**Administración de Recursos – Prof. Rubén Sualdea – Resumen 2019**

**Control y seguimiento de proyectos**

¿Por qué utilizar una metodología para gestionar?

Para evitar o disminuir las fallas comunes en los proyectos, tales como:

* Falta de coordinación de recursos y actividades.
* Productos finales que no representan lo que el cliente esperaba.
* Proyectos que terminan fuera de tiempo y consumen más dinero del planificado.
* Planificación inadecuada de recursos y actividades.
* Falta de conocimiento del estado real del proyecto.
* Entregables inutilizables o con calidad inaceptables.
* Alcance del proyecto poco definido y falta de gestión de los cambios al mismo.

¿Estándar o metodología?

¿PMBOK o Scrum?

* PMBOK: es un estándar de gestión de proyectos que recoge las mejores prácticas del sector, gestionado y actualizado periódicamente por el PMI.
* Scrum: es un marco de referencia y metodología de gestión de proyectos con enfoque ágil, especialmente útil en proyectos de desarrollo de software, pero también portable a otro tipo de proyectos y sectores.

¿Qué metodología elegir?

|  |  |  |
| --- | --- | --- |
| **Predictivos** | **Iterativos incrementales** | **Ágiles** |
| Los **requisitos** son definidos por adelantado antes de que comience el desarrollo. | Los **requisitos** pueden ser elaborados a intervalos periódicos durante la entrega. | Los **requisitos** se elaboran con frecuencia durante la entrega. |
| Se entregan planes para el eventual entregable. Luego se **entrega** un único producto final al final del proyecto. | La **entrega** puede ser dividida en subconjuntos del producto global. | La **entrega** ocurre frecuentemente con subconjuntos del producto global valorados por el cliente. |
| El **cambio** es restringido tanto como sea posible. | El **cambio** es incorporado a intervalos periódicos. | El **cambio** es incorporado en tiempo real durante la entrega. |
| Los **interesados** clave son involucrados en hitos específicos. | Los **interesados** clave son involucrados periódicamente. | Los **interesados** clave son involucrados continuamente. |
| **El riesgo y los costos** son controlados mediante una planificación detallada de las consideraciones que se conocen. | **El riesgo y los costos** son controlados mediante la elaboración progresiva de los planes con nueva información. | **El riesgo y los costos** son controlados a medida que surgen los requisitos y limitaciones. |

|  |  |
| --- | --- |
| **Metodologías tradicionales** | **Metodologías ágiles** |
| Proyectos de **duración** media o elevada. | Proyectos de corta **duración**. |
| Proceso mucho más **controlado**, con numerosas políticas y normas. | Proceso menos **controlado**, con más flexibilidad. |
| Respuesta lenta a los **cambios**. | Respuesta rápida a los **cambios**. |
| **Gestión de equipos** grandes de personas, generalmente distribuidas. | **Gestión de equipos** pequeños (menos de diez personas). |
| El **cliente** interactúa con el equipo de proyecto mediante reuniones. | El **cliente** suele ser parte del equipo de proyecto. |
| **Curva de aprendizaje** media o larga. | **Curva de aprendizaje** corta. |

Definición de proyecto

* Según estándar PMI: es un esfuerzo temporal comprometido con la creación de un producto o servicio de resultado único.
  + Temporal: significa que un proyecto tiene principio y fin definidos. El proyecto termina cuando ha alcanzado sus objetivos o resulta claro que no pueden alcanzarse o bien la necesidad del proyecto ya no existe.
  + Un proyecto genera productos, servicios o resultados únicos.
  + La elaboración progresiva es una característica de los proyectos que acompaña a la temporalidad y unicidad. Esto significa avanzar en pasos e incrementos continuos.
* Según estándar Prince: es una organización temporal creada con el propósito de entregar uno o más productos de negocio de acuerdo con un caso de negocio especificado.
  + Foco en la justificación del negocio.
  + Enfoque de planificación basada en producto.
  + Estructura organizativa definida por el equipo de gestión.
  + Énfasis en la división del proyecto en etapas gestionables y controlables.
  + Metodología flexible para su aplicación en un nivel apropiado al proyecto.

Ciclo PDCA

Es un ciclo de mejora continua. Se basa en la definición de grupos de procesos.

* Plan: identificar y analizar el problema
* Do: elaborar e implementar una solución
* Check: evaluar los resultados
* Act: estandarizar la solución y capitalizarla en nuevas oportunidades

Fases de un proyecto

Son segmentos temporales en los cuales se dividen los procesos involucrados en el proyecto. Las fases varían según el método de gestión utilizado y además pueden presentar particularidades en un proyecto determinado. Una de las funciones del PM de un proyecto es saber determinar cuáles de estos procesos se utilizarán según el tipo de proyecto gestionado. Estos procesos se dividen en áreas de conocimiento.

* Pre-proyecto: proveer info necesaria para determinar la factibilidad técnica y económica y así decidir si comenzamos el proyecto.
* Inicio: comprender qué producirá el proyecto, cuándo, con qué costo y con qué calidad para elaborar un plan que permita realizarlo.
* Ejecución: elaborar los paquetes de trabajo del producto o servicio, controlar el flujo de trabajo de los equipos, gestionar riesgos y problemas y monitorear e informar el avance del proyecto.
* Cierre: comprobar que todo se ha realizado, determinar qué debe ser pasado a la organización e informar cómo ha finalizado el proyecto.

Riesgos

* Un riesgo es un evento o condición incierta que, si sucede, tiene un efecto en por lo menos uno de los objetivos del proyecto. Es una contingencia que puede llegar a afectar los objetivos del proyecto.
* Los riesgos están asociados a la incertidumbre.
* Se pueden clasificar en:
  + Conocidos: aquellos que identificamos y analizamos y para los cuales podemos planificar respuestas.
  + Desconocidos: aquellos que no podemos gestionar de manera proactiva, pero para los cuales debemos crear un plan de contingencia.
* P x I = S.  Calcular la severidad nos ayuda a decidir cuáles gestionaremos (en gral los de severidad media o alta).
  + Probabilidad de ocurrencia: posibilidad de que el riesgo se materialice.
  + Impacto: resultado de la materialización del riesgo.
  + Severidad: producto de los dos previos.
* Siempre hay riesgos, pero estos pueden o no ser preocupantes.

Gestión de riesgos: si no es posible evitar la aparición de un riesgo, será posible gestionarlo.

* Identificación: reconocimiento de las fuentes de riesgo y sus consecuencias potenciales.
* Análisis: determinación de la necesidad de tratamiento del riesgo y la prioridad de su implementación.
* Tratamiento o respuesta: selección de opciones para actuar sobre el riesgo y la implementación de las mismas.
  + Evitar: eliminar por completo la amenaza. Asegurarse de que la amenaza no ocurra o que no tenga efecto en el proyecto.
  + Transferir: trasladar a un tercero todo o parte del impacto negativo de una amenaza. La transferencia de un riesgo simplemente confiere a una tercera persona la responsabilidad de su gestión, no lo elimina. (Ej: seguros).
  + Mitigar: implica reducir a un umbral aceptable la probabilidad y/o el impacto de un evento adverso. En general es más efectivo que reparar el daño ya causado.
  + Aceptar o asumir: se asume que el riesgo se manifestará y se decide no tomar acción. Esto ocurre debido a que es muy costoso tomar acción sobre él o no se pudo identificar ninguna estrategia de respuesta posible. (Ej: terremotos).
* Monitoreo y revisión: evaluación del progreso en la implementación del tratamiento.

Problema

* Evento o condición esperada o no que afecta negativamente los objetivos de un proyecto.
* Un problema inesperado viene de un entorno ajeno a la compañía. No es un riesgo porque no se puede predecir.
* Un problema esperado puede ser un riesgo que se materializa (deja de ser una posibilidad para transformarse en un hecho).
* No requieren mitigación sino solución.

Gestión de problemas: a diferencia de los riesgos –que representan incertidumbre-, los problemas son hechos sobre los que debe actuar para evitar o minimizar consecuencias negativas sobre los objetivos.

Etapas del proceso de gestión:

* Registro
* Evaluación
* Resolución
* Monitoreo

Seguimiento y control

* ¿Por qué controlar y monitorear? Porque no se puede controlar lo que no se puede medir (Tom DeMarco).
* ¿Cómo controlar y monitorear?
  + Dado que los proyectos evolucionan de forma progresiva, necesitamos realizar mediciones para determinar su grado de avance.
  + Comparando el grado de avance medido con el esperado verificaremos si se observan desvíos significativos respecto de lo planificado. Si ese resulta el caso, tomaremos acciones correctivas.
  + La información de avance del proyecto (incluye alcance, cronograma, costos, calidad, equipo de trabajo, riesgos y problemas) y el análisis de la misma se vuelca en un informe de seguimiento que es tratado por la dirección del proyecto.
* EVM: Es un método para el seguimiento y control de proyectos. Integra alcance, cronograma y costos para medir el rendimiento y el avance del proyecto en forma objetiva.

**Planificación**

Alcance de un proyecto: Es la definición exacta y unívoca de todo lo que estará (y lo que no) comprendido dentro del proyecto a ejecutar. Proporciona un entendimiento común entre los interesados del mismo.

Documento de alcance

* Es la guía del equipo de trabajo durante la ejecución.
* Proporciona la línea base para evaluar si las solicitudes de cambio que puedan surgir se encuentran dentro o fuera de los límites establecidos.
* Base para la estimación de esfuerzo y duración del proyecto.
* Input para la fase de detalle de trabajo/tareas a realizar (EDT).

Características de los requerimientos:

* Necesario: su omisión provoca una deficiencia en el producto. Son los inevitables.
* Conciso: fácil de leer y entender, de redacción simple y clara. Pedido explícito del cliente.
* Completo: proporciona la información suficiente para ser comprendido.
* Consistente: no es contradictorio con otro requerimiento.
* No debe ser ambiguo: tiene una sola interpretación. Su definición no causa confusiones.
* Verificable: puede ser cuantificado a través de inspección, pruebas, análisis.

Herramientas para recolectar los requerimientos:

* Entrevistas: le hago consultas al cliente.
* Talleres: lo veo trabajar y, en función de eso, planteo una solución.
* Brainstorming: todos plantean alternativas.
* Técnica Delphi: requiere de personas inmersas en el tema de las soluciones + soluciones cuantificables.
* Diagrama de afinidad: resolución de problemas similares.
* Cuestionarios: preguntas. Es distinto a entrevistas, no tiene feedback en el momento.

Componentes:

* Descripción clara y unívoca del **objetivo** de proyecto.
* Listado de todos los **requerimientos.**
* Listado y descripción de los **entregables** del proyecto.
* Definición clara de los **límites** del proyecto.
* Descripción de los **supuestos** del proyecto.
* Descripción de las **restricciones** del proyecto.
* **Hitos** del proyecto.

Verificación del alcance

* Es el proceso que consiste en formalizar la aceptación de los entregables del proyecto que se han completado.
* Incluye revisar los entregables con el cliente o el sponsor para asegurarse que se han completado satisfactoriamente y para obtener de ellos su aceptación formal.
* Ocurre al final de cada fase y al final del proyecto.
* Es que el cliente acepte de manera formal el alcance planteado por mí, sumado a que después de cada fase revisemos y demos conformidad de que el alcance se cumplió.

Pasos para desarrollar una planificación

* **Definir** actividades: es el proceso de identificar las acciones específicas a realizar para producir los entregables del proyecto.
* **Secuenciar** las actividades: es el proceso de identificar y documentar las relaciones entre las actividades del proyecto.
* **Estimar recursos** de las actividades: es el proceso de estimar el tipo y cantidad de materiales, personas, equipamiento o suministros requeridos para realizar cada actividad.
* **Estimar la duración** de las actividades: es el proceso de estimar el número de periodos de tiempos laborales necesarios para completar individualmente las actividades con los recursos estimados.
* Desarrollar el **cronograma**: es el proceso de analizar la secuencia de actividades, duraciones,  necesidades de recursos y las restricciones de cronograma para crear el cronograma del proyecto.

Desarrollo de planificación

Herramientas y técnicas para definir actividades:

* Descomposición: el objetivo es descomponer los paquetes de trabajo en actividades/entregables.  Las actividades muestran el esfuerzo necesario para completar el paquete de trabajo.
* Planificación gradual: es una forma de planificación mediante elaboración gradual, donde se planifica en detalle el trabajo que debe desarrollarse en el corto plazo y el trabajo futuro se planifica a un nivel superior de la EDT.
* Juicio de expertos: los interesados en el proyecto pueden proveer info sobre las actividades del proyecto basados en su experiencia, habilidades y conocimientos.

Secuenciar actividades: Método de Diagramación por Precedencia (PPM o PDM)

* Utiliza círculos o rectángulos, llamados nodos, que representan las actividades, así como flechas que interconectan esos nodos y representan las relaciones lógicas. Esto también se conoce como Actividad en el nodo (AON).

Herramientas y técnicas para estimar los recursos de las actividades:

* Juicio de expertos: se utiliza para evaluar las entradas a este proceso. Todo grupo o persona con conocimientos especializados puede aportar dicha experiencia.
* Análisis de alternativas: consiste en encontrar la mejor forma de completar las actividades mediante la combinación de recursos. Tengo que conocer mis recursos disponibles.
* Datos de estimación publicados: índices de producción actualizados y costos unitarios. Extensa variedad de industrias, materiales y equipos. Diferentes países y ubicaciones geográficas.
* Estimación ascendente: consiste en descomponer una actividad o paquete de trabajo en mayor detalle para poder estimarla. Una vez que el detalle de la descomposición está disponible, se suman entre ellos de manera de obtener una estimación más confiable.

Herramientas y técnicas para estimar la duración de las tareas:

* Juicio de expertos: es guiado por info histórica y debe usarse cuando sea posible. Miembros individuales del equipo también pueden aportar. Si no se cuenta con esto, las estimaciones son inciertas y arriesgadas.
* Estimación análoga (top-down): se usa en los comienzos del proyecto, cuando se tiene poca info. Utiliza el costo/tiempo real de un proyecto previo y similar como base para la estimación del costo/tiempo del proyecto actual.
* Estimación por 3 valores: usa un promedio ponderado de estimaciones para calcular la duración de la actividad. Se basa en 3 valores: más probable, optimista y pesimista.

Desarrollar el cronograma:

* Diagrama de Gantt: plasmo tareas y sus dependencias y duraciones, junto con sus responsables.
* Diagrama de hitos: el hito toma 0 hs pero es considerado una actividad. El diagrama plasma cómo vengo con respecto a los hitos definidos.
* Camino crítico: diagrama de caminos hasta llegar al final, calculando el tiempo por tarea.

Estructura de Desglose de Trabajo (EDT)

* Consiste en una descomposición jerárquica del trabajo orientada a entregables.
* El último nivel se denomina “Paquete de trabajo” y es que deberá figurar en nuestra planificación con su esfuerzo asociado.
* Su objetivo es organizar y definir el alcance total del proyecto.
* Se definirán entregables y subentregables en cada nivel hasta que las tareas que impliquen ese entregable permitan:
  + Identificar el grado de avance durante la ejecución.
  + Estimar su duración.
  + Identificar a una persona o grupo responsable.
  + Estimar su costo.
* El costo de una actividad es la sumatoria de los costos de sus tareas.
* El EDT puede plantearse por actividades (es el “cómo”) o por entregables (el “qué”).

Hitos y entregables

* Entregable: objeto tangible producido como resultado del proyecto. (Ej: DER, selección equipos).
* Hito: punto o evento relevante del proyecto. Simboliza el haber conseguido un logro importante en el proyecto. (Ej: relevamiento finalizado, comunicación a la comunidad).
* Ambos tienen duración 0 pero pueden tener, o no, un responsable asignado.
* Los hitos están totalmente ligados a los entregables. La entrega a tiempo de entregables es la evidencia que indica el éxito de un hito.
* Un entregable **difiere** de un hito del proyecto en que el hito es una medida de progreso hacia el resultado final del proyecto, mientras que el entregable es el resultado del proceso.

Gantt

* El Gantt es una herramienta que nos permite modelar la planificación de tareas del proyecto. Permite realizar una representación gráfica del progreso del proyecto.
* Línea base: es la “foto” de mi Gantt completo (del cronograma). Me servirá de referencia, cuando haga cambios, para comparar el desempeño y descubrir desvíos. Puede cambiar a medida que haya cambios en la planificación, pero la LB original se guarda.

**Business Plan**

* Es una evaluación económica y financiera sobre cómo se va a llevar adelante el negocio (lo que hace una organización).
* *Económica* se refiere a si una organización gana o pierde dinero, es decir, el resultado del ejercicio. Para los resultados (ingresos y gastos) se aplica el principio de devengado, esto es, se contabiliza ni bien se conoce independientemente del momento en que el dinero entra o sale de la empresa. Es el dinero que tengo en total.
* *Financiera* se relaciona con el momento en que entra y sale el dinero de la empresa. Es decir al momento en que se realizan cobros y pagos. Es el dinero que tengo en cierto momento.

Impuestos

Nos centramos en los que afectan al flujo de fondos.

* Directos: gravan (imponen) directamente un conjunto de operaciones. Gravan el *sujeto*.
* Indirectos: gravan la operación (transacción comercial). Gravan al *bien* o *servicio*.

|  |  |  |
| --- | --- | --- |
|  | **Nacionales** | **Provinciales** |
| **Indirectos** | IVA, IDCB | IIBB |
| **Directos** | Ganancias |  |

IVA

* No afecta económicamente al flujo de fondos (porque es trasladable), pero sí financieramente (porque se paga todos los meses).
* Se lo considera distorsivo porque las transacciones de compra y venta no suelen ser simultáneas. El contribuyente presta al estado a tasa cero porque se paga el IVA compras antes de vender (y descontarlo del IVA ventas).
* Se grava las compras y ventas devengadas (no las percibidas, aún en las compras a plazo) con total independencia de los plazos de pago.
* Este impuesto incrementa los costos de producción.
* El IVA que pagan las empresas es la diferencia del IVA ventas y el IVA compras.
* Alícuota variable del 0% al 27% dependiendo del rubro.

Ingresos Brutos (IIBB)

* Grava los ingresos provenientes de la explotación del negocio.
* Tasa: 0% a 15%.
* No es trasladable. El fabricante está exento. Como se aplica sobre el total de la operación, puede afectar significativamente la rentabilidad (3,5% es el 17,5% del 20%).

Impuestos a los Débitos y Créditos Bancarios – “Impuesto al cheque” (IDCB)

* Grava TODOS los débitos y créditos bancarios Tasa: 0,6% débitos y 0,6% créditos.
* Las mutuales están exceptuadas.

Ganancias

* Grava las ganancias (ventas - gastos). Tiene una tasa del 35%.
* Cuando dentro de los gastos hay bienes de uso, aunque el contribuyente los haya pagado todos al contado, el fisco considera que es amortizable y cobra el impuesto en base a los años de amortización establecidos para ese bien. Por ende, el contribuyente sólo podrá deducir de ganancias una parte por cada periodo.

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 de *gasto*.
* Inversión: se resta del capital de la empresa y permite aumentar el valor productivo. Está asociada a un bien o servicio NO consumible a corto plazo. La inversión se amortiza. Favorece de alguna forma a mi negocio, por ende, a mi economía. Ej: patentes, rodados, maquinarias, etc.
* Gasto: se resta del capital de la empresa y NO permite aumentar el valor productivo. Está asociado a un bien o servicio consumible a corto plazo. El gasto no se amortiza. No contribuye a mi economía, es dinero que se pierde. Ej: electricidad, teléfono, sueldos, alquiler, papelería, etc.
* Periodo de amortización:
  + Muebles y útiles, rodados: 5 años.
  + Maquinarias, instalaciones: 10 años.
  + Edificios: 50 años.

Leasing

* Es un contrato de alquiler de un bien con opción de compra del mismo al finalizar el periodo de uso.
* Las cuotas son deducibles de impuesto a las ganancias (es un gasto).
* Se aplica cada vez a más actividades y en grandes montos.

Costo laboral

Está compuesto por:

* Cargas sociales: carga impositiva al empleador (35%/37%)
* Aportes y contribuciones: aporte del empleado al sistema laboral (17%/19%)
* Para el cálculo del costo laboral se tienen en cuenta no solo esos dos factores sino también las vacaciones, licencias con goce de sueldo, horas efectivas de trabajo.
* 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.

**Benchmark**

* Un benchmark es un análisis comparativo entre productos, servicios y/o procesos existentes bajo diseño/construcción con el fin de analizar comportamientos, rendimiento, eficacia y/o eficiencia y relaciones costo/beneficio.
* Puede implementarse como una herramienta de medición de rendimiento de una inversión.
* Es un proceso sistemático, continuo y comparativo para llegar a una decisión.
* Sirve para:
  + Comparar elementos a través de características claves para la solución: se estudian los componentes que aportan datos relevantes a la decisión.
  + Obtener un resultado **objetivo**, si bien el procedimiento en el que está incorporado no es objetivo al 100%.
  + Obtener la mejor relación costo/beneficio.
  + Comprobar si los elementos estudiados se adecuan a las necesidades.
* Es importante destacar que un benchmark, por el hecho de encontrarse realizando una comparación, siempre trae aparejada una UNIDAD DE MEDIDA.
* Siempre se deben comparar las mismas características entre todos los postulantes.

Etapas del proceso

1. Determinar el elemento de estudio
   1. Determinar qué se va a someter a estudio.
   2. Elegir los factores y variables claves.
   3. Seleccionar las opciones del mercado con las que se va a comparar.
2. Preparar el entorno de prueba
   1. Recopilar requerimientos del tipo de benchmark.
   2. Realización de tareas previas a la etapa de ejecución (preparación del ambiente en donde se realizará la prueba).
3. Realizar el benchmark
   1. Someter el elemento a las pruebas.
   2. Tomar muestras de las respuestas de las distintas variables analizadas.
   3. Realizar comparaciones y obtener resultados.
4. Analizar resultados
   1. Descartar elementos que no cumplen con las necesidades.
   2. Realizar informes para mostrar los resultados.
   3. Determinar si se requiere recalibrar el benchmark.
   4. Desarrollar planes de acción.

Preparar el entorno de prueba requiere mucho tiempo y dinero, por ende no siempre es conveniente realizar un benchmark. Conviene en casos en los que tengo mucho volumen de información.

Clases y tipos de benchmark

* Rendimiento del sistema (Ej: Hint, SPEC, TPC, Linpack).
* Animación/video (Ej: BAPCO, 3D Mark, Aquamark).
* Dispositivos móviles (Ej: Antutu, Passmark Software, Quadrant).

TPC mide el procesamiento de transacciones en bases de datos.

Tipos de benchmark de sistemas:

* Sintéticos: están especialmente diseñados para medir el rendimiento de un componente individual del hardware, normalmente llevando el componente escogido a su máxima capacidad. (Ej: Whetstone, Dhrystone).
* Aplicaciones: herramientas basadas en aplicaciones reales, simulan una carga de trabajo para medir el comportamiento global del equipo.
* Test de bajo nivel: miden directamente el rendimiento de los componentes. Ej: el reloj de la CPU, los tiempos de la DRAM y de la caché SRAM, etc.
* Test de alto nivel: están más enfocados a medir el rendimiento de la combinación componente/controlador/SO de un aspecto específico del sistema.

**Arquitectura de Software**

* Definición: define de manera abstracta el conjunto de estructuras que la componen. Son elementos de tecnología, relaciones y propiedades entre ellas.
* Objetivos: los sistemas de SW son construidos para satisfacer los objetivos del negocio.
* En qué consiste: Estructuras -> Elementos -> Relaciones. La arq omite ciertos detalles internos de cada elemento para abstraerse de su dificultad y ocuparse de lo exterior.
* Interfaces: dividen lo privado de lo público. Se centran en la complejidad de la interacción de los elementos.
* No todas las arqs son buenas, depende de la estructura de mi empresa. Es buena si se adapta.

Importancia de la arq de SW

* Usuario dependiente de la rapidez, disponibilidad y confiabilidad de los sistemas.
* Cliente preocupado porque se implemente una arq bajo calendario y presupuesto seleccionado.
* Project Manager, preocupado porque los equipos trabajen en forma independiente interactuando con disciplina.
* El Arquitecto se ocupa de que los 3 puntos antes mencionados funcionen correctamente y en forma sincronizada.
* Interesados de una arq:
  + Clientes: es el que solicita el producto.
  + Usuarios: son los que lo van a usar.
  + Project Manager: controla presupuestos, tiempos, etc.
  + Project Leader: define tareas y hace seguimiento del proyecto.
  + Arquitecto
  + Desarrolladores
  + Etc.
* Decisiones de diseño: las toman los involucrados y es importante informarlas. A tener en cuenta:
  + Procesamiento distribuido o no?
  + SW dividido en capas? Cuántas?
  + Comunicación sincrónica o asincrónica?
  + Se depende del SO?
  + Se depende del HW?

Contexto

Las arqs se eligen en base al contexto. El contexto puede ser:

* Negocio: a dónde va la organización.
* Técnico: se basa en las tecnologías usadas, redes, etc. Ej: elegir el SO.
* Ciclo de vida del proyecto: se basa en los planes de entrega, los requerimientos, etc.
* Profesional: a qué están acostumbrados los empleados, qué conocen.

Atributos de calidad

Es una propiedad de medida o de testeo que permite indicar qué tan bien funciona un sistema y cómo satisfacer las necesidades de los interesados. Pueden agruparse en:

* Requerimientos funcionales: qué debe hacer el servicio, con qué tiene que cumplir.
* Requerimientos de calidad del sistema (no funcionales): cómo se hacen los funcionales, de qué manera.
* Restricciones: cosas que no se pueden hacer en la arq.

Algunos atributos de calidad son:

* **Disponibilidad**: que el sistema esté accesible para que uno lo pueda usar.  
  Cómo se logra:
  + Minimizar las interrupciones del servicio.
  + Mitigar posibles fallas que puedan ocurrir.

Tácticas: detectar fallas, recuperación de fallas, prevención de fallas.

* **Interoperabilidad**: capacidad para relacionarse.  
  Cómo se logra:
  + 2 o más sistemas pueden intercambiar info vía interfaces y hasta comprender dicha info.
  + Si conocemos las interfaces de los sistemas externos, donde nuestros sistemas operan, podemos diseñar este conocimiento.

Tácticas: locate (los sistemas que operan deben ser descubiertos en tiempo de ejecución), manage interfaces (agrega o elimina capacidades de una interfaz).

* **Adaptabilidad**: que siga funcionando ante cambios en el entorno. Ante un cambio, qué costo puede tener y a qué riesgo nos expone.
* **Performance**: relación tiempo-habilidad. Es el rendimiento.
* **Seguridad**: que integre elementos de seguridad como autenticación, protocolos, encriptación, etc.
  + Detectar ataques: detección de intrusos, denegación de un servicio, verificación de integridad de msj, atraso en los msjs.
  + Resistir ataques: autenticación de actores, límite de acceso, encriptación de datos.
* **Usabilidad**: representa cuán fácil es para un usuario ejecutar una tarea deseada. Es una de las formas más fáciles de mejorar la calidad de un sistema.
* **Capacidad de prueba y testeo**: si es rápido y fácil de probar. Gran porcentaje del costo de desarrollo es absorbido por las pruebas.
* **Variabilidad**: adaptación al contexto.
* **Portabilidad**: cambios de plataforma.
* **Desarrollo distribuido**: diseño del SW.
* **Escalabilidad**: capacidad de agregar más recursos, como por ej un server.
* **Monitoreo**: investigar el sistema mientras trabaja.
* **Comerciabilidad**: no siempre se adapta a lo que necesitamos.

Tácticas de arq y patrones

Usarlos es una forma de reutilizar el conocimiento arquitectónico.

Un patrón de arq:

* Es un paquete de decisiones de diseño que se encuentra en la práctica.
* Es una solución de diseño probada para un problema conocido.
* Conoce propiedades que permiten reutilización.
* Describe un class de arqs.
* Es distinto a un patrón de diseño de desarrollo.
* Relacionan el contexto con el problema y con la solución.

 ASR en los ciclos de vida

ASR = Requerimientos de Arquitectura Significativos.

Sirven para obtener la info que necesito.

* Reunir los ASRs de los documentos de requerimientos.
* Reunir los ASRs entrevistando a los interesados.
* Reunir los ASRs etendiendo los objetivos del Negocio.

Arquitectura en proyectos ágiles

* Alta satisfacción del cliente cuando se entrega una versión.
* Si cambian los requerimientos, aunque sea tarde, se adapta bien.
* Entregas de software: Entre semanas y meses, tiempos en general cortos.
* Hay gran interacción entre la gente del negocio, y la gente de IT. Interacción cara a cara.
* Motivación del grupo de trabajo.

En muchas organizaciones existe una combinación de arqs que se basan en tipos de proyectos ágiles y arqs de paradigmas estructurados, no siempre hay que caer en lo que ofrece el mercado como solución.

Gestión y gobierno

* Evaluación de una arq:
  + El Arquitecto debe interesarse por la gestión de proyectos.
  + El PM es la persona que junto al arquitecto, deben trabajar en conjunto, por la perspectiva de la organización.
  + A Mayor complejidad de proyectos, más útil es la implementación de una arq.
* Planificación:
  + Sucede constantemente, pero existe un plan inicial para convencer a la dirección de construir el sistema y dar una idea de costo y agenda.
  + El PM debe educar a otros managers para poder corregir desvíos en el desarrollo del SW.
* Organización:
  + Team Leader: gestiona las tareas del equipo.
  + Developer: diseñan e implementan los subsistemas de código.
  + Configuration Manager: ejecutan y construyen test de integración.
  + System Test Manager: testeo de sistema y testing de aceptación.
  + Product Manager: representan el marketing.

Arquitectura cloud

Tengo mis aplicativos en la nube.

* Servicios a demanda: se consumen los aplicativos que se necesitan.
* Acceso único de red: todos entramos por el mismo lugar.
* Pool de recursos: muchos recursos disponibles.
* Independencia de ubicación: no me interesa dónde están físicamente.
* Elasticidad rápida: se obtienen más, rápidamente, ante la necesidad. No hay que contemplar escalabilidad.
* Servicios medidos: se paga lo que se consume.

Los modelos de desarrollo del cloud se diferencian por quienes son dueños y quienes lo operan. Existen 2 modelos básicos que tienen 2 variantes:

* Cloud privado: lo contrato para mi organización.
* Cloud público: los que contrato, los vendo al resto.

Atributos de calidad en arquitectura cloud

El Arquitecto necesita prestar atención en la adaptabilidad, la usabilidad, la interoperabilidad y el testeo, como haría en otra plataforma. “Los atributos de calidad que tienen diferencias, son:

* Seguridad: ante un problema de seguridad, hay responsabilidad del proveedor y de quien lo contrató. La responsabilidad no se delega al 100%. Lo mismo ocurre para los dos siguientes.
* Performance
* Disponibilidad

**Procesamiento de datos**

Se espera que una infraestructura de procesamiento de datos (IPD) brinde:

* Confiabilidad: poder mantener lo actual en el tiempo. Dada por la estabilidad del sistema. Asegura que la tarea encomendada finalice de forma exitosa.
* Rendimiento: predictibilidad del tiempo de obtención del resultado. Depende de la dimensión de la solución.
* Sustentabilidad económica: la mejor solución posible con los recursos que tenemos. Optimización de costo/rendimiento.

Una IPD está conformada por:

* Unidades de procesamiento: equipos de computación.
* Unidades de almacenamiento: elemento donde se persisten los datos a procesar y los resultados.
* Sistemas de comunicaciones: permiten la interconexión física y lógica entre los elementos.
* SW de procesamiento: encargado de aplicar la lógica de procesamiento.
* SW de base: sirve para gestionar la interrelación entre todos los elementos.

Unidades de procesamiento (UPD)

Se espera que una UPD brinde:

* Confiabilidad: estabilidad del funcionamiento de los recursos.
* Disponibilidad: funcionamiento continuo y regular ante eventuales fallas o degradaciones del servicio.
* Tolerancia a fallas: poder seguir operando sin afectar a los servicios ante una ocurrencia de falla. Lo logra mediante redundancia.
* Escalabilidad: capacidad de reducir/adicionar recursos ante la disminución/aumento de la demanda presente/futura.
* Compatibilidad: de los componentes de HW y controladores disponibles.
* Administración remota: gestión del equipo incluso cuando éste se encuentra apagado o fuera de servicio.
* Mantenimiento en caliente: para evitar la salida del servicio.

Mainframes y supercomputadoras

**Mainframe**

* Equipo de procesamiento de datos utilizado para procesos críticos a gran escala con gran **confiabilidad** y **capacidad de procesamiento**.
* Sirve para trabajos pesados.
* Recursos centralizados.
* Sirve para dar soporte en línea a miles de usuarios conectados mediante terminales remotas.
* Requiere instalaciones especiales y refrigeración adecuada.
* Ventajas:
  + Puede ser reparado sin detener los servicios.
  + Estabilidad/confiabilidad.
  + Capacidad de procesamiento y almacenamiento muy elevada.
  + Terminales de poca potencia.
  + Es poco probable que tenga Punto Único de Falla (componente aislado que si se rompe, para todo).
  + Se puede sustituir HW sin parar el resto de los componentes ni el SO.
* Desventajas:
  + Costo muy alto (CAPEX).
  + Costo de instalación elevado (OPEX).
  + Pocas empresas proveedoras.
  + Incompatibilidad con otras plataformas.
* Usos:
  + Sistemas bancarios y administrativos.
  + Grandes empresas o entes de gobierno con sistemas de operatoria crítica y continua.

**Supercomputadora**

* Sistema de procesamiento de cálculo de gran escala.
* Se usa principalmente para aplicaciones especializadas que requieren enormes cantidades de cálculos matemáticos.
* Orientada a cálculo en punto flotante (muy complejos).
* Se suelen hacer a medida.
* La mayoría usa un SO Linux.
* Ventajas:
  + Gran poder de cálculo.
  + Estabilidad/confianza.
  + Capacidad de procesamiento y almacenamiento.
  + Gran tamaño de memoria.
* Desventajas:
  + Costo (CAPEX)
  + Costo de instalación (OPEX)
  + Consumo de energía
* Usos:
  + Investigaciones científicas.
  + Diseño automotriz y aeronáutico.
  + Estudio y predicción del clima.

**Mainframe vs. Supercomputadora**

|  |  |
| --- | --- |
| Mainframe | Supercomputadora |
| Problemas que requieran extrema fiabilidad y manejo de gran número de dispositivos de E/S | Problemas que necesiten gran velocidad de cálculo |
| Decenas de procesadores | Miles de procesadores |
| Optimizados para problemas que implican grandes cantidades de datos externos | Optimizadas para cálculos complejos |
| Suelen utilizarse en empresas y entes de gobierno | Suelen utilizarse en investigación científica y desarrollos industriales |

Métricas populares de rendimiento

La gestión de MIPS sirve para reducir los costos de TI a través de mediciones automáticas del consumo de las aplicaciones y la identificación del uso abusivo y recurrente de subrutinas de sistema y fallas de código crónicas.

Servidores

* Tower server:
  + Se utilizan en pequeñas y medianas empresas.
  + No requieren de infraestructura de instalación especial más allá de la básica para una PC.
  + No soportan mecanismos de HA más allá de esquemas de RAID de discos internos.
* Rack server:
  + Servidores para instalar en centros de datos, dentro de un rack.
  + Implementa características de alta disponibilidad (HA).
  + Factor de forma: determina el espacio ocupado (ocupan menos lugar pq tienen un tamaño estandarizado).
* Blade server:
  + Consolidar recursos.
  + Reducir espacio.
  + Disminuir el consumo de energía.
  + Aumenta el consumo energético por unidad de rack.
  + Cada “librito” es un servidor blade, va uno al lado del otro.
  + No suele usarse este tipo de servers.
* Hyper Convergence Infrastructure (HCI):
  + Virtualiza mediante SW de tal manera que no sabés cuántos discos tenés sino cuánto espacio.
  + Disminuye los movimientos físicos.
  + Consolidar recursos.
  + Reducir espacio.
  + Disminuir el consumo de energía.
  + Escalabilidad ilimitada.
  + Rápido despliegue.
* Virtual server:
  + Segmentación lógica de recursos con entidad propia.
  + Tienen en mayor o menor medida abstracción del HW subyacente y de su mantenimiento.
  + Permiten rápido despliegue.
  + Los recursos disponibles pueden no ser reales y/o reservados.
  + Su existencia permite la optimización del uso de HW.
  + Facilitador del cloud, sin la existencia de los servidores virtuales no existiría cloud.
* Containers:
  + Optimización de consumo de recursos técnicos y humanos para el despliegue de aplicaciones de ciclo continuo.
  + Potencia las arquitecturas basadas en microservicios.
  + Componentes reutilizables.
  + Despliegue rápido:
    - Facilita la integración continua.
    - Facilita la administración de entornos.
    - Facilita el versionado de entornos.
  + Containers vs. VMs: 2 formas distintas de vituralización.

Virtualización

Virtualizar es que el SW simule aspectos del HW que no son realmente así. Esto sirve para optimizar recursos y permite disminuir costos.

¿Por qué es importante?

* Optimiza el uso de recursos.
* Aumenta mucho la velocidad de despliegue y redimensionamiento de recursos.
* Permite disminuir el tiempo de parada por mantenimiento de HW.
* Aumenta los niveles de disponibilidad de los servicios.
* Permite la delegación de la gestión de los recursos.
* Permite consolidar infraestructura.
* Permite mantener compatibilidad con el HW real.

La virtualización en IT se usa en infraestructuras de procesamiento, almacenamiento, redes, seguridad de la información, SW, datacenter, escritorios de trabajo, entre otros.

Aplicaciones de la vitualización en IT:

* Servidores virtuales: máquinas virtuales para aplicaciones.
* Appliances de seguridad: firewall, IDS, SIEM, etc.
* Almacenamiento distribuido (DSS) y/o remoto (NAS).
* HW de redes (stack), switches virtuales, redes virtuales (VLAN), encapsulamiento de direcciones (NAT).
* Implementación de servicios de contenedores.
* Sistemas de cluster de servicios.

Cluster de procesamiento

Un cluster es un grupo de recursos de procesamiento individuales (denominado nodo) trabajando en conjunto bajo una solución de SW y conectividad que se ponen al servicio del procesamiento de una determinada tarea.

Objetivo: agrupar subsistemas de procesamiento para aumentar la escalabilidad y/o la disponibilidad del sistema, logrando así alta confiabilidad del sistema.

Componentes:

* Nodos de procesamiento (CPU/memoria)
* Almacenamiento
* Sistemas operativos
* Conexiones de red
* Protocolos de comunicación y servicios
* SW de aplicación para su gestión

Tipos de cluster

* **Load balancing cluster (LB-C)**: provee escalabilidad basada en la distribución de la carga entre los nodos activos del sistema. El front-end del servicio balancea la carga entre los procesadores o servidores para no saturar unos y que otros estén ociosos.

Permite aumentar la capacidad del sistema incorporando nuevos nodos al cluster para incrementar la potencia total del sistema. De la misma manera, permite disminuir el tamaño del cluster para que acompañe el decrecimiento de las cargas de trabajo.

* **High performance cluster (HP-C):** explota el potencial del procesamiento en paralelo entre múltiples computadoras. Divide la carga antes de enviarla y manda distintas partes a diferentes nodos. Es el cluster más indicado para el procesamiento de funciones complejas.
* **High availability cluster (HA-C):** mediante la redundancia de nodos agrupados permite aumentar el nivel de disponibilidad del servicio que ofrecen. Ante la falla de uno de los nodos, el sistema de gestión de cluster transfiere el servicio activo a otro nodo. La HA hace que no paremos, siempre y cuando no muera todo el datacenter. El nivel de redundancia determinará la cantidad de fallas admisibles sin pérdida de servicio.
* También existen los cluster combinados, por ejemplo LB+HA.
* Grid computing: conjunto de elementos heterogéneos trabajando en conjunto con un fin específico. Es un cluster con nodos en distintos equipos distribuidos en el mundo. Factores característicos:
  + Descentralización
  + Diversidad de recursos y dinamismo
  + Administración descentralizada
* Grid computing vs. Cluster:
  + Las computadoras que forman parte pueden ejecutar diferentes SO y tener HW diferente, mientras que las computadoras de cluster tienen el mismo HW y SO.
  + Gris computing suele hacer uso de la potencia de computación residual en una computadora de escritorio conectada a una red, mientras que las máquinas en un cluster están dedicadas a trabajar como una sola unidad y nada más.

Sistemas de caché

La caché es un buffer especial de memoria que guarda info de rápido acceso que necesite un sistema particular.

Se accede por primera vez a caché para dejar una copia de un dato que el sistema predice que puede reutilizar en un futuro cercano. Los siguientes accesos se realizar para ejecutar la lectura de ese dato, haciendo que el tiempo de acceso medio al dato sea menor.

**Datacenter y Cloud**

Datacenter

* Un Centro de Cómputo, es el conjunto de recursos físicos, lógicos, y humanos necesarios para la organización, realización y control de las actividades informáticas de una empresa.
* Es la dependencia responsable de la generación de la información necesaria para una adecuada toma de decisiones en los diferentes niveles de gestión.
* Concentra los principales recursos de IT de una organización.

Misión del datacenter

Su misión es prestar servicios a diferentes áreas de una organización ya sea dentro de la misma empresa, o bien fuera de ella, tales como: producción, control de operaciones, captura de datos, programación, dibujo, biblioteca, etc. Los diversos servicios que puede prestar un centro de cómputo, pueden dividirse en departamentos a áreas específicas de trabajo.

TIA942

Es un estándar que indica el nivel de fiabilidad de un datacenter en función del nivel de disponibilidad (cuentan las paradas programadas y no programadas). Los clasifica en TIER.

* TIER I – Básico:
  + Disponibilidad del 99,671%.
  + Sensible a las interrupciones, planificadas o no.
  + Un solo paso de corriente y distribución de aire acondicionado, sin componentes redundantes.
  + Sin exigencias de piso elevado (Piso elevado: paneles que por debajo pasan los cables).
  + Generador independiente.
  + Tiempo de inactividad anual: 28,817 horas.
  + Debe cerrarse completamente para realizar mantenimiento preventivo.
* TIER II – Componentes redundantes:
  + Disponibilidad del 99,741%.
  + Menor sensibilidad a las interrupciones.
  + Un solo paso de corriente y distribución de aire acondicionado, con un componente redundante.
  + Incluye piso elevado, UPS y generador. (UPS: mantiene la energía por un tiempo. Es a batería).
  + Tiempo de inactividad anual: 22,688 horas.
  + El mantenimiento de la alimentación y otras partes de la infraestructura requieren de un cierre de procesamiento.
* TIER III – Mantenimiento concurrente:
  + Disponibilidad 99,982%.
  + Mantenimiento planificado sin interrupción de funcionamiento, pero posibilidad de problemas en las no previstas.
  + Múltiples accesos de energía y refrigeración, por un solo encaminamiento activo. Incluye componentes redundantes (N+1).
  + Tiempo de inactividad anual: 1,5768 horas.
* TIER IV – Tolerante a errores:
  + 99,995% de disponibilidad.
  + Mantenimiento planificado sin interrupción de funcionamiento de los datos críticos. Posibilidad de sostener un caso de improviso sin daños críticos.
  + Múltiples pasos de corriente y rutas de enfriamiento. Incluye componentes redundantes (2(N+1)).
  + Tiempo de inactividad anual: 26,28 minutos.

Para la redundancia muchas veces conviene contratar a distintos proveedores, ya que si a uno se le cae todo el sistema, no se me caen todas las redundancias.

Costo de caída de un datacenter

El costo de caída incluye:

* Costo de Inactividad de los empleados / Pérdida de Productividad.
* Pérdida de Operaciones: ventas, etc.
* Incumplimientos de normativa, acuerdos o SLA: en ciertos casos existen normativas penalizadoras.
* Impacto en marca, pérdida de confianza: la marca queda con una mala imagen.
* Otros costos

Causas comunes de las caídas

Las principales causas son controlables o editables.

* Fallo de las baterías del UPS (55%): es evitable. Se puede controlar el estado de las baterías.
* Apagado de emergencia accidental (EPO)/Error humano (48%).
* Se excedió la capacidad del UPS (46%): problema de dimensionamiento.
* Ataque cibernético (34%).
* Fallo del equipo de TI (33%).
* Fugas de agua (32%).
* Relacionado con el clima (30%).
* Fallo de CRAC/ relacionado con el calor (29%).
* Fallo del equipo del UPS (27%).
* Fallo de PDU/ interruptor (26%).

Cloud

Cloud computing es un modelo para permitir el acceso ubicuo, conveniente y bajo demanda a través de la red a un conjunto compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar y lanzar rápidamente con un mínimo esfuerzo de gestión o interacción del proveedor de servicios.

Difiere del datacenter en que cloud se conecta mediante internet.

En cloud es típica la autogestión.

Características esenciales

* **Autoservicio bajo demanda**: un cliente puede proporcionarse capacidades de cómputo de manera unilateral, como tiempo de servidor y almacenamiento en la red, según sea necesario de forma automática sin necesidad de interacción humana con cada proveedor de servicios. “Lo que quiero cuando quiero, meto y saco”.
* **Amplio acceso a través de la red**: las capacidades están disponibles a través de la red y se accede a ellas mediante mecanismos estandarizados que promueven el uso de plataformas heterogéneas de clientes por parte de clientes (por ejemplo, teléfonos móviles, tabletas, notebooks y puestos de trabajo).
* **Conjunto compartido de recursos**: los recursos informáticos del proveedor se combinan para brindar servicios utilizando un esquema compartido entre múltiples clientes, con diferentes recursos físicos y virtuales asignados dinámicamente y reasignados de acuerdo con la demanda de quien consume. No se tiene un servidor específico para un cliente. Cada cliente elige la cantidad de memoria a consumir pero no el servidor en el cual tenerla.
* **Rápida elasticidad**: las capacidades pueden ser aprovisionadas y desplegadas elásticamente, en algunos casos automáticamente, para escalar rápidamente aumentando y disminuyendo de acuerdo con la demanda. Para el consumidor, las capacidades disponibles para el aprovisionamiento a menudo parecen ser ilimitadas y pueden asignarse en cualquier cantidad en cualquier momento. Se sacan o agregan features casi de forma instantánea. Se puede hacer de forma automática programada (sale más caro contar con este servicio).
* **Servicio medido**: los sistemas en cloud controlan y optimizan automáticamente el uso de recursos al aprovechar una capacidad de medición en algún nivel de abstracción apropiado para el tipo de servicio (por ejemplo, almacenamiento, procesamiento, ancho de banda y cuentas de usuario activas). Se mide TODO el uso de los recursos hasta el detalle. No es fácil saber qué y cuánto contratar de cada cosa (hay consultoras para esto).

Modelos de servicio

* **Software as a Service (SaaS)**: la capacidad proporcionada al consumidor es utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura de nube. Se puede acceder a las aplicaciones desde varios dispositivos, como un navegador web (por ejemplo, correo electrónico basado en web) o una interfaz de programa. Ej: Office 365.
* **Platform as a Service (PaaS)**: la capacidad proporcionada al consumidor es implementar en la infraestructura en la nube aplicaciones creadas por el consumidor o adquiridas creadas con lenguajes de programación, bibliotecas, servicios y herramientas compatibles con el proveedor. Ej: tener una plataforma para testing.
* **Infrastructure as a Service (IaaS)**: La capacidad provista es aprovisionar procesamiento, almacenamiento, redes y otros recursos informáticos fundamentales donde el consumidor puede implementar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. Es el más básico.

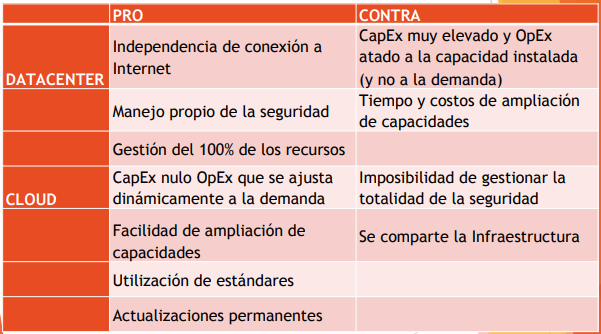
Modelos de despliegue

* **Private cloud:** la infraestructura se proporciona para uso exclusivo por una sola organización que comprende múltiples consumidores (por ejemplo, unidades de negocios). El cliente pone los “fierros”, se monta sobre su infraestructura.
* **Community cloud:** la infraestructura se proporciona para uso exclusivo de una comunidad específica de consumidores de organizaciones que tienen inquietudes compartidas (por ejemplo, misión, requisitos de seguridad, políticas y consideraciones de cumplimiento).
* **Public cloud**: la infraestructura está prevista para uso abierto por el público en general.
* **Hybrid cloud:** la infraestructura es una composición de dos o más distintas (privada, comunitaria o pública) que siguen siendo entidades únicas, pero están unidas por una tecnología patentada o estandarizada que permite la portabilidad de datos y aplicaciones (por ejemplo, la explosión de la nube para el equilibrio de carga entre nubes).

Crecimiento en IaaS, PaaS, SaaS y Serverless

* Las empresas con infraestructuras antiguas están reemplazando el hardware con IaaS.
* Los departamentos de TI de todo el mundo utilizan SaaS para proporcionar a sus empleados aplicaciones empresariales (ej: correo electrónico, almacenamiento y procesamiento de textos), así como para proporcionar aplicaciones a los clientes (ej: seguimiento de paquetes, catálogo y carrito de compras).
* La cantidad de ofertas para PaaS continúa creciendo con compañías como Amazon y Microsoft que ofrecen plataformas específicas para las aplicaciones y la infraestructura subyacente que desean los clientes.
* Amazon, Google e IBM también ofrecen opciones de serverless computing, también conocidas como cloud functions.
* Serverless permite a los desarrolladores cargar código solo para funciones individuales y hacer que se ejecuten sin preocuparse por la máquina o los problemas de carga.
* Si la informática empresarial tradicional era como comprar un automóvil y cloud computing tradicional era como alquilarla por un día, serverless es como tomar un taxi.

Planificación estratégica – Datacenter vs. Cloud

- Datacenter no necesita conexión a internet pero, si no la hay, puede que el datacenter no pueda brindarnos algunos servicios.

Migración a cloud

Cada vez más organizaciones enfrentan los desafíos de migración a la nube asociados con el mantenimiento de una infraestructura híbrida para obtener rápidamente los beneficios de la misma. Si bien hay una serie de beneficios para la migración a cloud, también hay riesgos operativos, de seguridad y financieros que deben tenerse en cuenta. Migrar a la nube no es fácil, por ej cuando hay apps conectadas entre sí, puede dificultar la migración si querés migrar una y la otra no.

Estrategias de migración a cloud

Las más relevantes son:

* Rehosting: una aplicación es re-hosteada “como está” en una plataforma cloud. Implica mover servidores físicos y virtuales a una solución IaaS que resulte compatible. Se muda a la nube y se hace todo para que CORRA en la misma, nada más. No aprovecha toda la nube sino que se mueve “lo menos posible”.
* Replatforming / Cloudification: implica mantener una aplicación hosteada on-premise pero utilizando servicios cloud para extender capacidades en reemplazo de algunos componentes on-premise. Hago que una parte de la app se haga en la nube.
* Replacement / Repurchasing: implica reemplazar una solución on-premise por una cloud evitando la re-ingeniería. Sería comprar una app nueva que esté en la nube y reemplazar la vieja.
* Multi-cloud Refactor: la arquitectura de una aplicación on-premise se reformula para proveer un mejora en la calidad de servicio. Tener más de un proveedor cloud, reemplazo parcial.
* Hybrid Refactor: la arquitectura de una aplicación on- premise se reformula para desplegarla parcialmente en un ambiente cloud y parcialmente en su plataforma on- premise actual. Una parte de la app la reescribimos y la desplegamos en la nube. Difiere con el rehosting en que esa parte que va a la nube fue pensada para tal fin.

**Bases de datos**

Tipos de bases de datos

* Relacionales (SQL): mySQL, SQL Server, Oracle, etc.
* No relacionales (NoSQL): MongoDB, Cassandra, etc. Hay NR orientadas a distintas cosas:
  + Key value: almacena datos como un conjunto de pares clave-valor.
  + Documentales: orientadas al almacenamiento y gestión de documentos.
  + Column family: apuntan a columnas organizadas por familias de columnas.
  + Graphics: se basan en un modelo de grafo flexible.
  + In-memory: se almacenan completamente en memoria. Útiles para almacenar logins.

Ventajas y desventajas de los tipos de BD

**DB Relacional**

* Ventajas:
  + Manejo de integridad referencial.
  + Seguridad desde la BD.
  + Correcta concurrencia.
  + Manejo de rollback.
* Desventajas:
  + Almacenamiento normalizado, poco flexible.
  + No optimizadas para alto rendimiento.
  + La escalabilidad tiene alto costo.
  + Tienen que pagar licencias.

**DB No relacional**

* Ventajas:
  + Almacenamiento no normalizado (más flexibilidad).
  + Muchas son open source (licencias gratuitas).
  + Permite persistencia en distintas formas.
  + Carga de información por sistema de clustering.
  + Se maneja más info en memoria, suelen ser más rápidas.
* Desventajas:
  + El control de los datos se debe hacer por aplicación.
  + Puede ser más costosa la primera implementación.
  + Puede existir más redundancia de datos.

DataWarehouse vs. BigData

|  |  |  |
| --- | --- | --- |
| **Características** | **DW** | **BD** |
| Grandes volúmenes de datos | Sí | Sí |
| Se utilizan cuando hay que manejar datos históricos | Más | Menos |
| Se utilizan para hacer datos analíticos (procesar online de distintas fuentes a la vez) | Menos | Más |
| Manejan datos normalizados | Más | Más |
| Manejan datos no normalizados | No | Sí |
| Fuente de datos | Internas a la organización | Más ext que internas |
| La implementación lleva bastante tiempo | Sí | Sí |
| Los errores a veces no se notan rápido | Sí | Sí |
| El mantenimiento requiere de tiempo | Sí | Sí |
| El costo suele ser grande | Sí | Sí |

* OLAP: agiliza la consulta de grandes cantidades de datos. Se suele usar para DW.
* Hadoop: sistema de almacenamiento distribuido. Grandes volúmenes de datos que son redundantes y se guardan en muchos lugares pareciendo un lugar grande. Se suele usar para BigData pero se puede usar tmb para DW.

Conceptos de bases de datos

* Consistencia: cada dato debe estar validado de acuerdo a las reglas definidas.
* Durabilidad: una vez que los datos están grabados deben quedar así en el tiempo.
* Concurrencia: varias transacciones puedan acceder a la misma BD a la vez.
* Persistencia: la forma en que los datos quedan grabados en la BD.
* Atomicidad: cada transacción es a todo o nada, si una parte falla, fallarán todas las operaciones.
* Aislamiento: permite que distintas transacciones se ejecuten de forma independiente.

Cuadro comparativo

|  |  |  |
| --- | --- | --- |
| **Característica** | **DB No relacional** | **DB Relacional** |
| Performance | Alta | Media |
| Confiabilidad | Media (se valida por app) | Buena |
| Disponibilidad | Buena | Buena |
| Consistencia | Pobre | Buena |
| Data storage | Optimizado para grandes vols | Tamaño medio a grande |
| Escalabilidad | Alto | Alto (pero caro) |

Teorema de CAP

* **Consistencia (C)**: todos los nodos garantizan la misma info al mismo tiempo (sea inserción, actualización, consulta). Toda transacción termina en un estado válido, no error.
* **Disponibilidad (A)**: si un nodo se ha caído o deja de emitir rta, el sistema debe seguir operando y, una vez que se reestablece, volver a sincronizar con los demás. Siempre se debe poder acceder.
* **Tolerancia de partición de red (P)**: los sistemas distribuidos pueden estar divididos en particiones (generalmente de forma geográfica) por ende esta condición implica que el sistema tiene que seguir funcionando aunque existan fallos o caídas parciales que dividan el sistema. Disponibilidad ante caída de una zona.

Escenarios de CAP: se pueden cumplir 2 de los 3 al mismo tiempo:

* **AP**: garantizan disponibilidad y tolerancia a particiones pero no la consistencia, al menos de forma total. Algunas de ellas consiguen una consistencia parcial a través de la replicación y la verificación (asociado a NR x ej Cassandra, Infinite Graph, etc).
* **CP**: garantizan consistencia y tolerancia a particiones. Para lograr la consistencia y replicar los datos a través de los nodos, sacrifican la disponibilidad (asociado a otras NR x ej MongoDB).
* **CA**: garantizan consistencia y disponibilidad, pero tienen problemas con la tolerancia a particiones. Este problema lo suelen gestionar replicando datos (asociado a R).

Persistencia políglota

Combinado de diferentes tecnologías para dar rta a las diferentes necesidades de almacenamiento. Es persistir los datos de una app en distintos tipos de BBDD en lugar de persistir todo en el mismo tipo. Va a depender del tipo de datos que se necesite almacenar. Se suele aplicar cuando tengo muchas apps.

Ventajas:

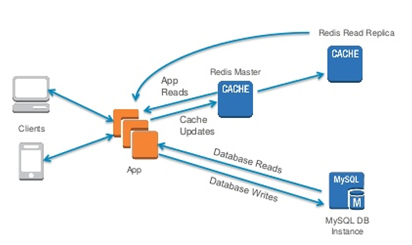
* **Mejora del rendimiento** ya que se utiliza una DB dependiendo de las necesidades funcionales que se tengan.
* **Manejo de mayor volumen de datos** debido a que las NoSQL están diseñadas para operar en cluster. **Mayor escalamiento horizontal**.
* **Consultas mucho más flexibles** porque se adaptan a la necesidad y están directamente enlazadas al formato de datos manejado por la DB.

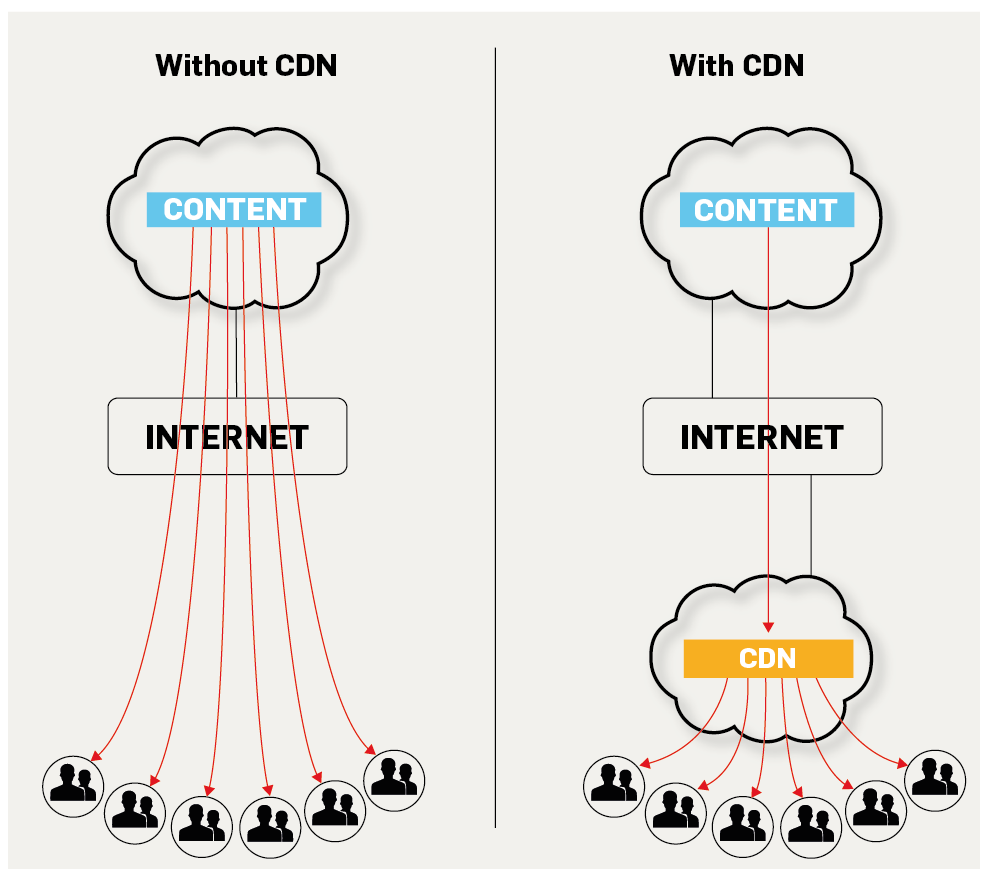
Desventajas:

* **Dificultad** para que la info sea almacenada entre distintas BBDD y que sea **consistente**.
* Hay que **decidir qué DB utilizar** entre las que hay disponibles. Esto puede traer problemas si no hay mucha experiencia.
* La **complejidad de la app aumenta** debido a que ahora hay que manejar mayor cantidad de BBDD y además que funcionen en conjunto todas ellas.
* Se adquieren **nuevas responsabilidades administrativas** porque ahora hay que estar a cargo de más BBDD y asegurar su funcionamiento.
* Si no se tiene experiencia con las BBDD a utilizar, eso **exige** **entrenamiento** el cual lleva tiempo y puede afectar al desarrollo.

**Sistema REDIS**

Almacenamiento de datos en memoria utilizando un esquema clave-valor.



**Content delivery Network (CDN)**

Sistema distribuido y escalable de entrega de contenidos basado en minimizar el costo de red entre el punto de distribución y el usuario. Ej: akamai.

(Memcache y varnish en ppt de [persistencia de datos](https://drive.google.com/file/d/1g2capyd9IjZmh3G7vm03crIzHkJq4EtB/view?usp=sharing))

**Seguridad de la información**

La información es un activo que, como otros, resulta esencial para el negocio de la organización y consecuentemente debe protegerse adecuadamente.

La seguridad de la información hace referencia a todas aquellas medidas preventivas, reactivas de las organizaciones y de los sistemas tecnológicos que permitan resguardar y proteger la información buscando mantener la **confidencialidad**, la **disponibilidad** y la **integridad** de la misma.

Eventos vs Incidentes

* **Evento de seguridad de la información:** 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 de seguridad de la información:** 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. Tiene que ser detectado lo antes posible.

Sistema de Gestión de Seguridad de la Información (SGSI)

Es una parte de la norma ISO.

La gestión de la Seguridad de la Información busca establecer y mantener programas, controles y políticas que tengan como finalidad conservar la confidencialidad, integridad y disponibilidad de la información. Es un proceso continuo. Debe comprender los siguientes elementos:

* Elementos administrativos.
* Políticas de seguridad.
* Organización y división de responsabilidades: definir quién es responsable de cada cosa.
* Seguridad lógica: este elemento indica qué tipo de seguridad se usa para cada cosa.

Principios básicos de la seguridad

* Confidencialidad: es que el mensaje que se envía sea auténtico y le llegue solamente a quien corresponde.
* Disponibilidad: que el mensaje esté disponible en el momento del envío y recepción.
* Integridad: que el mensaje que se envía no haya sido alterado.

Plan de respuesta a incidentes

* Fases:
  + Acción inmediata para detener o minimizar el incidente: que salga o se afecte la menor cantidad de info posible.
  + Investigación del incidente: no se puede detener la operatoria de la org mientras investigamos.
  + Restauración de los recursos afectados.
  + Reporte del incidente a los canales apropiados: hay que tener cuidado con quién es el responsable del incidente y a quién se le reporta.
* Componentes:
  + Equipo de expertos
  + Estrategia legal revisada y aprobada
  + Soporte financiero de la organización
  + Soporte ejecutivo de la gerencia superior de la compañía o áreas afectadas: hacerle llegar reportes a la alta gerencia.
  + Recursos físicos: usados para alcanzar la confidencialidad, disponibilidad e integridad.

Ataques de seguridad

* Atacantes: hacker, cracker, lammer, copyhacker, bucaneros, phreaker, newbie, script kiddie.
* Principales ataques a las org:
  + Propagación de malware por e-mail y spam
  + Botnets
  + Ataques de phishing alojados en sitios web
  + Ataques XML sobre arquitectura orientada a servicios (SOA) y Web Services

Seguridad lógica

* Restringir el acceso a los programas y archivos.
* Asegurar que los usuarios puedan trabajar sin supervisión minuciosa sin afectar ningún dato, programa ni archivo que no deban.
* Asegurar que se están utilizando los datos, archivos y programas correctos en cada situación.
* Que la información transmitida sea recibida sólo por el destinatario deseado.
* Que la información recibida sea la misma que ha sido enviada.
* Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos. Ej: cuando hacés una transferencia y te mandan el código por SMS.

Dentro de la seguridad lógica tenemos servicios de seguridad como el **no repudio**. Es validado por la firma digital.

* No repudio de origen: Prueba que el mensaje fue enviado por la parte específica.
* No repudio de destino: Prueba que el mensaje fue recibido por la parte específica.

**Tipos de usuario**:

* Propietario
* Administrador
* Usuario principal o referente (Key user)
* Usuario de explotación
* Usuario de auditoría

**Sistemas de cifrado**:

* Cifrado simétrico: se usa la misma clave para cifrar y descifrar el mensaje.
* Cifrado asimétrico: se usa una clave pública y una privada en el cifrado y descifrado. Tanto el emisor como el receptor tienen una clave de cada tipo.
* Cifrado híbrido: es una combinación de los dos anteriores. Le aporta más seguridad al canal.

Seguridad física

* Administración de respaldos de información (Backups).
* Alta disponibilidad.
* Gestión de centros de cómputos principales y secundarios.

ISO/IEC 27000

* Marco de gestión de la seguridad de la información utilizable por cualquier tipo de organización.
* ISO/IEC 27001: Norma principal de la serie,  contiene los requisitos del sistema de gestión de seguridad de la información y es la norma que se certifica por auditores externos.
* ISO/IEC 27002: Es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. No es certificable.

COBIT 5 (for Information Security)

* Complemento de procesos COBIT 5 (Gestión de la seguridad, Gestión de la continuidad y Gestión de Servicios de seguridad).
* Seguridad como disciplina transversal con un enfoque holístico.
* Alinear la seguridad de la información con los objetivos de la empresa.
* Mapeo de procesos con cláusulas y controles de otros estándares como ISO 27001 o NIST.

ITIL - Gestión de la Seguridad de la Información

Se compone de los 3 pilares de la seguridad de la información (I+D+C) + Legalidad.

Legalidad: Asegura que la información que estamos utilizando cumple los requisitos legales del ámbito que le atañe y en el que está envuelta.

Fases:

* Planificación:
  + Gestión de los niveles de servicio (SLA)
  + Gestión de las incidencias
  + Centro de servicios (service desk)
* Ejecución:
  + Gestión de los problemas
  + Gestión de la capacidad
  + Gestión de cambios
  + Gestión de versiones
  + Gestión de la continuidad
* Seguimiento:
  + Centro de servicios (service desk)
  + Gestión de la disponibilidad

ONTI - Oficina Nacional de Tecnología Informática

Definición de seguridad de información según ONTI: La información es un activo que, como otros activos importantes, es esencial y en consecuencia necesita ser protegido adecuadamente. Puede estar impresa o escrita en un papel, almacenada electrónicamente, transmitida por correo o utilizando medios electrónicos. La seguridad de la información es la protección de la información de un rango amplio de amenazas para poder asegurar la continuidad del negocio, minimizar el riesgo de la operación y la operación normal del organismo.

Se compone de los 3 pilares de la seguridad de la información (I+D+C) + No repudio + Autenticidad + Auditabilidad + Legalidad + Confiabilidad + Protección a la duplicación.

Firma digital

Es una herramienta tecnológica que permite garantizar la autoría e integridad de los documentos digitales. La firma digital es un instrumento con características técnicas y normativas. Esto significa que existen procedimientos técnicos que permiten la creación y verificación de firmas digitales, y existen documentos normativos que respaldan el valor legal que dichas firmas poseen.

Si un documento tiene firma digital significa que ese documento lo creó esa persona firmante. Solo se aplica en documentos.

**Firma digital vs firma electrónica**

La firma digital se registra legalmente y se establece mediante un procedimiento matemático con info que conoce exclusivamente el firmante.

La firma electrónica se compone de datos electrónicos ligados entre sí para identificar al firmante, pero carece de ciertos requisitos para ser considerada firma digital. Se puede usar para documentos u otros tipos de información. No se registra de forma legal. No establece el no repudio.

Dirección Nacional de Protección de Datos Personales

Incluye un centro de asistencia a las víctimas de robo de identidad, un registro nacional de bases de datos y una consulta de documentos de identidad cuestionados, entre otras cosas.

**Almacenamiento y disponibilidad de datos**

Se almacenan datos porque no puedo tener todo “en la mano”, no me entra.

Evolución de los sistemas de almacenamiento corporativos

* **Sistemas y almacenamiento centralizado**: un sistema de almacenamiento es un SW que maneja muchos dispositivos de almacenamiento (ej: mainframes).
  + Se accede mediante terminales de un sistema central.
* Cajas de discos conectados mediante controladoras de acceso directo (**DAS**): discos conectados a un equipo que los administra.
  + Accesible mediante controladoras e interfaces conectadas en forma directa a los servidores implementando protocolos de comunicación específicos (iSCSI, SAS, FC).
  + El Servidor puede tener una o más tarjetas (NIC/SAS/HBA Adapter) y cada una puede tener uno o más puertos.
  + El Storage puede tener una o más tarjetas “Host Adapter”. Cada una puede tener uno o más puertos.
  + Se trabaja a nivel de bloque.
* Cajas de discos accedidas mediante redes de almacenamiento (**SAN**): red de discos. Los discos se conectan a una red.
  + No va por red IP.
  + Accesible mediante controladoras e interfaces conectadas a través de una red de almacenamiento con protocolos de comunicación específicos (FC, FICON).
  + El Servidor puede tener una o más tarjetas Host Bus Adapter. Cada una puede tener uno o más puertos.
  + El Storage puede tener una o más tarjetas “Host Adapter”. Cada una puede tener uno o más puertos.
  + Alto costo, rendimiento alto y muy eficiente. Escalable.
* Cajas de discos accedidas mediante redes de datos IP (**NAS**): sistema de almacenamiento conectado a la red.
  + Permite llegar lejos pero la tx es más lenta.
  + Consiste en almacenamiento accesible por medio de redes IP de transferencia de datos.
  + Los servicios son expuestos bajo protocolos NFS, CIFS, HTTP, FTP.
  + Se trabaja a nivel de archivo.
  + Bajo costo pero rendimiento bajo y poco eficiente.
* Servicios de almacenamiento en la **nube**.

Disponibilidad (Availability)

Es la posibilidad de poder continuar dando servicios de storages en el evento de una falla de hardware o software. En este caso refiere a la disponibilidad de los equipos y no de la info en sí.

Por lo pronto lo aseguramos con redundancia. Ejemplos:

* Controladoras Redundantes
* Fuentes de energía y ventiladores Redundantes
* Switches SAN redundantes
* Tarjetas con dos puertos o dos tarjetas
* Protección RAID
* Discos de repuesto (spare)

Sistema de alta disponibilidad (HA): es un SW que es como un gran hermano que se ocupa de que cada vez que llega info, haya 2 copias por si pasa algo.

Rendimiento (Performance)

Es la métrica usada para definir la velocidad de un sistema de almacenamiento. Existen 3 métodos de medición:

* Input/Outputs per second (IOPS) – Bases de datos.
* Throughput per second (MB/sec) – Streaming media.
* Response Time – el tiempo que tarda en responder el almacenamiento un pedido de la aplicación (se mide en milisegundos [ms]).

Soluciones de resguardo (Backups)

Sirven mucho para casos en donde hay un error humano y la HA lo replica en todos lados (pq la HA pisa sin preguntar ni avisar, a diferencia del backup). También sirve para cuando hay un error desde hace bastante tiempo y se necesita restaurar algo muy anterior.

Hay opciones online y offline. La opción se elegirá en base a factores:

* Disponibilidad para el uso.
* Costos de inversión (CAPEX) y de explotación (OPEX).

Soluciones Offline

**Cinta magnética**

Es barata. Lee y escribe de corrido, no podemos ir a buscar algo a un lugar específico. Tipos:

* Tape drives manuales: 1 cabezal + 1 cinta + operación manual.
* Tape drives semi-automáticos (autoloader): 1 cabezal + 8 o 9 cintas + operación automática.
* Tape drives automáticos (librerías): n cabezales + m slots para cintas + operación automática.
* Tape drives virtuales (librerías virtuales VTL): bkp a disco que emula LTO + emula n cabezales + emula m slots + operación automática + mejora tiempos de bkp y restore + reduce espacio con técnicas de deduplicación/compresión.

**RAID (Arquitecturas de discos)**

Factores involucrados en la selección de una arquitectura de discos:

* Costos: $/Byte
* Performance: Velocidad de acceso R y W
* Confiabilidad: Disponibilidad de acceso a los datos

Las arquitecturas de hardware de almacenamiento toman en cuenta estos factores y clasifican a las diversas soluciones dentro de esquemas de RAID (Redundant Array of Independent Disks).

En RAID hay virtualización pq te muestra un solo disco pero en realidad usa varios.

* 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 pq no redunda la info, por lo cual podría no ser considerada estrictamente como RAID.
  + Cantidad mínima de discos: 2.
* RAID 1: Se basa en la utilización de discos adicionales sobre los que se realiza una copia bloque a bloque. Se crean discos espejo, con la exacta misma info.
  + Ventaja: provee de tolerancia a fallos con alta disponibilidad. Lectura rápida.
  + Desventaja: Muy costoso (Proporcional al nivel de redundancia).
  + Cantidad mínima de discos: 2 unidades.
* RAID 3: Dedica un disco al almacenamiento de información de paridad. La info de paridad me sirve para regenerar la info del disco que se caiga. Si se caen todos los de la paridad no sirve (ej, el 1 y el 2 y yo tengo una paridad 1,2).
  + Ventaja: altas tasas de transferencia y fiabilidad.
  + Desventaja: Rendimiento de transacción pobre.
  + Cantidad mínima de discos: 3 unidades.
* RAID 5: Utiliza bloques distribuidos para el almacenamiento de información de paridad que le permite ofrecer tolerancia a fallas. No guarda toda la info de paridad en el mismo disco.
  + 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 de discos: 3 unidades.
* RAID 6: Implementa doble paridad usando dos discos. Soluciona el cuello de botella del RAID 3 con las escrituras simultáneas.
  + Ventaja: A diferencia del RAID 5, permite la falla de dos discos de manera simultánea pq tiene doble info de paridad, en distintos discos.
  + Desventaja: Costo de las controladoras. Aumenta la relación $/Byte por doble paridad.
  + Cantidad de discos mínima: 4.
* RAID 0+1: ganas en velocidad. Si se cae un disco se anula toda la rama del RAID 1 (rama superior). Soporta la caída de 1 o los 2 discos de la misma rama.
* RAID 1+0: soporta la caída de 1 solo disco de cada rama como máximo.
* RAID 50: divide la info y en cada rama del 0 guarda la info de paridad.
* Disco de repuesto (spare): disco de repuesto para cualquier RAID que soporte tolerancia a fallos (todos menos el 0).

Plan de contingencia y recuperación ante desastres

Existen una variedad de riesgos que pueden afectar significativamente las operaciones de IT:

* Desastres naturales
* Pérdida del suministro eléctrico
* Fallas de hardware/software
* Errores humanos

Un Plan de Contingencia incluye las medidas técnicas, humanas y organizativas necesarias para garantizar la continuidad del negocio y las operaciones de una compañía ante un desastre. Actúa cuando falla hasta la HA inclusive.

**Recovery Time Objective (RTO)**: Tiempo que pasará una infraestructura antes de estar disponible (tiempo que tardamos en levantarnos desde que empezamos). Para reducir el RTO, se requiere que la Infraestructura (Servidores, Redes, Almacenamiento, Base de Datos, Aplicaciones, etc) esté disponible en el menor tiempo posible pasado el evento de interrupción.

Tecnologías de recuperación RTO:

* Clustering
* Migración manual: se pierde tiempo verificando la consistencia de datos.
* Tape restore

Los tiempos de recupero incluyen:

* Detección de fallas
* Recuperar el acceso a las redes
* Recuperar los datos
* Restaurar las aplicaciones

**Recovery Point Objective (RPO)**: es a qué momento volvés cuando vuelvas. Representa cuantos datos está dispuesta a perder la organización ante la ocurrencia de un incidente. Para reducir un RPO es necesario aumentar el sincronismo de réplica de datos.

Tecnologías de replicación RPO:

* Replicación sincrónica: inmediata.
* Replicación asincrónica: se va acumulando y va replicando la info.
* Replicación periódica: cada cierto tiempo.
* Tape backup: bkp guardados.

**Estrategias de protección de datos**

* Copias de resguardo en discos locales y externos:
  + Ventajas:
* Rápido acceso.
* Integración con aplicaciones y bases de datos.
* Desventajas:
  + Costo alto
  + No es transportable a otro datacenter.
  + Si la falla se produce en el datacenter no puedo recuperar el dato.
* Copias de resguardo periódicas en cinta, sin y con almacenamiento de manera externa en discos locales y externos:
  + Ventajas:
    - Costo bajo.
    - Transportable a otro datacenter.
  + Desventajas:
    - Requieren mayor tiempo de recupero.
* Replicación de datos en sitio externo:
  + Ventajas:
    - Permite tener un resguardo de los datos fuera del datacenter principal.
  + Desventajas:
    - Implica un costo en licencias de replicación.
    - No me permite continuar la operación ante una contingencia en el datacenter principal.
* Replicación de datos en centro de datos externo implementado como sitio de contingencia. (Para garantizar continuidad de negocio offsite):
  + Ventajas:
    - Ante una contingencia en el datacenter principal se puede continuar la operación en el datacenter de contingencia.
    - Permite volver a operar rápidamente (depende del RTO) y de forma más sencilla (comparado con backup y restore de cintas).
  + Desventajas:
    - Representa un costo alto dado que se debe duplicar la infraestructura necesaria para operar.
    - Implica un costo en licencias de replicación.
* Tener 2 datacenter más chicos y dividir las apps entre los 2.
  + Ventajas:
    - Menor costo que tener 2 idénticos.
    - Si se cae uno, solo tengo que levantar la mitad de las cosas en el otro.
  + Desventajas:
    - Tengo que levantar cosas en el que no se cae, no tengo todo ya disponible, eso implica RTO.

**Medidas para la recuperación ante desastres**

* Medidas preventivas: Acciones para evitar la ocurrencia de eventos no deseados.
* Medidas de detección: Controles para la detección de eventos no deseados.
* Medidas correctivas: Acciones para recuperar la operatoria de los sistemas.