo que se planeó) pero no al revés.

- Saber administrar
- Saber planificar (un poco está dentro de administrar)
- Saber adaptarse al equipo (un poco se debería tener en cuenta a la hora de administrar ya que forma parte de la relevación de la organización)
- Saber delegar
- Saber comunicar
- Saber resolver conflictos.

Si lo tenemos que comparar, podríamos decir que un administrador ve a las personas como máquinas, a quien le da tareas y les dice exactamente como hacerlas. Además, les define objetivos y se los evalúa exactamente por esos.

El líder ve a las personas como personas a las que tiene que ayudar a cumplir sus metas. Los tiene que motivar e inspirar, les tiene que allanar el camino, los tiene que escuchar, etc.

El líder convence, el administrador manda. El líder se basa en la confianza, el administrador en el estricto control.

Cambio

Niveles

1. **Quiebre:** cambio abrupto y total de ideas que surge de una persona en particular como puede ser el gerente.
2. **Transformación:** Cambio gradual y total/parcial de ideas que surge de los empleados.
3. **Cambio:** Cambio abrupto/gradual y total que surge de factores externos como puede ser una nueva ley.

Gestión del cambio

Proceso encargado de identificar, definir, evaluar y controlar los posibles cambios que pueden surgir en una organización. Este proceso se encarga de minimizar el impacto de los cambios.

Fuerzas impulsoras

- Motivación: lograr el cambio prometiendo cosas. No necesariamente monetarias ni física, sino también personales y de “sentimientos”.
- Persuasión: lograr el cambio no prometiendo cosas. Básicamente convencer que debes hacer el cambio.

Fuerzas restrictivas

- Resistencia individual: a la mayoría no le gusta cambiar como hace las cosas
- Resistencia organizacional: a la mayoría no le gusta cambiar como hace las cosas, por el riesgo que genera

Negociación

Interacción humana que tiene por objetivo generar beneficios a ambas partes.

Dos personas distintas tienen intereses encontrados. Una quiere algo de la otra, a una no le gusta algo de la otra, etc. Por lo tanto, acceden a interactuar para encontrar una solución que signifique un win-win para ambos.

Plan de negocio

Evaluación económica y financiera de un proyecto. La parte económica refiere a saber cuánto voy a ganar o perder al finalizar un mes, año o proyecto. La parte financiera refiere a como es el flujo de dinero, es decir, a la plata que entra y sale de mi bolsillo al finalizar un mes, año o proyecto.

Como se ve en Sistemas de Gestión, para la parte económica se aplica el principio de devengado y para la financiera la de percibido.

Puede darse el caso de que un proyecto, al final del año, me deje un millón de pesos, pero en los primeros años tengan que pedir prestado al banco medio millón. En ese caso, la situación económica es buena pero la financiera podría no serlo.

Documento que aclara qué va a hacer una organización en los próximos X años y cómo va a alcanzarlo, es decir, el plan de negocio aclara como llegar a una situación en donde tengo un millón de pesos en activos -por ejemplo-, si hoy en día tengo medio millón -por ejemplo-.

Impuestos

Tributo que toda persona debe pagarle al Estado para financiar las necesidades públicas.

Gravar

Gravar es el acto de imponer un impuesto a algo.

Económicos vs Financieros

Un impuesto afecta económicamente si te produce una ganancia o una pérdida. Es decir, si tu patrimonio neto (diferencia entre todo lo que tienes menos todo lo que debes) aumentó o disminuyó, te afectó económicamente.

Un impuesto afecta financieramente si te produce un ingreso o una pérdida. Es decir, si a/de tu bolsillo entró/salió plata, te afectó financieramente.

Todos los impuestos son financieros porque sale plata de tu bolsillo, salvo que haya crédito fiscal y canceles una deuda nueva con un crédito que tenías con la AFIP.

Directos

IVA

Grava las compras y ventas. Cada vez que se hace una transacción se agrega el impuesto al IVA, es decir, si soy un productor y le vengo a mi distribuidor, al precio de venta le tengo que agregar un

21%. El distribuidor cuando le vende el producto al vendedor, también le agrega un 21% de impuesto al valor agregado. Los consumidores finales tienen que pagarlo si o si cada vez que compran, mientras que los no finales pueden pedir un reembolso, es decir, cuando yo le vengo a mi distribuidor, yo le debo al Estado el 21% del precio de venta. Cuando el distribuidor le vende el producto al vendedor, el primero le debe al Estado el 21% del precio que le vendió al vendedor menos el IVA que pagó al productor.

Vendedor	Comprador	Precio (con IVA=10%)	Deuda con Estado
Productor	Distribuidor	$100 + 100 * 0.1 = 110$	10
Distribuidor	Vendedor	$120 + 120 * 0.1 = 132$	$12 - 10 = 2$
Vendedor	Consumidor final	$150 + 150 * 0.1 = 165$	$15 - 12 = 3$

Como conclusión se ve que el consumidor final termina pagando los \$15 adicionales de IVA. Sin embargo, el consumidor final no se los paga al Estado directamente, sino que los vendedores lo hacen.

No afecta económicamente: ya que si sos productor, lo que te terminan pagando de más se lo debés después al Estado. Es decir, que te aumentó el activo pero en la misma proporción te aumentó el pasivo, por lo que tu patrimonio neto no varió.

IDCB (al cheque)

Grava los créditos y débitos bancarios. Cada vez que se deposita o se debita de una cuenta bancaria (no importa la manera en que se haga), se paga un poquito más a la entidad bancaria que es responsable de pagarle al Estado.

Afecta económicamente (porque te generó una pérdida de plata, tu patrimonio neto varió negativamente).

IIBB

Grava los ingresos provenientes de la explotación del negocio, es decir, grava los ingresos de cada venta que realizás.

Afecta económicamente (porque te generó una pérdida de plata, tu patrimonio neto varió negativamente). Los fabricantes están exentos.

Indirectos

Ganancias

Grava los resultados operativos positivos. Cada vez que realizas el resultado operativo de cada año (puede variar el tiempo), se calcula ganancias y se la paga al Estado.

Afecta económicamente (porque te generó una pérdida de plata, tu patrimonio neto varió negativamente)

Ganancia mínima presunta

Grava los resultados operativos negativos. Cada vez que realizas el resultado operativo de cada año (puede variar el tiempo), se calcula ganancia mínima presunta y se la paga al Estado.

Afecta económicamente (porque te generó una pérdida de plata, tu patrimonio neto varió negativamente)

Cuando se realiza el resultados operativos, se paga el mayor entre ganancias y ganancias mínima presunta. Además, se lo divide en 10 meses esa ganancia y se las paga en los siguientes meses. Esto se puede sumar a los activos en los próximos resultados operativos.

Amortización

Desvalorización periódica que sufren los bienes a causa del paso del tiempo. Las inversiones (bienes y servicios no consumibles como las computadoras) se amortizan. Los gastos (bienes y servicios consumibles como la papelería) no se amortizan.

+ Ventas
- Costo por ventas
Resultado bruto
- Amortizaciones
+/- Otros gastos
Resultado parcial
- Impuesto a las ganancias o ganancia mínima presunta
+ Amortizaciones
Resultado final

Es importante tenerlo en cuenta ya que cuanto más chico sea el resultado parcial, menos voy a tener que pagar de ganancias.

- Afecta económicamente: ya que tu activo vale menos, es decir, tu patrimonio neto disminuye.

- No afecta financieramente: ya que la plata en tu bolsillo es la misma, vos no le pagaste a nadie por esa desvalorización.

Leasing

Contrato (operación financiera) que consiste en **alquilar** un bien por un tiempo con la opción de comprarlo al finalizar el mismo.

La idea es la siguiente:

- Santiago quiere comprar una máquina para su empresa.
- Santiago contacta al proveedor de la máquina, Pablo.
- Santiago y Pablo se ponen de acuerdo en las características de la máquina y el precio.
- Pablo ofrece un leasing a Santiago.
- Santiago contacta al financiador, Jorge.
- Jorge se asegura que Santiago le va a poder pagar las cuotas mes a mes.
- Jorge le dice a Santiago cuánto le va a cobrar de cuota mes a mes.
- Santiago acepta y firma el contrato de leasing por la máquina.
- Jorge le paga a Pablo por la máquina y se la entrega a Santiago.
- Santiago le paga a Jorge mes a mes por la máquina.

Las ventajas que aporta el leasing son las siguientes:

- **Adquirir el bien sin movilizar capital:** Santiago se ahorra de tener que pagar todo el precio de la máquina de una, ya que lo hace en cuotas y a Jorge.
- **Deducción de impuestos a las ganancias:** las cuotas que vaya pagando Santiago pueden ser deducidas del impuesto a las ganancias, ya que se pueden imputar como “gastos de explotación”.
- **Diferir el pago del IVA:** en vez de pagar el IVA de una -al hacer la compra- Santiago va pagando el IVA en cada cuota.
- **Tasas más bajas:** ya que Jorge, como financiador, se asegura de que, si Santiago no le paga, le puede quitar el bien y listo.

Fuente:

- <http://www.iprofesional.com/notas/165859-Por-ventajas-financieras-e-impositivas-se-hace-fuente-el-leasing-para-comprar-autos-y-mquinas>

Abastecimiento

Proceso para adquirir un bien o servicio para contribuir a los objetivos de la organización.

Fases del abastecimiento

1. Definición de necesidades que debe cumplir el bien o servicio y método de evaluación
2. Definición de método de adquisición (licitación, compra directa, etc.)
3. Recepción de propuestas
4. Evaluación de propuestas (de acuerdo al método de evaluación. Podría ser un benchmark o podría ser MEP)
5. Adjudicación de propuesta (firma de contratos, garantías, etc.)
6. Adquisición de bien o servicio (recepción, instalación, postventa)
7. Seguimiento y monitoreo del contrato

Método de evaluación de propuestas (MEP)

Conjunto de procedimiento que permiten comparar y evaluar propuestas del mismo tipo.

Pasos

1. Armar el cuadro de pesos relativos

Debe haber una columna para las categorías, cada una con su nivel de agregación y varias columnas por cada nivel de agregación, indicando que tan influyentes son en la valoración final.

Categorías	Nivel 1	Nivel 2	Nivel General
1 - Físico	45%		
1.1 - Tamaño de pantalla		30%	30% del 45% = 13.5%
1.2 - Duración de batería		70%	70% del 45% = 31.5%
2 - Funcionales	35%		
2.1 - Conectividad		20%	20% del 35% = 7%
2.2 - SSD		50%	50% del 35% = 17.5%
2.3 -		30%	30% del 35% = 10.5%
3 - Costo	20%		20%

Totales	100%	100%	100%
---------	------	------	------

Importante:

- Los totales siempre deben sumar 100%

2. Armar la tabla de valoración de atributos

Debe haber tres columnas: una para la categoría (misma que el cuadro anterior), una para los posibles atributos para esa categoría y otra para el valor que se le da a ese atributo.

Categorías	Atributos	Valor
1 - Físico		
1.1 - Tamaño de pantalla	7"	0
	(7" - 8.5"]	10
	(8.5" - 11"]	100
	> 11"	60
1.2 - Duración de batería	4 hs	0
	(4 - 5] hs	10
	(5 - 6] hs	40
	> 6 hs	100
2 - Funcionales		
2.1 - Conectividad	WiFi	20
	Bluetooth	20
	3G	10
	4G	50
2.2 - SSD	Si	100
	No	0
2.3 -		
3 - Costo	\$ [100, 200]	$f(\text{costo}) = x - 100$

Importante:

- Solo se ponen los atributos para las últimas desagregaciones
- En el caso de Tamaño de pantalla y Duración de batería, se dice que son M.E. Por lo tanto, un atributo debe tener valor 0, otro 100 y los otros entre 0 y 100
- En el caso de Conectividad, se dice que es Aditiva. Por lo tanto, la suma de los atributos debe ser 100
- En el caso de SSD, se dice que es Binaria. Por lo tanto, un atributo debe ser 100 y el otro 0
- La función de costo se calcula teniendo en cuenta un costo mínimo y un costo máximo. El costo mínimo debería dar el valor 100, mientras que el máximo debería dar 0.

3. Armar la tabla de ponderación de propuestas

Debe haber una columna para las categorías (las mismas que en las dos tablas anteriores), una para el peso (que es el porcentaje del nivel general de la primera tabla) y tres columnas por cada propuesta, donde la primera sea atributo (la característica real de la propuesta con respecto a esa categoría), otra sea valor (el valor que ese atributo tendría, según la segunda tabla) y la ponderación (que es la multiplicación del valor obtenido por el peso dividido 100)

Categorías	Peso	Propuesta 1		
		Atributo	Peso	Ponderación
1 - Físico				
1.1 - Tamaño de pantalla	13.5%	7"	0	0
1.2 - Duración de batería	31.5%	6 hs	100	31.5
2 - Funcionales				
2.1 - Conectividad	7%	Wifi, 4G	70	$0.7 * 7\% = 4.9$
2.2 - SSD	17.5%	SI	100	17.5
2.3 -	10.5%		50	$0.5 * 10.5\% = 5.25$
3 - Costo	20%	175	75	$0.75 * 20\% = 15$
Total				74.15

4. Armar el cuadro de costos

Categoría	Propuesta 1	Propuesta 2	Propuesta 3
Precio de compra	175	160	180
Precio de garantía	50	60	40
Valor residual (el valor al que vas a vender el bien después de usarlo. Se estima)	-30	-20	-60
Total	195	200	160

Punto de ponderación

Un punto de ponderación, es un punto más o menos en el total de la ponderación del producto. Pero, ¿por qué es importante? Porque se analiza cuánto vale un punto más en la ponderación de la propuesta. Para eso, se calcula el valor de un punto de ponderación (VPP).

$$VPP = \frac{CM - Cm}{\text{peso del costo}}$$

Siendo:

- CM: Costo máximo con margen de seguridad
- Cm: Costo mínimo con margen de seguridad

La idea a la que responde el VPP, es ¿cuánto estoy dispuesto a pagar por el aumento en un punto en la ponderación de esa propuesta?

Por ejemplo, tengo dos propuestas, una me dan una pc con 4GB de RAM y la otra con 8GB. La primera propuesta me dio una valoración total de 70 y la otra de 71. Se presentan dos escenarios:

- VPP = 200 => ¿estoy dispuesto a pagar \$ 200 por 4 GB más de RAM?
- VPP = 2000 => ¿estoy dispuesto a pagar \$ 2000 por 4 GB más de RAM?

Este cálculo le da información al administrador de recursos para tomar la decisión de si conviene comprar una PC con 4 u 8 GB de RAM.

Proyecto

Emprendimiento temporal, realizado para crear productos o servicios únicos, hecho por un equipo de personas dirigidas por un Project Manager y con recursos limitados.

Conjunto de actividades interrelacionadas (emprendimiento) para lograr un objetivo (crear productos o servicios únicos) con una serie de limitaciones (recursos limitados): tiempo, recursos, alcance.

Se diferencia del concepto de proceso en que este último es repetitivo, en cambio, el proyecto tiene un comienzo y un final, es decir, es temporal.

Los proyectos surgen de una necesidad y finalizan cuando se cumplen los objetivos (satisfaciendo la necesidad), cuando la necesidad deja de ser necesidad o cuando se agotan los recursos.

Una definición formal nos dice que es un esfuerzo planificado, temporal, realizado para crear productos o servicios únicos que agreguen valor o cause un cambio retroactivo. En cambio, en los procesos se opera en forma permanente, creando los mismos productos o servicios una y otra vez.

Fases de un proyecto

1. Pre-proyecto: factibilidad técnica, económica y operativa.
2. Inicio: gestión del proyecto. Se realizan todos los planes para gestionar cada parte del proyecto.
3. Ejecución: ejecución de los planes junto con el control y seguimiento de cada uno de ellos.
4. Control y Seguimiento: si bien no es una fase en si, forma parte del proyecto desde el inicio hasta el cierre.
5. Cierre: comprobar objetivos, evaluar resultados y realizar/proponer mejoras.

Fallos de un proyecto

- Objetivos poco claros (no se sabe que quiero hacer ni como saber si lo logré)
- Roles poco claros (no saber quien tiene la autoridad para decidir, retrasa tiempos)
- Responsabilidades poco claras (no saber quien tiene que hacer cada cosa, retrasa tiempos)
- Mala comunicación (un mensaje que no llegó a quien debería haber llegado, retrasa tiempos)
- Mala estimación (no se tienen en cuenta contingencias, se prometen cosas que son imposibles solamente por tener el trabajo)

- Mala coordinación (que dos personas hagan lo mismo, retrasa tiempos)
- Nula gestión de riesgos (un riesgo que se convierte en problema que se podría haber evitado, retrasa tiempos)
- Nula gestión de cambios (un cambio que afecta el funcionamiento de otro componente, retrasa tiempos. Con gestión de cambios, se podría haber advertido la situación)

Gestión de proyectos

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del mismo.

La idea es definir, planear, controlar y retroalimentarse para mejorar la eficiencia.

Para gestionar un proyecto se deben tener en cuenta las siguientes áreas a atacar:

- **Alcance del Proyecto:** describe los procesos requeridos para asegurar que el proyecto incluirá todo trabajo requerido, y sólo el trabajo requerido. Consiste de la definición del alcance, planeación del alcance, verificación del alcance, y control de cambio al alcance.
- **Tiempo del Proyecto:** describe los procesos requeridos para asegurar que el proyecto termine a tiempo. Consiste en la definición de las actividades, planificación de las actividades, estimación de duración de las actividades, planeación del cronograma y control del mismo.
- **Costos del Proyecto:** describe los procesos requeridos para asegurar que el proyecto se completará dentro del presupuesto aprobado. Consiste en la definición de los recursos, planificación del uso de los recursos, estimación de costos, presupuestación de costos, y control de costos.
- **Calidad del Proyecto:** describe los procesos requeridos para asegurar que el proyecto satisficará las necesidades para lo cual fue desarrollado. Consiste en la definición de la calidad, planificación de controles para asegurar de la calidad, y control de calidad.
- **Recursos Humanos del Proyecto:** describe los procesos requeridos para asegurar el uso más eficiente de las personas involucradas en el proyecto. Consiste en la definición de los recursos, planeación organizacional, adquisición de staff, y desarrollo del equipo.
- **Comunicaciones del Proyecto:** describe los procesos requeridos para asegurar la generación apropiada y a tiempo, colección, diseminación, almacenamiento, y la disposición final de la información del proyecto. Consiste en la definición de los documentos de comunicación, planeación de la comunicación, distribución de la información, reportes de desempeño, y el cierre administrativo.

- **Riesgo del Proyecto:** describe los procesos requeridos para asegurar el control de los riesgos del proyecto. Consiste en la identificación del riesgo, cuantificación del riesgo, planificación de la respuesta al riesgo, y el control de la respuesta al riesgo.
- **Abastecimiento del Proyecto:** describe los procesos requeridos para la adquisición de bienes y servicios de fuera de la organización ejecutora. Consiste en planear la solicitud, la solicitud, elección de proveedores, administración de contratos, y cierre de contratos.

Fuente:

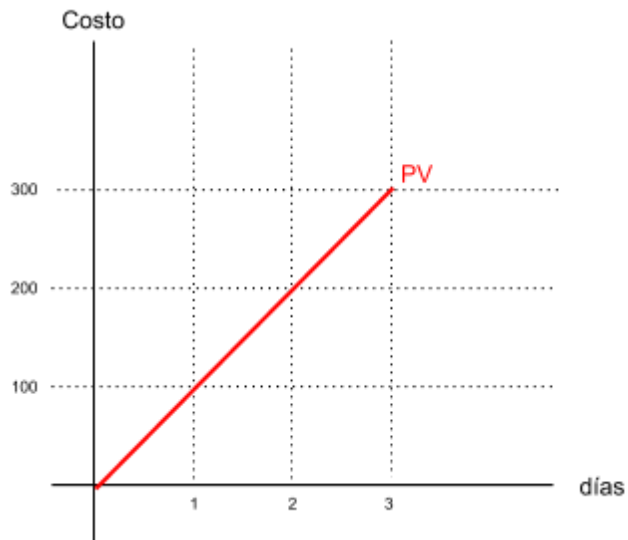
- <https://es.wikipedia.org/wiki/Proyecto>
- <http://www.monografias.com/trabajos12/pmbok/pmbok2.shtml#ADMALCANCE>

Gestión de valor ganado (EVM)

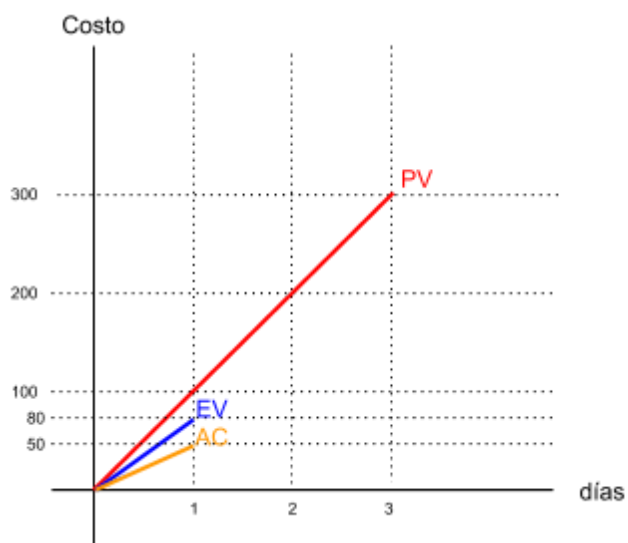
Técnica/**Método** de gestión de proyectos que permiten **medir** -tomar medidas cuantitativas- y **evaluar** -comparar con resultados esperados- **el estado de un proyecto** de manera objetiva a través del presupuesto y del cronograma. Es decir, permite realizar un control y seguimiento del proyecto, permitiéndote ver los desvíos con respecto a lo planificado para hacer correcciones.

- **PV:** Costo que se presupuestó para el trabajo planeado. Lo que tendría que haber hecho hasta el día analizado en términos monetarios. Es decir, es la referencia que vamos a tomar para realizar las evaluaciones a lo largo del proyecto.
- **EV:** Costo que se presupuestó para el trabajo realizado. Lo que hice hasta el día analizado. Es decir, es el trabajo que hiciste hasta ahora, pero en términos monetarios.
- **AC:** Costo que se gastó en el trabajo actual. Lo que gasté hasta el día analizado por el trabajo hecho. Es decir, es la cantidad de plata que gastaste para realizar todo el trabajo que hiciste hasta hora.

Si tengo un trabajo que se realiza en tres días, cada día un tercio del trabajo, donde cada día planeo que cuesta \$100, la variable PV va a ser \$100 en el día uno, \$200 en el día dos y \$300 en el día tres.



Si llegado el día uno hice \$80 del trabajo, la variable EV va a ser \$80. Puede pasar que aunque hice \$80, gasté \$50, por lo tanto, la variable AC va a ser \$50.



- **CV:** varianza del costo = $EV - AC$. Indica cuánto más arriba o abajo del presupuesto estoy.
- **SV:** varianza del cronograma = $EV - PV$. Indica cuánto más arriba o abajo del cronograma estoy.

En el ejemplo anterior, en el día uno, la variable CV va a ser \$30 por lo que voy a estar mejor que el presupuesto que planeé. Por otra parte, la variable SV va a ser -\$20 por lo que voy a estar peor que el cronograma que planeé.

- **CPI:** indicador del rendimiento del costo = EV / AC . Es otra forma de ver el CV.
- **SPI:** indicador del rendimiento del cronograma = EV / PV . Es otra forma de ver el SV.

A partir de estos valores se pueden analizar los rendimientos.

- **BAC:** *Budget at Completion*. Costo que se presupuestó para todo el proyecto. BAC = valor del PV final (300 en el ejemplo)

- **EAC:** *Estimated at Completion*. Estimación del costo total del proyecto cuando se complete:

$$EAC = AC + \frac{(BAC - EV)}{CPI} = \frac{BAC}{CPI}$$

- **ETC:** *Estimated to Complete*. Estimación del costo que todavía le falta al proyecto para completarse.

$$ETC = EAC - AC$$

Riesgo

Evento que, si se materializa, tiene un impacto negativo.

Tiene una probabilidad de ocurrir asociada y un impacto si sucede. La multiplicación de ambas características determina la severidad.

$$\text{Probabilidad de ocurrencia} * \text{Impacto} = \text{Severidad del riesgo}$$

A los riesgos se los puede eliminar, trasladar a terceros o mitigar.

Gestión del riesgo

Proceso encargado de identificar los riesgos y determinar su probabilidad e impacto para poder planificar en consecuencia

Los pasos son

1. Identificación de riesgos
2. Análisis de los riesgos
3. Selección de los que serán tratados (es decir, no todos los riesgos merecen la pena ser tratados. Todo depende de su severidad. No existe una regla de “estos riesgos hay que tratarlos y estos no hacen falta” sino que depende más de la experiencia del PM y lo que dicen las “buenas prácticas”)
4. Planificación de las acciones
5. Control y seguimiento de los riesgos

6. Retroalimentación (esta realimentación es la que permite mejorar los planes, ajustar las acciones o dejar de seguir/controlar un riesgo. Por ejemplo, un riesgo puede dejar de tener probabilidad de ocurrir a partir de una nueva ley)

Tratamiento de riesgos

- **Aceptar:** no hacer nada porque el impacto es más bajo que el costo de hacer algo.
- **Mitigar:** bajar la probabilidad.
- **Trasladar:** transferir el impacto a un tercero.
- **Prevenir/Eliminar:** eliminar el impacto.

Problema

Evento que afecta (positivamente o negativamente) a los objetivos de un proyecto. También se lo define como un riesgo que se materializa.

Indicadores de proyectos

VAN: valor actual neto

Mide la rentabilidad como valor absoluto después de recuperar la inversión, es decir, te dice **cuánto dinero ganaste de forma neta** (ganancias totales - inversión)

La desventaja es que no toma en cuenta la duración del proyecto.

TIR: tasa interna de retorno

Mide la rentabilidad como porcentaje, es decir, mide la tasa de interés que hace que el VAN sea 0, es decir, la tasa de interés que hay que tener para recuperar la inversión.

La desventaja es que no toma en cuenta la duración del proyecto como el VAN y, además, no sirve para comparar proyectos por sí solo ya que se necesita saber también el costo de oportunidad.

PRI: período de recuperación de la inversión

Mide en cuánto tiempo se recupera la inversión, es decir, te dice el momento en dónde se empieza a ganar dinero y dónde se empieza a calcular el VAN. Además, en ese momento, el VAN es igual a 0.

ROI: costo beneficiario

Mide el beneficio que se obtiene por cada unidad invertida durante un periodo de tiempo. Cuando se recupera la inversión, es decir, en el PRI o cuando el VAN es igual a 0 entonces el ROI es igual a 1.

Auditoría

Proceso sistemático -seguidilla de pasos- hecho por uno o varios auditores a una organización o empresa en donde se revisa, evalúa -compara contra una referencia- y se genera un informe sobre lo analizado.

Generalmente cuando se habla de auditoría se lo hace desde el lado contable, para que los auditores (expertos en la materia) opinen sobre la razonabilidad del estado contable y del cumplimiento de las normas.

Verificaciones a los sistemas

- **Vigencia:** de los objetivos planeados como base del diseño original
- **Concordancia:** del sistema con los objetivos
- **Permanencia:** del diseño, sin degradarse
- **Eficiencia:** del sistema

Objetivos de la auditoría

- Resguardo de activos empresariales.
- Cumplimiento de normas y regulaciones internas y externas.
- Confiabilidad de procesos
- Eficiencia de los procesos y de la utilización de los recursos para llevarlos a cabo.
- Exactitud e integridad de datos.

Auditoría de sistemas

Evalúa los mecanismos de control que están implementados en la organización, determinando si son adecuados y cumplen con los objetivos.

Mecanismos de control a evaluar

- **Preventivos:** evitar error, omisión o actos maliciosos.
- **Correctivo:** corregir los errores, omisiones o actos maliciosos.
- **Detectivos:** detectar los errores, omisiones o actos maliciosos.

Pasos de una auditoría

1. **Planificación:** se definen cosas como a quién entrevistar, qué documentos mirar, etc en base a la evidencia que envía la organización y en base a la experiencia que te dice qué áreas son más propensas a fallar.
2. **Identificación y evaluación de riesgos:** se analizan las diferentes contingencias que pueden surgir a la hora de auditar y se planean en consecuencia.
3. Se crea el **cronograma:** de tareas y fechas.
4. **Proceso de auditoría:** se piden y revisan las muestras.
5. **Evaluación** de controles: se determinan cuáles controles fueron aprobados y cuáles no.
6. Creación de un **informe:** dirigido al que contrató la auditoría con hallazgos, opiniones sobre los sistemas y como reaccionaron ante los controles, sugerencias, recomendaciones y anexos.

Básicamente lo que se hace es lo siguiente:

Santiago tiene que auditar la empresa Marcadores S.A., entonces lo primero que hace es mandarle un mail a Pablo que es el contacto de la empresa, pidiéndole información sobre la misma. Por ejemplo, organigrama, información contable, documentación sobre los sistemas que usan y los procesos que siguen, etc.

Santiago, con toda esa información, se hace una lista de las cosas que le parecen sospechosas o sabe que generalmente no son del todo correctas. También piensa en los posibles problemas que puede tener a la hora de auditar, como por ejemplo, si un empleado no fue a trabajar ese día y qué hacer en tal caso. Por último, como sabe que no va a llegar a revisar todo en un día, hace un pequeño cronograma de las cosas que tiene que hacer cada día y con quien se debería reunir. Parte de esta info, se la manda a Pablo para que la vaya preparando -si hace falta presentar algún documento- o que advierta a los empleados -si hace falta tener alguna entrevista.

Llega el día de la auditoría y Santiago se presenta en las oficinas de Marcadores S.A., se saluda con Pablo y empieza a realizar las tareas que planeo. Revisa documentos, revisa procesos, revisa sistemas, entrevista gente. Audita. Mientras tanto, va tomando nota de todas las evaluaciones que fue haciendo.

Con toda la información que fue recabando, Santiago arma su informe, indicando cuales controles se pasaron y cuales no. Además, incluye recomendaciones y sugerencias para mejorar en los puntos flojos.

Controles de la auditoría a los sistemas

- A los logs (o pistas de auditoría) para las operaciones pueden ser rastreadas en todo momento.
- A la entrada de los datos para que no se dejen entrar a los sistemas datos erróneos.
- A la integridad de los datos en el proceso para que no corrompan.
- A las transacciones para que sea correctas.
- Al manejo de excepciones y fallas para que no rompan el sistema y se reporten.
- A los sistemas para que se comporten según la documentación.
- A la documentación para que se comporte bajo el marco de las políticas y objetivos organizacionales.
- A la seguridad del sistema para que protejan los datos de la organización.

Áreas que revisa la auditoría de sistemas

- A los controles generales
- A los procesamiento de información como transacciones
- A los controles de seguridad
- A los sistemas de software
- A los sistemas operativos
- A las metodologías de desarrollo
- A los planes de contingencia

Controles internos

- Contables internos:
 - salvaguarda de activos
 - confiabilidad de contabilidad
- Operativos

- ☐ operativos de procesos
- ☒ Administrativos
 - ☐ eficiencia de área no funcional
 - ☐ Cumplen normas