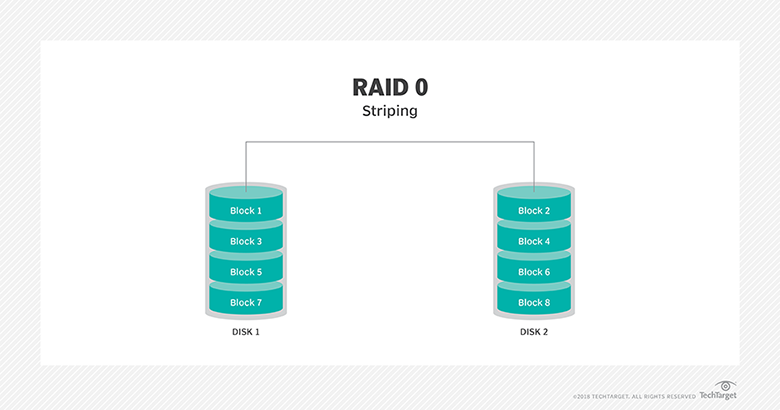
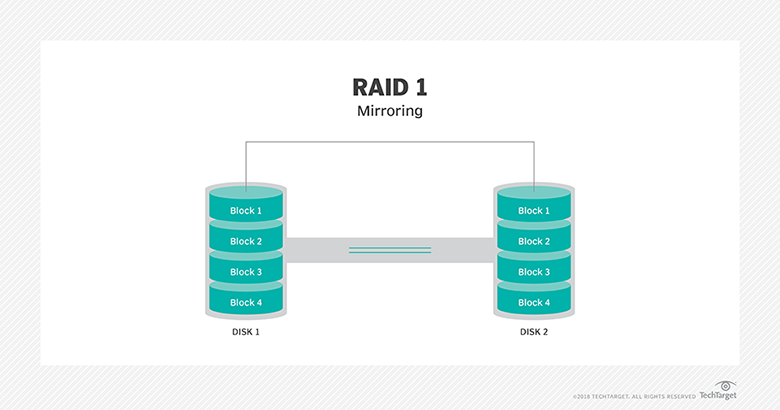
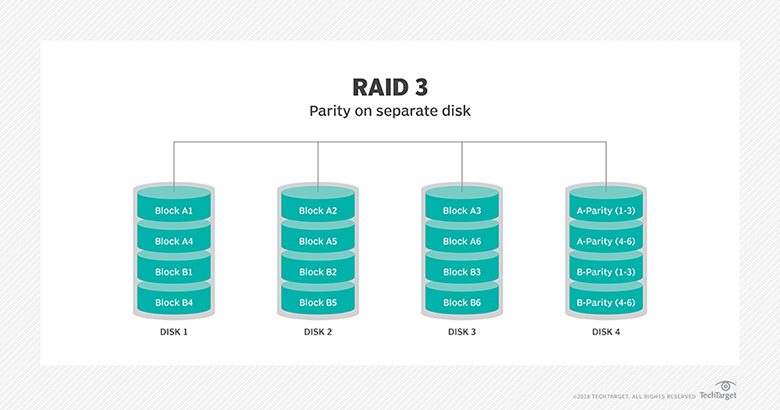
RAID

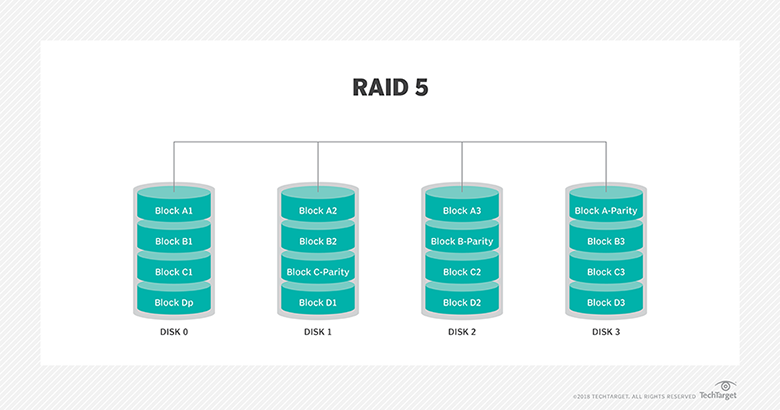
RAID 0: tiene stripping. Da buena performance pero no tiene tolerancia a fallos. Como la información está dividida en dos discos, la lectura y la escritura es más rápida.



RAID 1: funciona como espejo. Mejora la performance de lectura porque puede leer de dos al mismo tiempo pero la de escritura es igual. Tenemos más seguridad pero desaprovechamos capacidad, redundancia por cada disco.



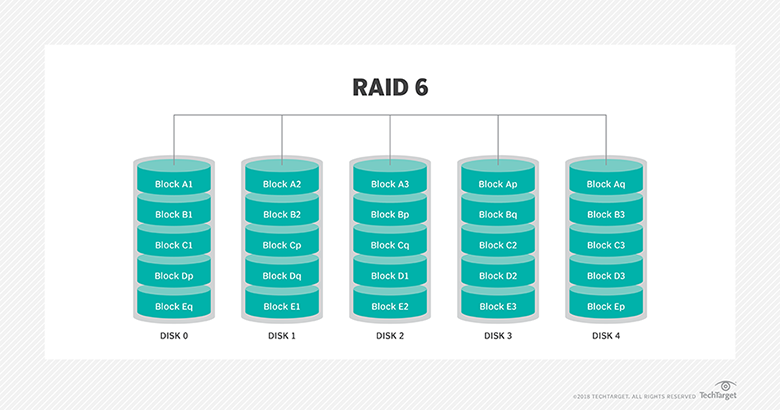
RAID 3: usa stripping y un disco entero para paridad. Puede recuperar data por este disco usando XOR. Por esta razón no puede tener operaciones de lectura/escritura simultáneas, conviene para sistemas mono-usuario. Si se cae el disco de paridad perdés todo.



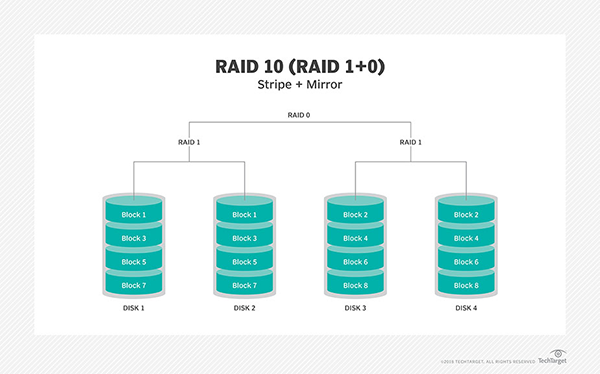
RAID 5: reparte la paridad en los distintos discos, en el caso de que se caiga cualquier disco podés recuperar la información, pero perdés mucha performance en el momento de la restauración.

La performance de lectura es mejor que en un disco solo pero no tan buena como en un RAID 0. Empeora la performance de escritura por el tema de la paridad.

Se puede a partir de 3 pero se recomienda que haya mínimo 5 discos.



RAID 6: tiene doble paridad, lo que te permite que te fallen dos discos al mismo tiempo. El problema es que empeora la performance comparado con RAID 5 y también el costo por Giga.



RAID 1 + 0: hace stripping del espejo. Es más caro porque necesita mayor cantidad de discos y aumenta el costo del Giga por la cantidad de redundancia, pero es más confiable.

RAID 0 + 1 es igual pero al revés.

Software base (SO)

Es un conjunto de software de sistema que controla cómo las aplicaciones acceden al hardware (Ej: Dispositivos de entrada / salida y administración de memoria) y controla la manera en que los usuarios acceden a los recursos. Permite almacenar, procesar y recuperar datos de manera abstracta a la implementación interna del procesamiento y/o del almacenamiento de las estructuras de datos dentro del hardware.

Criterios de elección:

* Seguridad
* Administración de tareas (mono o multi → cuantos procesos puede ejecutar al mismo tiempo)
* Administración de usuarios (mono o multi, el último tiene que tener protección de datos por si se tocan los mismo recursos al mismo tiempo)
* Manejo de recursos (centralizado o distribuido)
* Aplicaciones disponibles
* Estabilidad y confiabilidad (que se usen bien los recursos, que se traten bien los bugs, etc.)
* Escalabilidad (cantidad de memoria, de procesadores, si se puede hacer clustering, replicación, LB)
* Usabilidad
* Licenciamiento
* Soporte / Disponibilidad de RRHH capacitados
* Open Source o no

Virtualización

Vistas: puedo hacer que una computadora sea vista como muchos recursos diferentes o que muchos recursos se vean como uno solo.

Para vs. full VER☀️

La virtualizacion por HW requiere de HW y SW de virtualización. → V. Necesitas el SW para poder hacerlo, pero además el HW tiene que ser virtualizable (actualmente todos lo son).

Tipos:

* Acceso: me permite que cualquier dispositivo pueda entrar a cualquier app y viceversa. Ej.: Citrix, Microsoft Terminal Server, escritorio remoto.
* Aplicaciones: tecnología software que permite que las aplicaciones corran en diferentes sistemas operativos y plataformas de hardware (contenedor, simil app portable). Ej.: App-V, escritorio remoto.
* Almacenamiento: se oculta dónde están los sistemas de almacenamiento y qué tipo de dispositivos realmente almacenan aplicaciones y datos.
* Servidores: existe de tipo 1 (sobre el HW) y de tipo 2 (sobre el SW).
* Procesamiento: un sistema se muestra como varios (máquina virtual) o varios sistemas se muestran como uno solo (load balancer). Equivalente a la de Servidor de tipo 1.
* Red: me varia como veo la red, la vista física es diferente a la lógica. Ej.: Ruteo, NAT (red privada se disfraza de la pública para salir).

Iaas

Se contrata capacidad de **proceso** (CPU) y **almacenamiento**. En este entorno se puede desplegar aplicaciones propias que por motivos de coste o falta de conocimientos no queremos instalar en nuestra propia empresa. El proveedor se encarga de su gestión y para el cliente se convierten todos los gastos en variables (sólo se paga por lo que se usa).

Ejemplo: la conocida EC2 de Amazon y Azure de Microsoft.

Paas

Se proporciona además un **servidor** de aplicaciones (donde se ejecutarán nuestras aplicaciones) y una **base de datos**. Podremos **instalar las aplicaciones y ejecutarlas.** Normalmente hay que seguir una serie de restricciones para poder desarrollar estas para un proveedor (por ejemplo en cuanto a los lenguajes de programación).

Ejemplo: Google App Engine

Saas

Es lo que comúnmente se identifica con “cloud”. Es una aplicación para el usuario final donde paga un alquiler por el uso de software. No es necesario adquirir un software en propiedad (como Microsoft Office), instalarlo, configurarlo y mantenerlo.

Ejemplo: Google Docs o Office365.

Lenguajes

Java: Semi-interpretado, orientado a objetos, que corre sobre una máquina virtual. Simple de aprender a usar, ya que es intuitivo y tiene sintaxis similar a la de C. Es un lenguaje multiplataforma y al correr sobre una máquina virtual permite de forma relativamente sencilla portar las aplicaciones a otras arquitecturas.

* Es gratuito y puede armarse toda la infraestructura de desarrollo con productos gratuitos.
* Además como es muy utilizado mundialmente hay mucha comunidad, encontramos librerías gratuitas, foros, tutoriales y ayudas varias.
* Es una tecnología ampliamente probada.
* Tiene una arquitectura de **threads**. Surgen para liberar de trabajo al Sistema Operativo. Es como un proceso pero más liviano y se ejecuta dentro del área de memoria de un proceso ya existente, comparten memoria. Crear y matar hilos es mucho más liviano que matar procesos. La arquitectura de servidor de threads se adapta mucho mejor a la naturaleza de las **aplicaciones Web**, pero requiere una mejor programación.
* Es Open source, por lo cual no existe soporte licenciado, por lo que puede generar desconfianza en las empresas.
* Es un lenguaje de propósito general, que no está específicamente pensado para que un desarrollo sea rápido.
* Se pueden lograr cosas muy personalizadas y es ideal para aplicaciones fuera de lo común, pero no viene listo para ser productivo y eficiente en forma inmediata. Debemos armarnos un framework, recolectar varias apis, etc. Por lo cual contamos con un **tiempo de startup** para poder comenzar a trabajar.

.Net: no es un lenguaje de programación, es una solución entera, incluye un lenguaje IDE, editor visual, servidor web etc. Está desarrollado para utilizar la solución completa.

* Soporte licenciado pero todas las licencias son muy caras.
* Comienzo de desarrollo **inmediato**.
* Desarrollo más rápido que en otras plataformas.
* Aplicaciones integradas en el Framework Microsoft.
* Ampliamente probado.
* Mala adaptabilidad a otros productos (por ej. Google Maps o Google Docs.)
* La oferta de programadores capacitados está en crecimiento pero aún bastante más chica que la comunidad java.

PHP: Es un lenguaje 100% interpretado, no hay una compilación intermedia como en java. El hecho de ser interpretado puro podría parecer que es lento, pero a diferencia de JavaScript, Php fue pensado de entrada para interpretarse lo más rápido posible. Para un **diseñador web**, php es el lenguaje más **simple** de aprender.

* Es open source.
* Simplicidad y facilidad de uso.
* Comienzo de programación **inmediato**.
* No se requieren deploys o compilaciones previas.
* Su Aplication Server Apache es muy confiable. Posee arquitectura de **procesos** en lugar de threads, de allí su **confiabilidad**. Pero genera mucho procesamiento al Sistema Operativo.
* Menor robustez.
* Soporte limitado.

ABAP/SAP: es un producto enlatado genérico que consiste en un modelo armado y probado. Tiene varios módulos ya hechos, ya programados que cubren las necesidades más comunes de una gran empresa.

* Es un producto ya hecho y probado.
* No requiere tiempos de desarrollo previos.
* Permite su parametrización y customización.
* Ya viene con una forma de encarar la solución al problema.
* Tiene un amplio soporte.
* La customización que puede realizarse es limitada.
* El costo del producto es muy alto.
* No hay gran oferta de programadores ABAP, lo cual aumenta el costo de su customización.
* Los tiempos de implementación del productos son altísimos.

Atributos

Interoperabilidad

Tácticas:

* Locate: Los sistemas que operan deben ser descubiertos en tiempo de ejecución.
* Manage Interfaces: Agrega o elimina capacidades de una interface.

Teorema de CAP

Los sistemas de almacenamiento distribuidos (NoSQL) ayudan a crecer horizontalmente. Solo pueden tener 2 de las 3 características siguientes:

* Consistency: todos los nodos tienen la misma información al mismo tiempo. Si algo se rompe durante una transacción, se hace un rollback por completo para volver al estado anterior.
* Availability: siempre se contesta algo, por más que sea un éxito o un fail.
* Partition Tolerance: el sistema se banca cualquier problema en la red mientras no sea de esta por completo. La información está lo suficientemente replicada entre nodos y redes para mantener el sistema arriba ante cualquier eventualidad, no falla aunque se caiga un nodo.

AP: Contesta siempre pero si un nodo se cae puede perder la consistencia. Cassandra.

CP: Siempre me tiene que contestar lo mismo. Entonces, si se me cae un nodo, deja de estar siempre disponible ya que no sabe si tiene o no la última escritura. MongoDB.

CA: No existe como sistema distribuido, no entra en el esquema de CAP en realidad. RDBMS → responde a ACID.

**ACID:**

* Atomicity (atomicidad): Propiedad que verifica que todas las transacciones se confirmen o se aborten. Ante un fallo del sistema no puede quedar ninguna a medias.
* Consistency: Propiedad que verifica que solo se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
* Isolation (aislamiento): propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error.
* Durability: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema. Debe demostrar que puede recuperarse ante: pérdida de energía, pérdida de memoria, etc.

No SQL

Ventajas

* Resuelven problemas de escalabilidad horizontal
* Manejo de enorme cantidades de datos
* Fácil de escalar
* Excelentes tiempos de respuesta en almacenamiento y recuperación

Desventajas

* No cumplen con ACID
* Utilizan mayormente memoria en vez de disco como la principal ubicación de escritura
* Falta de madurez
* Falta de estándares

Clasificación

* Clave-Valor: los datos se almacenan en pares del tipo clave-valor. El valor es un dato de tipo blob. Ej. Riak, Dynamo, Azure, Redis
* Column Family: permiten almacenar claves mapeadas a valores y esos valores agrupados en múltiples familias de columnas siendo cada columna un mapa de datos. Ej. Cassandra, HBase, Amazon SimpleDB
* Basadas en documentos: La base de datos almacena y recupera documentos que pueden estar en XML, JSON o BSON. Ej. MongoDB, Couchbase, CouchDB, Lotus Notes, Oracle NoSQL Database
* Basadas en grafos: permiten almacenar entidades y relaciones entre esas entidades. Tanto los nodos como las relaciones tienen sus propiedades asociadas. Ej Neo4J, InfiniteGraph, OrientDB, FlockDB

Business Intelligence (BI)

Es un término paraguas que abarca los procesos, las herramientas y las tecnologías para convertir datos en información, información en conocimiento y planes para conducir de forma eficaz las actividades de los negocios. Objetivo: asistir en la toma de decisiones y, posteriormente, descubrir conocimiento desconocido para la organización hasta el momento.

Big Data

Conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, tales como bases de datos relacionales y estadísticas convencionales o paquetes de visualización, dentro del tiempo necesario para que sean útiles.

Se relaciona con la virtualización de almacenamiento y procesamiento.

TIA

Estándar para la disponibilidad que solo tiene en cuenta la infraestructura del datacenter (no tiene en cuenta la virtualización, por ejemplo).

El concepto de **TIER** nos indica el nivel de fiabilidad de un datacenter asociados a cuatro niveles de disponibilidad definidos. A mayor número en el Tier, *mayor disponibilidad*, y por lo tanto mayores costes asociados en su construcción y más tiempo para hacerlo.

UPS: es un [dispositivo](https://es.wikipedia.org/wiki/M%C3%A1quina) que gracias a su batería y otros elementos almacenadores de energía, durante un [apagón eléctrico](https://es.wikipedia.org/wiki/Apag%C3%B3n_el%C3%A9ctrico) puede proporcionar [energía eléctrica](https://es.wikipedia.org/wiki/Energ%C3%ADa_el%C3%A9ctrica) por un tiempo limitado a todos los dispositivos que tenga conectados. Otra función que se puede añadir a estos equipos es mejorar la calidad de la energía eléctrica que llega a las cargas, filtrando subidas y bajadas de tensión y eliminando armónicos de la red en caso de usar [corriente alterna](https://es.wikipedia.org/wiki/Corriente_alterna).

A día de hoy se han definido cuatro Tier diferentes, y ordenados de menor a mayor son:

Tier 1: Centro de datos Básico: Disponibilidad del 99.671%.

* El servicio puede interrumpirse por actividades planeadas o no planeadas.
* No hay componentes redundantes en la distribución eléctrica y de refrigeración.
* Puede o no puede tener suelos elevados, generadores auxiliares o UPS.
* Tiempo medio de implementación, 3 meses.
* La infraestructura del datacenter deberá estar fuera de servicio al menos una vez al año por razones de mantenimiento y/o reparaciones.

Tier 2: Centro de datos Redundante: Disponibilidad del 99.741%.

* Menos susceptible a interrupciones por actividades planeadas o no planeadas.
* Componentes redundantes (N+1)
* Tiene suelos elevados, generadores auxiliares o UPS.
* Conectados a una única línea de distribución eléctrica y de refrigeración.
* De 3 a 6 meses para implementar.
* El mantenimiento de esta línea de distribución o de otras partes de la infraestructura requiere una interrupción de las servicio.

Tier 3: Centro de datos Concurrentemente Mantenibles: Disponibilidad del 99.982%.

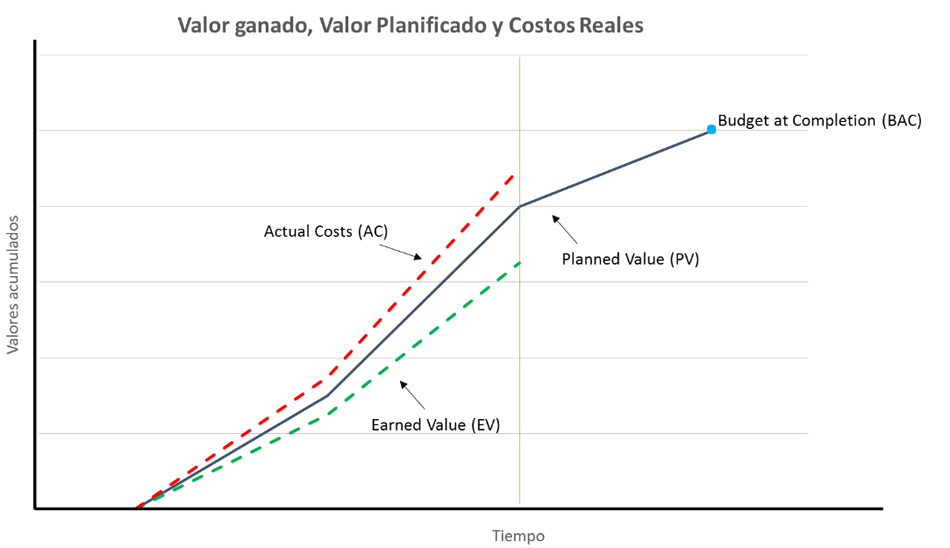
* Permite planificar actividades de mantenimiento sin afectar al servicio de computación, pero eventos no planeados pueden causar paradas no planificadas.
* Componentes redundantes (N+1)
* Conectados múltiples líneas de distribución eléctrica y de refrigeración, pero únicamente con una activa.
* De 15 a 20 meses para implementar.
* Hay suficiente capacidad y distribución para poder llevar a cabo tareas de mantenimiento en una línea mientras se da servicio por otras.

Tier 4: Centro de datos Tolerante a fallos: Disponibilidad del 99.995%.

* Permite planificar actividades de mantenimiento sin afectar al servicio de computación críticos, y es capaz de soportar por lo menos un evento no planificado del tipo ‘peor escenario’ sin impacto crítico en la carga.
* Conectados múltiples líneas de distribución eléctrica y de refrigeración con múltiples componentes redundantes (2 (N+1) significa 2 UPS con redundancia N+1). EJEMPLO FINAL.
* De 15 a 20 meses para implementar.

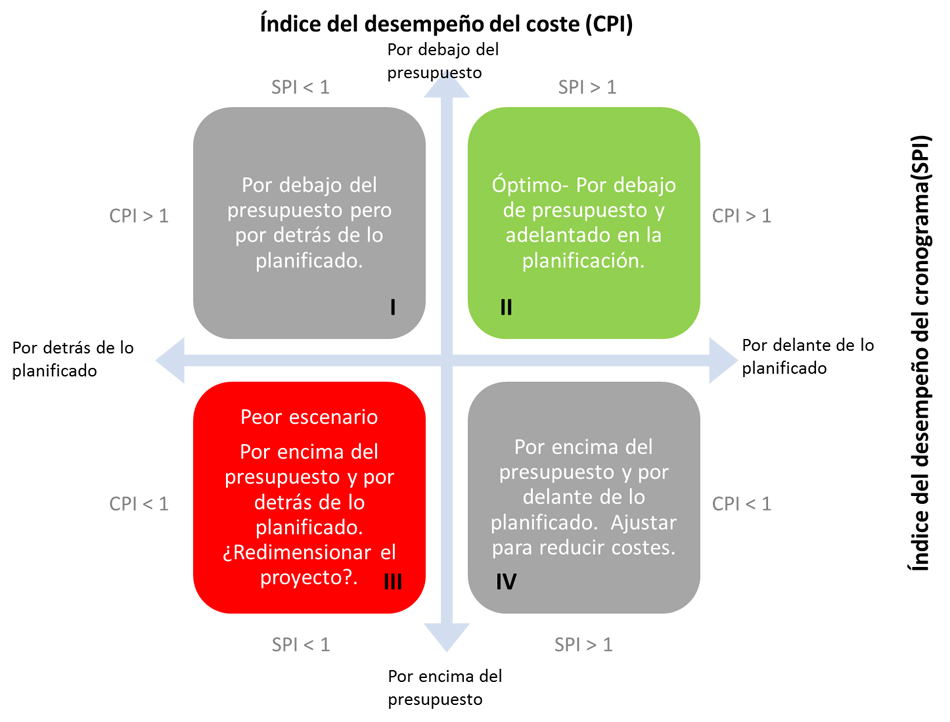
EVM

Es un método para el seguimiento y control de proyectos. Integra alcance, cronograma y costos para medir el rendimiento (CV) y el avance (SV) del proyecto en forma objetiva.

PV: lo que pensaba tener desarrollado para cierto momento. Valor monetario previsto en el plan de proyecto para una tarea.

EV: lo que realmente está desarrollado del proyecto. Valor monetario del trabajo conseguido en el período de evaluación. Para obtenerlo hay que convertir el porcentaje de proyecto completado en un valor monetario multiplicando el porcentaje por el presupuesto.

AC: lo que me salió hacer lo que está desarrollado. Coste actual del trabajo realizado.

CV (Cost Variance): Medida para indicar la desviación de los costes respecto del presupuesto previsto.

CPI (Cost Performance Index): Índice del rendimiento de cada unidad monetaria invertida en el proyecto.

SV (Schedule Variance): Medida histórica para indicar el porcentaje de avance respecto del plan previsto.

SPI (Schedule Performance Index): Índice de eficiencia relativa a cuánto valor se ha conseguido realmente respecto del que está programado para ser llevado a cabo. Porcentaje de avance respecto del plan previsto.

BAC (Budget at Completion): Presupuesto previsto y aprobado para todo el esfuerzo del proyecto.

EAC (Estimated At Completion): Lo que se piensa que se va a gastar al final en base al estado calculado en cierto momento (BAC/CPI).

ETC (Estimated To Completion): Lo que me falta para llegar al final (EAC - AC).

VAC (Variation At Completion): Lo que cambió del presupuesto inicial planeado (BAC - EAC)

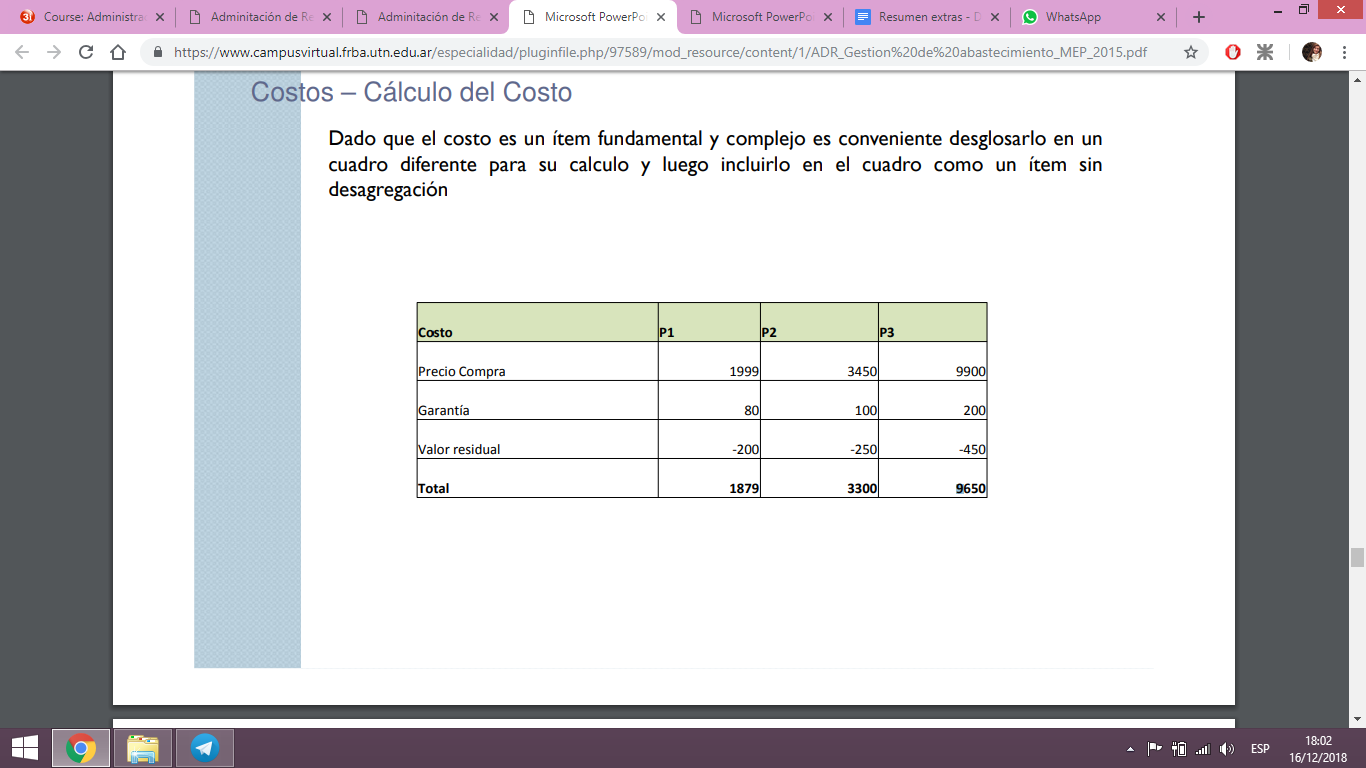
Para que verdaderamente funcione con efectividad, el EVM necesita apoyarse en una planificación que estime de manera objetiva y adecuada todas las actividades y recursos necesarios para la ejecución del proyecto. Sin un plan realista, el EVM no tiene sentido.

MEP

Costos

Para lograr una correcta selección es fundamental determinar cuál será la **vida útil del bien** a adquirir contextualizado en nuestro proyecto. Cualquier bien tiene una vida útil acotada ya sea por desgaste, deterioro o por llegar al límite de sus capacidades. También se debe tener en cuenta que una vez terminada la vida útil del bien en nuestro proyecto éste tiene un valor residual (ya sea en su totalidad o por componentes) que puede ser positivo o negativo en el caso que tengamos que pagar para que sea retirado, por ejemplo.

Se suele hacer una tabla aparte para explicar los costos ya que es un ítem fundamental y estos se deben incluir:



Considerando que el costo es un ítem para el cual tenemos valores continuos es lógico pensar que existe una función continua que define la satisfacción en **función del costo**. Esta función es lineal y con pendiente negativa, considerando que antes de recibir las propuestas de los proveedores, por conocimiento de mercado, tenemos un costo mínimo y máximo aproximado. En base a eso y contemplando un margen de seguridad la función quedaría de la siguiente forma:

F(Costo) = 100 \* (CM - Costo / CM - Cm)

* CM = costo máximo con margen de seguridad
* Cm = costo mínimo con margen de seguridad.

Punto de Ponderación: es el valor expresado en unidades monetarias de la diferencia de importes de costo que generan una diferencia de ponderación igual a 1.

VPP = (CM - Cm) / peso costo → la ponderación que le puse al costo en la tabla

Benchmark

Benchmark es un proceso sistemático y continuo que permite evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones. Estas mediciones son las que van a definirnos cuál es el rendimiento de lo que estamos evaluando en comparación con lo que se está analizando en la medición y así permitir la toma de decisión correspondiente. Podemos medir cualquiera sea la cualidad de lo que nos encontramos evaluando siempre y cuando comparemos las mismas características entre todos los postulantes que están siendo sometidos a través de esta técnica. Sirve para:

* Comparar elementos a través de características claves para la solución
* Obtener un resultado objetivo
* Obtener la mejor relación costo / beneficio
* Comprobar si los elementos estudiados se adecuan a las necesidades

Etapas:

1. Determinar el objeto de estudio
   1. determinar a qué se va a someter el estudio
   2. elegir los factores y variables claves
   3. seleccionar las opciones del mercado
2. Preparar el entorno de prueba
3. Realizar el benchmark
   1. someter los objetos a pruebas
   2. tomar muestras de las respuestas
   3. realizar comparaciones y obtener resultados
4. Analizar resultados
   1. descartar a los que no cumplen las necesidades
   2. realizar informes para mostrar los resultados
   3. determinar si se necesita recalibrar el benchmark
   4. desarrollar planes de acción

TCP - C: Benchmark de bases de datos más reconocido y utilizado en el mercado. Simula una workload con diferentes tipos de transacciones, estas tienen que cumplir ACID. Mide los resultados en transacciones por minuto.

Function points

Una vez que se han obtenido los Puntos de Función sin ajustar del sistema a partir de los Casos de Uso calcularemos los Puntos de Función Ajustados, que consiste en el cálculo de un Factor de Ajuste en base a la cuantificación de ciertos coeficientes vinculados con las características deseadas del sistema (comunicación de datos, rendimiento, facilidades de instalación, de operación, frecuencia de transacciones, etc.).

Se suman los aportes de cada una de las características, obteniendo el grado total de influencia (TDI, del inglés Total Degree of Influence), y se calcula el Factor de Ajuste como:

**AF = (TDI x 0.01) + 0.65**

Finalmente, los Puntos de Función Ajustados se obtienen como el producto de los Puntos de Función sin ajustar por el Factor de Ajuste:

**FP = UFP x AF**

ROI

El ROI (Return On Investment) es el valor económico generado como resultado de la realización de diferentes actividades de marketing. Con este dato, podemos medir el rendimiento que hemos obtenido de una inversión.

ROI = Valor para el negocio / costo del proyecto

PRI

El tiempo que se tarda en recuperar una inversión.

CapEx (Capital Expenditures)

Inversiones de capital que generan un beneficio futuro. Estos costos no pueden deducirse inmediatamente sino que deben hacerlo a lo largo de un período. Están generados fundamentalmente por erogaciones de compra e instalación.

OpEx (Operating Expenses)

Gastos operativos ordinarios. Son deducibles de manera inmediata a su ejecución. Básicamente se compone de las erogaciones incurridas para mantener en funcionamiento lo adquirido e instalado. Dado que OpEx es tres o cuatro veces el CapEx, constituye el foco de los esfuerzos de ahorro.

Los pagos del servicio Cloud mes a mes cuentan como OpEx, así como también la cuota del leasing.

Estándares y normas

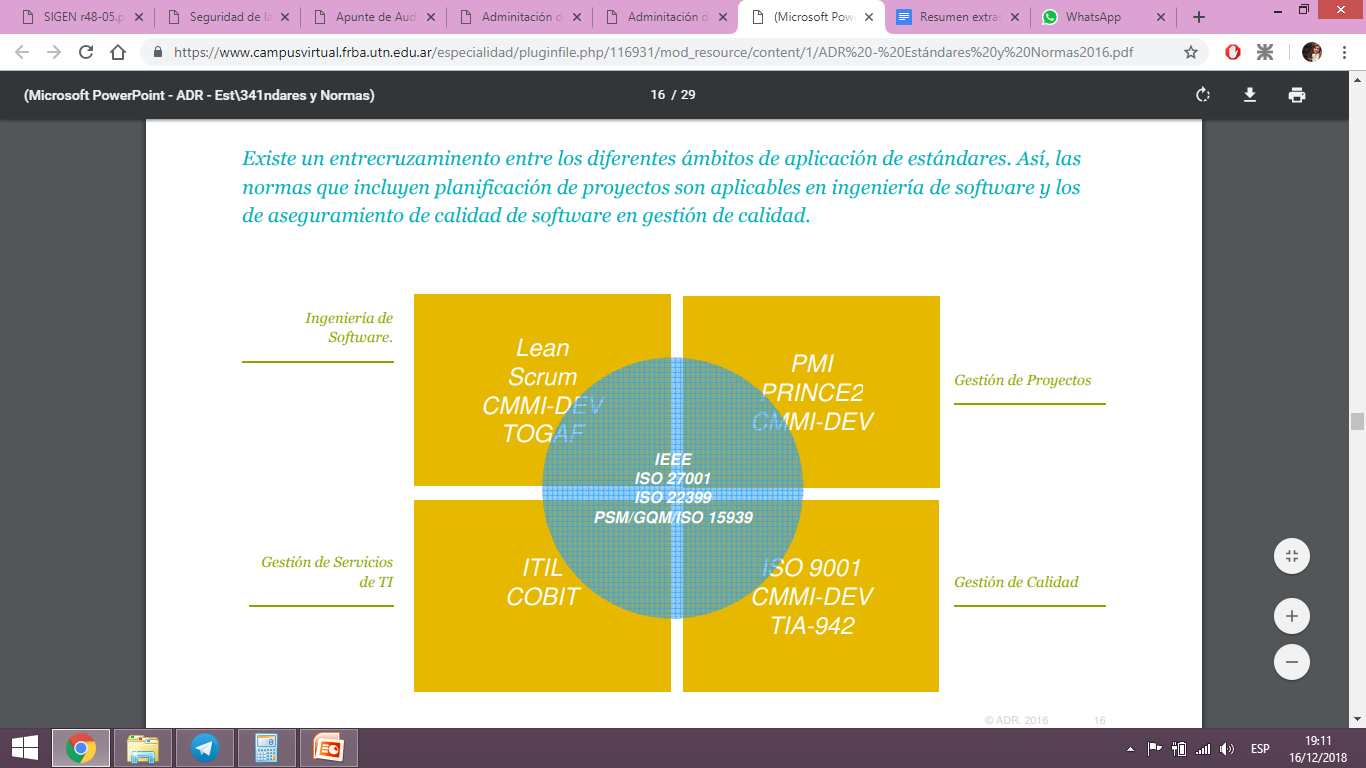
Estándar: Un estándar es un acuerdo sobre la forma de hacer algo. Los estándares no solo hacen la vida más simple sino que tienen una enorme importancia en el aumento de interoperabilidad, efectividad y eficiencia de toda acción repetitiva. Ahorran dinero y aumentan la eficiencia.

Estandarización: Es el proceso de unificación de características en un producto, servicio, procedimiento, etc. Esto implica en muchas ocasiones la redacción de normas de índole prescriptiva que deben seguirse con la finalidad de conseguir objetivos.

Personalización de estándares: Adaptación de estándares de forma de cumplir los objetivos de la organización. Esta adaptación define cómo se aplicarán esos estándares en la organización y puede aplicarse en dos niveles:

* Definición de la forma "estándar" en que se aplicarán en la organización
* Definición de los casos donde se podrá personalizar la forma "estándar" de la organización (situaciones y forma de personalización)

Dado que las normas resultan del análisis de múltiples experiencias exitosas en variados escenarios, el seguimiento de las normas no asegura el éxito pero baja considerablemente la posibilidad de fracaso y provee herramientas para comprender las razones de una falla.



Auditoría

Las auditorías pueden ser realizadas por gente de la misma empresa, o por auditores externos. De allí la división de auditorías externas e internas. El **objetivo** de la auditoría interna es ayudar a la gerencia a ejecutar sus funciones con efectividad.

Generalmente, las auditorías contables son de tipo **externo**, mientras que las operativas son de tipo **interno**. La auditoría de sistemas tiene naturaleza operativa.

Definición: revisión sistemática organizada de los sistemas en funcionamiento para ver si en ellos se verifican las propiedades de:

* VIGENCIA de los objetivos planeados como base del diseño original.
* CONCORDANCIA del sistema con los objetivos.
* PERMANENCIA del diseño por no haber sufrido alteraciones que lo degradan operativamente.
* EFICIENCIA del sistema

Business Plan

Es una evaluación económica y financiera sobre cómo se va a llevar adelante el negocio.

Impuestos directos: gravan al sujeto (impuesto a las ganancias y ganancia mínima presunta)

Impuestos indirectos: gravan al bien o servicio (IVA, ICDB, IIBB).

Amortización: es la depreciación que sufren los bienes por su uso, obsolescencia o transcurso del tiempo. Se contabiliza como una pérdida al depreciarse el bien. Se asocia al concepto de inversión, el cual debe diferenciarse del de gasto.

Se restan antes de calcular las ganancias (+ Ingresos - Egresos - Impuestos = *Resultado* - Amortizaciones = *Ganancias*)

Leasing: Es un contrato de alquiler de un bien (se paga por su uso) teniendo opción de compra del mismo al finalizar el período de uso. Las cuotas son deducibles de impuesto a las ganancias (es un gasto) y el IVA se puede repartir en las cuotas.

Horas efectivas de trabajo: Tienen una implicancia directa en la planificación ya que la duración de las tareas resulta del esfuerzo diario efectivo que puede entregar cada persona.

Costo laboral: Está compuesto por

* Cargas sociales: carga impositiva al empleador (35/37%)
* Aportes y contribuciones: aporte del empleado al sistema laboral (17/19%)

RAID PPTs

RAID 0

Los datos se dividen en pequeños segmentos y se distribuyen entre varios discos.

Ventaja: utilización en operaciones secuenciales con archivos de gran tamaño. Lectura y escritura rápida (Proporcional a los ejes).

Desventaja: no posee tolerancia a fallos, por lo cual podría no ser considerada estrictamente como RAID.

Cantidad de discos: 2

RAID 1

Se basa en la utilización de discos adicionales sobre los que se realiza una copia bloque a bloque.

Ventaja: provee de tolerancia a fallos con alta disponibilidad. Lectura rápida.

Desventaja: Muy costoso (Proporcional al nivel de redundancia)

Cantidad mínima: 2 unidades

RAID 3

Dedica un disco al almacenamiento de información de paridad

Ventaja: altas tasas de transferencia y fiabilidad.

Desventaja: Rendimiento de transacción pobre.

Cantidad mínima: 3 unidades

RAID 5

Utiliza bloques distribuidos para el almacenamiento de información de paridad que le permite ofrecer tolerancia a fallas.

Ventaja: A diferencia del RAID 3, permite la posibilidad de escrituras simultáneas.

Desventaja: A priori, no presenta. Ofrece la mejor relación de precio, rendimiento y disponibilidad

Cantidad mínima: 3 unidades

RAID 6

Implementa doble paridad usando dos discos

Ventaja: A diferencia del RAID 5, permite la falla de dos discos de manera simultánea.

Desventaja: Costo de las controladoras. Aumenta la relación $/Byte por doble paridad

Cantidad de discos mínima: 4

Seguridad de la información

Entendemos la gestión de cambio organizacional como el proceso diseñado que mitiga los efectos no deseados de este mismo cambio y potencia las posibilidades de crear futuro en la organización, su gente y contexto.

La existencia de fuerzas impulsoras (motivación y persuasión) y restrictivas (individual y organización) del cambio está vinculada a los beneficios y costos esperados del mismo.

**Evento**: ocurrencia identificada en un sistema, servicio o estado de una red que indica una posible violación de la política de seguridad o falla en los controles, o una situación previamente desconocida que podría ser relevante para la seguridad.

**Incidente**: evento individual o serie de eventos de seguridad de la información inesperados o no deseados que tiene una probabilidad significativa de comprometer las operaciones del negocio y amenazar la seguridad de la información.