

# Conceptos de IT

## INFRAESTRUCTURA IT

Conjunto de HW, SW, redes e instalaciones que se requieren para desarrollar, probar, entregar, monitorear, administrar y dar soporte a los servicios de IT.

## ON\_PREMISE

Instalación llevada a cabo dentro de la infraestructura de la empresa.

- La propia empresa es responsable de la seguridad, disponibilidad y gestión del SW.

## CLOUD COMPUTING

Infraestructura IT externa a la empresa, contratada bajo demanda.

## TCO (TOTAL COST OF OWNERSHIP • COSTO TOTAL DE PROPIEDAD)

Costo total de un producto a lo largo de su ciclo de vida completo, considerando todos los costos directos, indirectos y recurrentes.

## CapEx

Gastos de capital → gastos e inversiones asociados con los bienes.

- *Ejemplo: compra de un automóvil.*

## OpEx

Gastos operativos → costos relacionados con las operaciones y servicios.

- *Ejemplo: compra del combustible para dicho automóvil.*

## VIRTUALIZACIÓN

El hecho de imitar características de HW, vía SW, para crear un sistema informático virtual.

- *Ejemplo: varios discos físicos que, mediante SW, se muestran como uno solo.*

## HIPERCONVERGENCIA

Infraestructura definida por SW que virtualiza todos los elementos de los sistemas convencionales por HW.

## CLUSTER

Virtualización en la que se administran recursos de función similar.

## BALANCEO DE CARGA • LOAD BALANCING

Técnica en la que las solicitudes de Internet se distribuyen entre un cluster de servidores.

## CACHÉ

Copia de alcance rápido de un determinado recurso.

- Hay mecanismos que permiten mantener actualizada la copia respecto del original.

## DEMANDA

Lo que se nos pide.

- La comprensión de los modelos y patrones de capacidad y rendimiento ayuda a predecir la demanda y a hacer frente a incidentes.

## CAPACIDAD

Rendimiento máximo que puede ofrecer un servicio de IT o un componente del mismo en un marco temporal.

## PERFORMANCE

- Rendimiento que se obtiene de un servicio/componente del mismo en un marco temporal.
- Parte de la capacidad que se puede utilizar en un instante dado.

## SLA (SERVICE LEVEL AGREEMENT · ACUERDO DE NIVEL DE SERVICIO)

- Contrato documentado entre un proveedor y el cliente que identifica tanto los requerimientos como el nivel esperado de servicio.
- Lo que dice dicho contrato (*ejemplo: la disponibilidad medida en X horas al año*).

## ALTA DISPONIBILIDAD

Tener un porcentaje de disponibilidad<sup>1</sup> muy alto, cercano al 100%.

- Todo sistema debe tener establecido un SLA que defina cuánto tiempo y en qué horarios los recursos deben estar disponibles.
- La idea es que las interrupciones de servicio, en caso de existir, sean mínimas.

## TOLERANCIA A FALLAS

Capacidad del servicio para seguir funcionando y sin interrumpirse ante la falla o rotura de algún componente.

- La idea es que no haya interrupciones del servicio.

## REDUNDANCIA

Refiere a la (al menos) duplicación de componentes que realizan un trabajo crítico y cuya caída provocaría una interrupción del sistema.

## UPS

Dispositivo que permite mantener el suministro de energía eléctrica por un tiempo limitado a todos los dispositivos que tenga conectados independientemente de la continuidad de la tensión de la red eléctrica.

- La UPS complementa al grupo electrógeno → la UPS da un tiempo de gracia para bajar correctamente los equipos y para arrancar el grupo electrógeno, el cual será fuente de electricidad por un período de tiempo mayor.

---

<sup>1</sup> Disponibilidad → grado en que los recursos están disponibles para su uso (sin interrupciones) por el usuario final a lo largo de cierto período. Es una variable con la que se puede medir el nivel de servicio.

**RTO (RECOVERY TIME OBJECTIVE · TIEMPO OBJETIVO DE REPARACIÓN)**

- Tiempo que uno se compromete a cumplir para recuperarse de un incidente de gravedad.
- Por lo general, el RTO se especifica en un SLA.

**RPO (RECOVERY POINT OBJECTIVE · PUNTO OBJETIVO DE RECUPERACIÓN)**

- Tiempo que transcurre entre el instante del desastre y el último punto de restauración de los datos.
- Cantidad de datos que la organización va a perder en caso de que se produzca un incidente de gravedad.
- Por lo general, el RTO se especifica en un SLA.

**PLAN DE CONTINGENCIA**

Determinación precisa del *quién, qué, cómo, cuándo y dónde* realizar acciones en caso de producirse una anomalía en el sistema de información.

**PLAN DE PREVENCIÓN**

Contempla las contramedidas preventivas antes de que se materialice una amenaza.

**PLAN DE EMERGENCIA**

Contempla las contramedidas necesarias durante la materialización de una amenaza.

**PLAN DE RECUPERACIÓN**

Contempla las medidas necesarias después de materializada y controlada la amenaza.

*Hay mejores y peores marcos de trabajo.*

*No existe la mejor solución → cada una tiene sus debilidades y fortalezas.*

*A veces se evalúa muy ligeramente la calidad de una solución por la tecnología o la complejidad de la respuesta que damos.*

- *Si la necesidad a cubrir es básica, una solución básica puede ser la mejor solución de todas.*
- *A veces, un desarrollo para una solución puntual no suma mucho más que una planilla de Excel adecuadamente concebida y trabajada. Además, cuesta mucho más.*

*Es importante plantear cuál es la necesidad a cubrir...*

*Es muy común que no nos digan lo que necesitan, sino lo que quieren → piden algo que, según ellos, va a satisfacer su necesidad, pero si uno escarba un poco más, puede ser que se confirme esa presunción inicial o no: puede ser que la solución para satisfacer esas necesidades sea otra.*

*En IT, darle al cliente lo que quiere/pide no siempre es lo mejor.  
Lo mejor es darle lo que necesita, más allá de si es o no lo que quiere/pide.*

# Proyectos

## PROYECTO

Esfuerzo temporal comprometido con la creación de un producto/servicio de resultado único.

## Control, Seguimiento, Planificación, Alcance y Complejidad de Proyectos

### Características

Los proyectos son temporales → tienen un principio y un fin definidos.

- El proyecto termina cuando: (1) ha alcanzado sus objetivos; o bien (2) resulta claro que no pueden alcanzarse; o bien (3) la necesidad del proyecto ya no existe.
- Las actividades de mantenimiento realizadas una vez entregado el producto/servicio pueden ser tareas pequeñas o grandes, estas últimas son a su vez otro proyecto.

Los proyectos son de elaboración progresiva → avanzan en pasos e incrementos continuos.

### Fases

1. **Pre-Proyecto** → se revisa la idea para proveer la información necesaria para decidir si se comienza el proyecto, determinando factibilidades técnicas y económicas.
2. **Inicio** → se trata de que todos los involucrados entiendan qué producirá el proyecto, cuándo, con qué costos, con qué calidad, elaborando un plan que permita realizarlo.
3. **Ejecución** → se realiza el trabajo per se, se controla el flujo de trabajo de los equipos, se gestionan riesgos y problemas y se informan los avances del proyecto.
4. **Cierre** → se comprueba que todo se haya realizado y se informa cómo finalizó el proyecto.

### Metodologías y Estándares

En una cadena de trabajo, donde la información viaja entre distintos actores, hay un riesgo serio de que la información sufra distorsiones, que al final del camino generen un desvío extraordinario.

Algunas fallas que se pueden evitar o disminuir son:

- 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.
- Alcance del proyecto poco definido y falta de gestión de los cambios al mismo.

Tales fallas se pueden tratar mediante distintas metodologías de gestión (estándares), como la PMI (de EEUU) o la PRINCE2 (de Europa; más adecuada para proyectos grandes y complejos).

## Ciclo PDCA (Plan-Do-Check-Act)

- Plan → identificar y analizar el problema.
- Do → elaborar e implementar una solución.
- Check → evaluar los resultados.
- Act → aplicar acciones correctivas si es necesario. Finalmente, si el resultado fue exitoso, estandarizar la solución y capitalizarla en nuevas oportunidades.

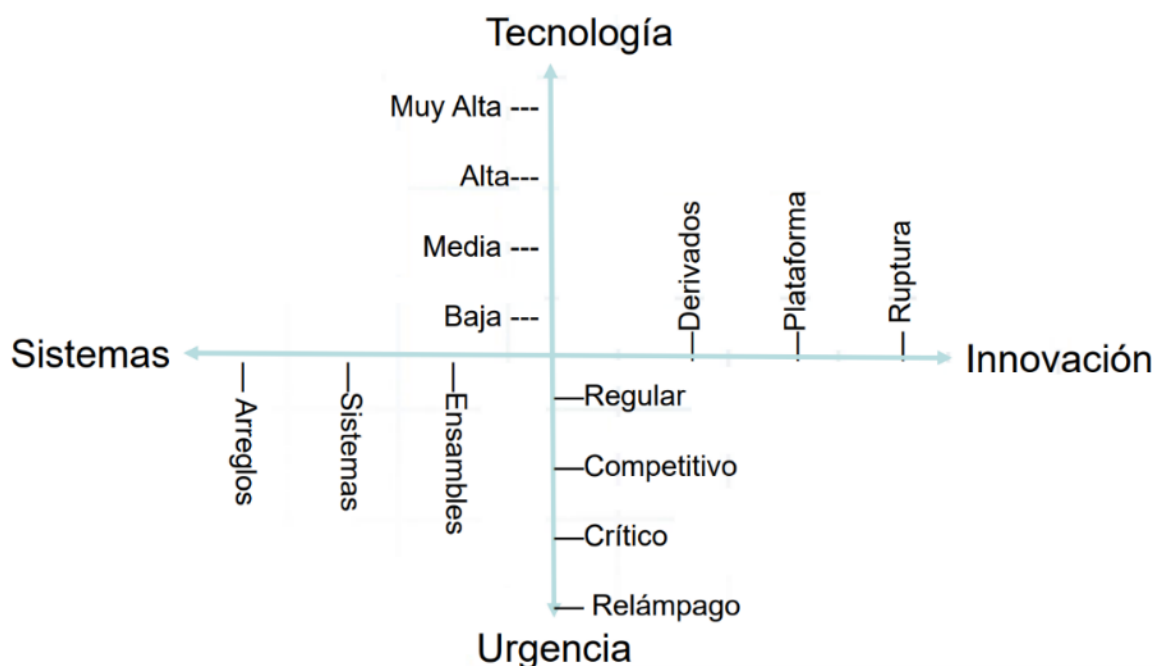
## Seguimiento y Control

- “No se puede controlar lo que no se puede medir”.
- Un proyecto es exitoso cuando contribuye al éxito de la organización.
- Es necesario realizar mediciones para determinar su grado de avance: comparando el grado de avance medido con el avance esperado, verificaremos si se observan desvíos significativos respecto de lo planificado (si ése resultara el caso, debemos tomar decisiones para corregir esa situación).

## Éxito y Complejidad – Enfoque de Diamante

Para evaluar la complejidad de los proyectos se definen 4 dimensiones, donde cada uno se representa en un eje:

- Innovación → mide cuán nuevo es el producto/servicio para los clientes, para los usuarios o para el mercado en general.
- Tecnología → mide la cantidad requerida de nuevas tecnologías.
- Sistemas Involucrados → mide la complejidad del producto/servicio, la tarea y la organización del proyecto.
- Urgencia → mide cuánto tiempo hay para completar el trabajo.



## Alcance

### ALCANCE

Definición exacta y unívoca de todo lo que está (y todo lo que no está) comprendido dentro del proyecto a ejecutar, proporcionando un entendimiento común entre los interesados en el mismo.

- Alcance del Producto → funciones y características que describen un producto o servicio.
- Alcance del Proyecto → trabajo que debe realizarse para entregar un producto de acuerdo con el alcance del producto.

## Requerimientos

### REQUERIMIENTO

Característica que un proyecto debe satisfacer.

#### Características<sup>2</sup>:

- Necesario → es necesario si su omisión provoca una deficiencia en el producto.
- Conciso → fácil de leer y entender, de redacción simple y clara.
- Completo → proporciona la información suficiente para ser comprendido.
- Consistente → no es contradictorio con otro requerimiento.
- NO Ambiguo → tiene una sola interpretación, sin causar confusiones.
- Verificable → puede ser cuantificado a través de inspección, pruebas y/o análisis.
  - Es fundamental para saber si el requerimiento en cuestión se cumple o no

#### Tipos de Requerimientos:

- Requerimientos Funcionales → describen qué es lo que el sistema debe hacer, estableciendo las funciones que el producto SW debe incluir.
- Requerimientos NO Funcionales → restricciones a las que está sometido el producto de SW a desarrollar y que influyen sobre el funcionamiento o sobre el proceso de desarrollo de SW.

## EDT (WBS) · Estructura de Desglose de Trabajo

### ESTRUCTURA DE DESGLOSE DE TRABAJO

Forma de planificación que consiste en la descomposición jerárquica del trabajo, con el objetivo de organizar y definir el alcance total del proyecto.

---

<sup>2</sup> De acuerdo a la norma IEEE-830.

## Planificación de un Proyecto – Pasos:

1. Definir actividades → identificar las acciones específicas a realizar.
2. Secuenciar las actividades → definir su orden.
3. Estimar recursos de las actividades.
4. Estimar la duración de las actividades.
5. Desarrollar un cronograma de las actividades:
  - Diagrama de Gantt → representación gráfica del cronograma.
  - Enfoques del avance del proyecto:
    - % Trabajo Completado → avance asociado al tiempo incurrido.
    - % Físico Completado → avance asociado al trabajo real ejecutado.
  - Tras planificar, se debe definir una **línea de base del proyecto** como una fotografía del cronograma para que, mientras el proyecto se vaya ejecutando, se pueda comparar la situación actual con la fotografía inicial.

## Planificación Tradicional vs Planificación Ágil

TRADICIONAL		ÁGIL
Cumplimiento de la ejecución de las tareas.	<b>Foco</b>	Esfuerzo de planificación y ejecución en los objetivos de corto plazo.
A largo plazo (pasos más largos). Puede haber entregas intermedias.	<b>Planificación</b>	A corto plazo (pasos más cortos). Hay demostraciones al cliente de los incrementos del producto/servicio.
Difíciles de realizar.	<b>Cambios en la Planificación</b>	Fáciles de realizar.
Menos frecuente. Única, al final del proyecto.	<b>Retroalimentación y Retrospectivas</b>	Más frecuente. Durante todo el proyecto.
Predictivo.	<b>Control sobre el producto/servicio</b>	Empírico, se puede probar.

Al planificar usando la metodología ágil, se asume un horizonte de incertidumbre a partir del cual no tiene sentido planificar tareas detalladas:

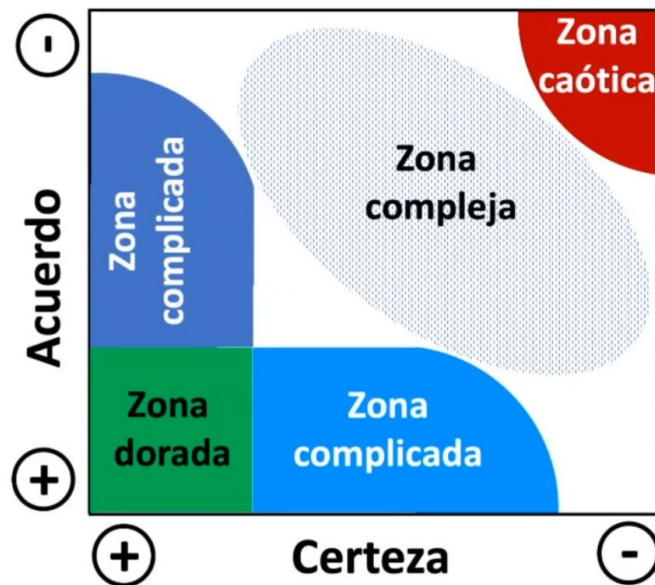
- Nivel Estratégico → planificación de los grandes objetivos del producto.
- Nivel Táctico → planificación referida a las tareas en curso.
- Nivel Operativo → replanificaciones diarias.

## Complejidad de Proyectos

Un proyecto puede ser complejo por aspectos tecnológicos, por falta de recursos, por razones de tiempo, etcétera.

El diagrama de Stacey es un enfoque de alto nivel que permite analizar la complejidad de los proyectos planteando 2 dimensiones:

- Certeza → grado de conocimiento sobre lo que se debe hacer.
- Acuerdo → grado de acuerdo político entre las partes sobre cómo hay que resolver aquello que se debe hacer.



Con tales dimensiones (certeza y acuerdo), aparecen 4 zonas posibles:

- Zona Dorada → zona ideal, donde uno quiere estar siempre.
  - Es recomendable trabajar con una metodología tradicional.
- Zonas Complicadas → situaciones en las que: (1) hay acuerdo sobre qué hacer pero se sabe menos sobre qué hacer, o bien, (2) se sabe qué hacer pero no hay acuerdo sobre cómo hacerlo.
  - Para mejorar esta situación, se puede elegir entre un enfoque ágil o uno tradicional.
- Zona Compleja → hay pocas certezas y acuerdos.
  - La solución es vía metodología ágil: aplicar pasos cortos y recalcular.
- Zona Caótica → no hay certezas ni acuerdos → no se puede hacer nada.



# Riesgos

## RIESGO

Evento posible que, si sucede, tiene un efecto sobre los objetivos del proyecto.

- **Riesgos conocidos** → aquellos que se identifican y analizan, para los cuales podemos planificar respuestas.
- **Riesgos desconocidos** → aquellos que no se pueden gestionar de manera proactiva, para los cuales debemos crear un plan de contingencia.

- Están presentes en todos los proyectos.
- Pueden ser positivos (conocidos como oportunidades) o negativos.

## Atributos de los Riesgos

- **Probabilidad de Ocurrencia** → posibilidad de que el riesgo se materialice.
- **Impacto** → resultado de la materialización del riesgo.
- **Severidad** → producto de la probabilidad por el impacto.
  - Determinada la severidad, se debe priorizar qué riesgos gestionar primero.

## Gestión de los Riesgos

1. **Identificación** → se reconocen las fuentes de riesgo y sus consecuencias potenciales.
2. **Análisis** → se determinan la necesidad de tratamiento del riesgo y la prioridad de su implementación.
3. **Tratamiento o Respuesta** → se selecciona una opción para actuar:
  - **Evitar** → la amenaza se elimina por completo, asegurando que no podrá ocurrir o que no tendrá efecto en el proyecto.
  - **Transferir** → se traslada el impacto negativo de una amenaza a un tercero, confiriéndole la responsabilidad de su gestión.
  - **Mitigar** → se reduce la probabilidad de ocurrencia y/o el impacto a un umbral aceptable.
  - **Aceptar o Asumir** → se asume que el riesgo se manifestará y se debe estar en constante monitoreo de la situación, aunque sin accionar.
4. **Monitoreo y Revisión** → se evalúa el progreso en la implementación del tratamiento. En esta instancia pueden aparecer nuevos riesgos.

## PROBLEMA

Evento, esperado o no, que afecta negativamente los objetivos de un proyecto.

- Un riesgo es algo posible → un problema es un hecho que requiere acción.
- Un problema esperado puede ser un riesgo que se materializa.

# Continuidad en las Operaciones de Negocio

## DESASTRE

Incidente o evento dramático mayúsculo que debe ser definido.

## CONTINUIDAD EN LAS OPERACIONES DE NEGOCIO

Forma de hacer gestión de riesgos donde uno se prepara para situaciones que no sabemos si van a ocurrir, pero, si ocurren, queremos que todo siga funcionando tras recuperarse lo más rápido posible.

Hay que definir qué riesgos hay y sobre qué riesgos debemos actuar → para medir la severidad se consideran su probabilidad de ocurrencia y su impacto.

Los riesgos que pueden afectar significativamente las operaciones de IT son:

- Errores humanos.
- Fallas de HW y SW.
- Pérdida del suministro eléctrico.
- Desastres naturales.

Ante desastres, debemos considerar:

- Planes de Contingencia → incluye todas las medidas técnicas, humanas y organizativas necesarias para garantizar la continuidad del negocio y las operaciones de una compañía ante un desastre.
- Planes de Recuperación.

## Métricas para Capacidad de Recuperación: RTO y RPO

### RTO (RECOVERY TIME OBJECTIVE)

Tiempo máximo tolerado desde que ocurrió el desastre hasta que el servicio se recupera y se vuelve a operar con normalidad.

- Si lo más crítico es perder tiempo, el **RTO** debe ser pequeño.
- Soluciones (ordenadas de menor a mayor pérdida de tiempo):
  - Esquema *clusterizado*.
  - Migración manual de datos.
  - Restauración con cintas magnéticas.

### RPO (RECOVERY TIME OBJECTIVE)

Cantidad de datos que se pueden perder antes de que ocurra el desastre, desde la última copia de seguridad hasta el instante del desastre.

- Si lo más crítico es perder datos, el **RPO** debe ser pequeño.
- Soluciones (ordenadas de menor a mayor pérdida de datos):
  - Replicación sincrónica.
  - Replicación asincrónica.
  - Replicación periódica.
  - Resguardo con cintas magnéticas.

## Gestión de la Continuidad del Negocio (BCM)

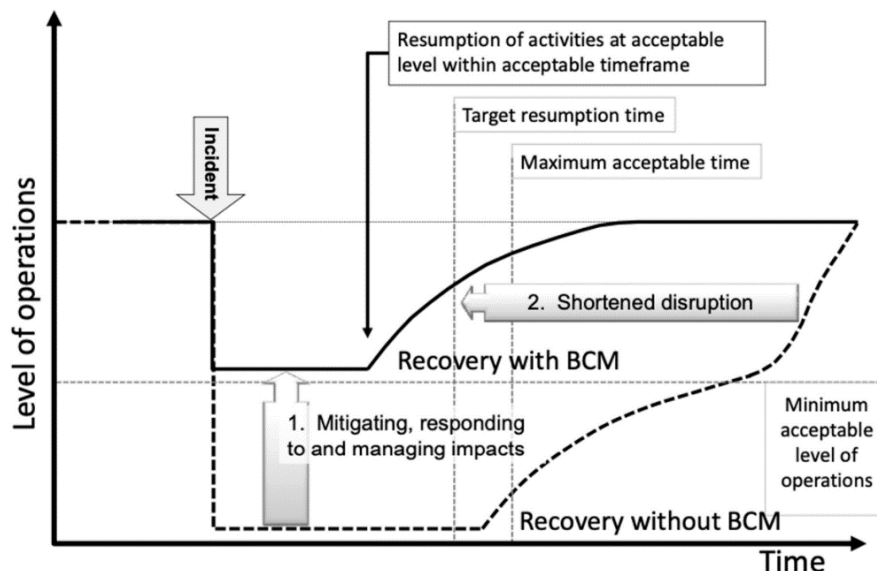
La gestión de la continuidad del negocio es una parte integral de un proceso de gestión de riesgos que salvaguarda los intereses de las partes interesadas clave, la reputación, la marca y las actividades de creación de valor de una organización. El impacto es mucho menor con **BCM**.

Esto se hace:

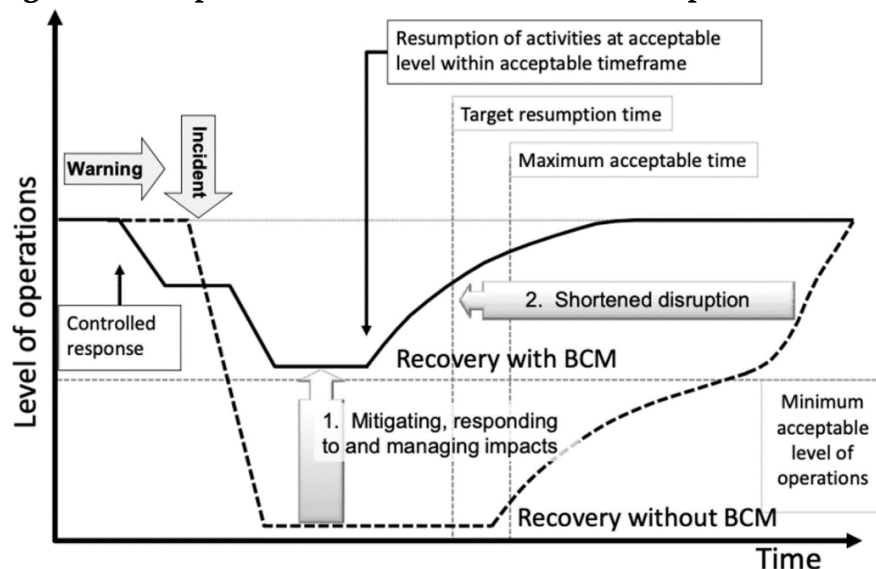
- Mediante identificación de amenazas potenciales que pueden causar impactos adversos en las operaciones del negocio de una organización y los riesgos asociados.
- Proporcionando un marco para desarrollar resiliencia para las operaciones del negocio.
- Proporcionando capacidades, instalaciones, procesos y listas de tareas de acción para respuestas efectivas a desastres y fallas.

La **continuidad del negocio** para aquellos eventos considerados como desastre no evitan que ocurran, pero sí puede en algunos casos bajar la probabilidad de ocurrencia. Ante la ocurrencia, la diferencia está en la respuesta, en estar preparados para lo que puede pasar: sabemos exactamente lo que hay que hacer, lo hemos ensayado y lo haremos en forma ordenada.

**Situación – Mitigación de Impacto mediante BCM efectiva – Disrupción Brusca:**



**Situación – Mitigación de Impacto mediante BCM efectiva – Disrupción Gradual:**



## Estrategias de Protección ante desastres

En las instalaciones *on-premise*, los datos son una parte importante, por lo que existen algunas estrategias de protección de datos para mitigar impactos de un desastre:

- *Backups* (copias de resguardo) en discos locales y externos.
- *Backups* periódicos (copias de resguardo) en cintas magnéticas, con y sin almacenamiento de manera externa en discos locales y externos.
- Replicación de datos en un sitio externo, fuera del ámbito donde se usan.
- Replicación de datos en un *datacenter* externo implementado como sitio de contingencia.

Ahora bien, los datos deben montarse sobre una infraestructura, la cual también es importante, ya que, si la infraestructura fue afectada por el desastre, los datos no se podrán usar.

Por esa razón, también tiene que haber estrategias de protección de infraestructura.

## Estrategias para Recuperación ante desastres

- Medidas de Prevención → acciones para evitar que los desastres ocurran.
- Medidas de Detección → controles para la detección de desastres e inmediatos avisos.
- Medidas de Corrección → acciones para recuperar la operatoria de los sistemas.
  - Una vez producido el daño, es importante que se frene y que no se dañe nada más.
  - Frenado el daño, se debe hacer un diagnóstico de los daños.
  - Luego, se debe trabajar para recuperar el nivel operativo mínimo y aceptable.
  - Finalmente, trabajar para volver a la normalidad.

# Gestión de Recursos Humanos

La gestión de RRHH es el conjunto de actividades que ponen en funcionamiento, desarrollan y movilizan a las personas para que una organización alcance sus objetivos:

- Intervienen todas las personas de la organización.
- Busca captar, atraer, conservar, retener y desarrollar los RRHH.
- Busca crear, mantener y desarrollar un conjunto de personas con habilidades, motivación y satisfacción suficientes para conseguir los objetivos de la organización.

## Liderazgo

### LIDERAZGO

Capacidad de influir en otros y apoyarlos para que trabajen con entusiasmo en el logro de objetivos comunes.

Los líderes efectivos tienen ciertas características:

- Son buenos comunicadores.
- Son flexibles → adaptan su estilo de liderazgo a las necesidades de sus subordinados.
- Te sacan presión.
- Saben cómo administrar y resolver los conflictos de grupo.
- Saben planificar y conocen con precisión los roles de cada integrante del equipo.
- Delegan la autoridad entre sus subordinados.
- Generan sinergia entre los miembros del equipo.
- Definen objetivos y roles claros, compartidos por todos los integrantes del equipo.

Liderar implica administrar, pero administrar no implica liderar:

- **Administrar** → refiere al planeamiento, al control y a la ejecución respecto de la asignación eficiente de los recursos/personas a las tareas.
- **Liderar** → no solamente se asignan tareas, se controla y gestiona, sino que también se influye en el comportamiento de las personas.

## Negociación

### NEGOCIACIÓN

- Proceso de comunicación empleado para influir en el comportamiento de los demás donde ambas partes lleguen a un acuerdo *win-win* (donde ambos ganen).
- Proceso por el cual las partes interesadas resuelven conflictos, acuerdan líneas de conducta, buscan ventajas individuales/colectivas, procuran obtener resultados que sirvan a sus intereses mutuos.

- Buen líder es aquel que gana al menos una negociación.
- Una buena herramienta para ganar una negociación es usar métricas para medir desempeños, disminuyendo la subjetividad y haciendo que sea un argumento más difícil de contrarrestar.

## Pirámide Motivacional de Maslow

La pirámide de Maslow es una jerarquía de necesidades humanas, donde los humanos desarrollamos necesidades/deseos más elevados conforme se van satisfaciendo las necesidades más básicas.

- Conforme se satisfacen las necesidades más básicas, los humanos desarrollamos necesidades o deseos más elevados.
- Las necesidades superiores ocupan nuestra atención únicamente cuando se han satisfecho las necesidades inferiores de la pirámide.

Las necesidades que aparecen en la pirámide son las siguientes:

- Necesidades de Autorrealización → encontrarle un sentido válido a la vida mediante el desarrollo potencial de una actividad.
- Necesidades de Estima → respeto a uno mismo y a las demás personas.
- Necesidades Sociales → relacionadas con los afectos del individuo.
- Necesidades de Seguridad → relacionadas a la seguridad física: empleo, casa, etc.
- Necesidades Fisiológicas → necesidades básicas referentes a la supervivencia.



## Gestión del Cambio Organizacional

La gestión del cambio organizacional es el proceso que busca tanto mitigar los efectos no deseados del cambio en cuestión (sea externo o interno) como aumentar las posibilidades de crear futuro en la organización, su gente y contexto.

- Quiebre → ruptura o cambio brusco en las recurrencias o “pilotos automáticos” en los que funcionan ciertos comportamientos, procesos, metodologías o prácticas.
- Transformación → proceso que surge por hechos externos a la organización en pos de un futuro mejor e implican estructuras profundas de los sistemas.
- Cambio → proceso que surge por hechos internos a la organización, respondiendo a una demanda de adaptación dentro del sistema.
- Fuerzas Impulsoras de Cambio:
  - Motivación → proporcionar motivos para una acción.
  - Persuasión → convencer con argumentos a alguien de algo.
- Fuerzas Restrictivas del Cambio → resistencias individual y organizacional.

# Plan de Negocios

## PLAN DE NEGOCIOS · BUSINESS PLAN

Evaluación económica y financiera sobre cómo va a funcionar una organización, del presente a 1 año o a 5 años.

El plan de negocios ataca dos aspectos:

- Aspecto Económico → se refiere al resultado del ejercicio, es decir, se refiere a si una organización gana o pierde dinero.
  - Los ingresos/egresos se contabilizan al comprar/vender, sin importar cuándo se realicen los pagos/cobros de dinero (→ devengado).
- Aspecto Financiero → se relaciona con el instante en que entra/sale dinero de la empresa, es decir, con el instante en que se realizan los cobros/pagos de dinero.
  - Los ingresos/egresos se contabilizan recién al realizarse los cobros/pagos de dinero.

## Impuestos

- Impuestos Directos → gravan un conjunto de operaciones → gravan el sujeto.
  - Ejemplo: impuesto a las ganancias.
- Impuestos Indirectos → gravan la transacción comercial → gravan el bien/servicio.
  - Ejemplo: IVA, IIBB.

Tipos de Impuestos:

- IVA · Impuesto al Valor Agregado → grava las compras/ventas devengadas con total independencia del pago/cobro percibido.
  - Recaudado por el Estado Nacional.
  - La tasa puede ser del 0%, 21% o 27%, según el bien/servicio.
  - Es trasladable → no afecta el aspecto económico (flujo de fondos).
  - Se paga todos los meses → afecta el aspecto financiero.
- IIBB · Ingresos Brutos → grava los ingresos provenientes de la explotación del negocio.
  - Recaudado por el Estado Provincial.
  - La tasa está entre 0% y 15%, según la provincia.
  - No es trasladable → el fabricante está exento del impuesto.
  - Como se aplica sobre el total de la operación, puede afectar la rentabilidad significativamente.
- IDCB · Impuestos a los Débitos y Créditos Bancarios · “Impuesto al Cheque” → grava todos los débitos y créditos bancarios, a una tasa de 0,6%.
- Impuesto a las Ganancias → grava las ganancias, según definición del fisco (considerando amortizaciones<sup>3</sup>), a una tasa de 35%.

---

<sup>3</sup> Las amortizaciones se pueden registrar como pérdidas, pero es el Estado Nacional quien regula cómo se registran esas pérdidas → los bienes de uso no se consumen en su totalidad en el primer año, sino que deben amortizarse a lo largo de su vida útil (cada bien de uso tiene su propia vida útil).

## Amortización

### AMORTIZACIÓN

Depreciación que sufren los bienes por su uso, obsolescencia o paso del tiempo.

- Como el bien pierde valor, se contabiliza como una pérdida.
- Se asocia al concepto de **inversión**, que es distinto al concepto de **gasto**:
  - La **inversión** aumentar el valor productivo, restando capital de la empresa.
    - Está asociada a un bien/servicio NO consumible a corto plazo.
    - El mencionado bien/servicio se amortiza.
  - El **gasto** NO aumenta el valor productivo, restando capital de la empresa.
    - Está asociado a un bien/servicio consumible a corto plazo.
    - El mencionado bien/servicio NO se amortiza.

## Leasing

### LEASING

Contrato de alquiler de un bien (se paga por su uso) con opción de compra del mismo al finalizar el período de alquiler.

- Terminado el período de alquiler, si quien alquiló el bien quisiera venderlo, quien tuvo el *leasing* tiene prioridad si desea comprarlo.

## Costo Laboral<sup>4</sup>

El costo laboral está compuesto por:

- Sueldo Bruto → compuesto por: sueldo neto y aportes y deducciones.
- Cargas Sociales.
- SAC (Sueldo Anual Complementario) o Aguinaldo.
- Vacaciones.
- Licencias por examen.
- Licencias por enfermedad.
- Regalos.

Las horas efectivas de trabajo hace referencia a la cantidad de horas en que realmente se produce y tiene una implicancia directa en la planificación, ya que la duración de las tareas resulta del esfuerzo diario efectivo que puede entregar cada persona.

---

Ejemplo de primer año de actividad: ventas por \$50M, inversión en bienes de uso por \$33M y otros costos por \$2M. Según el contribuyente, la ganancia sería de \$15M... En realidad, para el 1<sup>er</sup> año de actividad, sólo se toman \$3,3M (\$33M repartidos en 10 años son \$3,3M por año) en concepto de costo de amortización (y \$3,3M para cada año de los siguientes), por lo que la ganancia será de \$44,7M. Entonces, el impuesto a las ganancias será el 35% de dicha suma.

<sup>4</sup> Se refiere al costo laboral de un empleado desde el punto de vista del empleador.



# EVM • Gestión del Valor Ganado

## [EVM] EARNED VALUE MANAGEMENT • GESTIÓN DEL VALOR GANADO

Método utilizado para conocer el estado del proyecto. Integra alcance, cronograma y costos para medir el rendimiento y el avance del proyecto en forma objetiva.

## Variables Principales: PV, EV y AC

<b>[PV] PLANNED VALUE VALOR PLANEADO</b>	<b>[EV] EARNED VALUE VALOR GANADO</b>	<b>[AC] ACTUAL COST COSTO REAL</b>
<u>Lo que dijimos que costaría</u> aquello que <u>dijimos que íbamos a hacer.</u>	<u>Lo que dijimos costaría</u> aquello que <u>realmente hicimos.</u>	<u>Lo que realmente costó</u> aquello que <u>realmente hicimos.</u>

- Al inicio del proyecto,  $PV = 0$ .
- Al finalizar el proyecto, PV equivale al presupuesto del proyecto.

## Estado del Proyecto: SV, CV, SPI y CPI

**Avance (SV, SPI)**  
**según cronograma:**

**Rendimiento y Costo (CV, CPI)**  
**según presupuesto:**

<b>[SV] SCHEDULE VARIANCE VARIACIÓN DE CRONOGRAMA</b> $SV = EV - PV$ <b>[SPI] SCHEDULE PERFORMANCE INDEX ÍNDICE DE PERFORMANCE DE CRONOGRAMA</b> $SPI = \frac{EV}{PV}$ $0 < SPI < 1$  Si $SV > 0 \Leftrightarrow SPI > 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>adelantados</u> respecto del cronograma.</li><li>• <u>hicimos más</u> de lo esperado.</li></ul> Si $SV = 0 \Leftrightarrow SPI = 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>al día</u> respecto del cronograma.</li><li>• hicimos exactamente lo esperado.</li></ul> Si $SV < 0 \Leftrightarrow 0 < SPI < 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>atrasados</u> respecto del cronograma.</li><li>• <u>hicimos menos</u> de lo esperado.</li></ul>	<b>[CV] COST VARIANCE VARIACIÓN DE COSTOS</b> $CV = EV - AC$ <b>[CPI] COST PERFORMANCE INDEX ÍNDICE DE PERFORMANCE DE COSTOS</b> $CPI = \frac{EV}{AC}$ $0 < CPI < 1$  Si $CV > 0 \Leftrightarrow CPI > 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>por debajo del presupuesto</u>.</li><li>• <u>estamos gastando menos</u> de lo esperado.</li></ul> Si $CV = 0 \Leftrightarrow CPI = 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>dentro del presupuesto</u>.</li><li>• gastamos exactamente lo esperado.</li></ul> Si $CV < 0 \Leftrightarrow 0 < CPI < 1$ , entonces: <ul style="list-style-type: none"><li>• estamos <u>por encima del presupuesto</u>.</li><li>• <u>gastamos más</u> de lo esperado.</li></ul>
---	--

# Otros Indicadores: BAC, EAC, CPI<sub>NUEVO</sub>, VAC, ETC y TCPI

## BAC

### ~~BUDGET AT COMPLETION~~ • PRESUPUESTO HASTA LA CONCLUSIÓN

- Costo presupuestado inicialmente para todo el proyecto.
- Lo que dijimos inicialmente que iba a costar todo el proyecto.

## EAC

### ~~ESTIMATE AT COMPLETION~~ • ESTIMADO A LA CONCLUSIÓN

- Costo proyectado al final del proyecto.
- La reestimación del costo del proyecto durante su ejecución, la cual se espera que resulte más certera que la estimación inicial [BAC].

El valor del **EAC** depende de cómo imaginemos el futuro del proyecto.

Por esa razón, se plantean 4 posibles escenarios:

1. **Desempeño Típico** → si la **CPI** observada hasta el momento se mantendrá igual, entonces:

$$EAC = \frac{BAC}{CPI}$$

2. **Desempeño Atípico** → la **CPI** observada hasta el momento ha sido excepcional y, además, de aquí en adelante la **CPI** corresponderá a lo planificado, entonces:

$$EAC = AC + (BAC - EV)$$

3. **Cambio a Desempeño Diferente** → la **CPI** observada hasta el momento no se mantendrá, sino que de aquí en adelante habrá un **CPI nuevo** (pudiendo o no ser lo planificado), entonces:

$$EAC = AC + \frac{BAC - EV}{CPI_{NUEVO}}$$

... donde el valor del **CPI nuevo** no siempre será el mismo:

Si el **CPI nuevo** debe permitir concluir el proyecto dentro del **BAC**, entonces:

$$CPI_{NUEVO} = \frac{BAC - EV}{BAC - AC}$$

Si el **CPI nuevo** estará afectado por la **SPI** observada, entonces:

$$CPI_{NUEVO} = CPI \cdot SPI$$

4. **Nueva Estimación Detallada** → los supuestos de la estimación original no resultan válidos (ni siquiera modificando la estimación original) porque los desvíos en la performance son atribuibles a una mala estimación o bien porque las condiciones del proyecto han cambiado significativamente.

Para lo que resta del proyecto se debe realizar una nueva estimación detallada, entonces:

$$EAC = AC + \text{Nueva Estimación}$$

## VAC

### ~~VARIANCE AT COMPLETION~~ • VARIACIÓN A LA CONCLUSIÓN

- Desvío en el costo total del proyecto.
- Variación final entre lo planificado originalmente (**BAC**) y la nueva estimación (**EAC**).

$$VAC = BAC - EAC$$

## ETC

### ~~ESTIMATE TO COMPLETE~~ • ESTIMADO PARA COMPLETAR

- Costo estimado para completar el proyecto.

$$ETC = EAC - AC$$

## TCPI

### ~~TO COMPLETE PERFORMANCE INDEX~~ • ÍNDICE DE PERFORMANCE DE COSTOS REQUERIDO PARA FINALIZAR (DENTRO DE BAC o EAC)

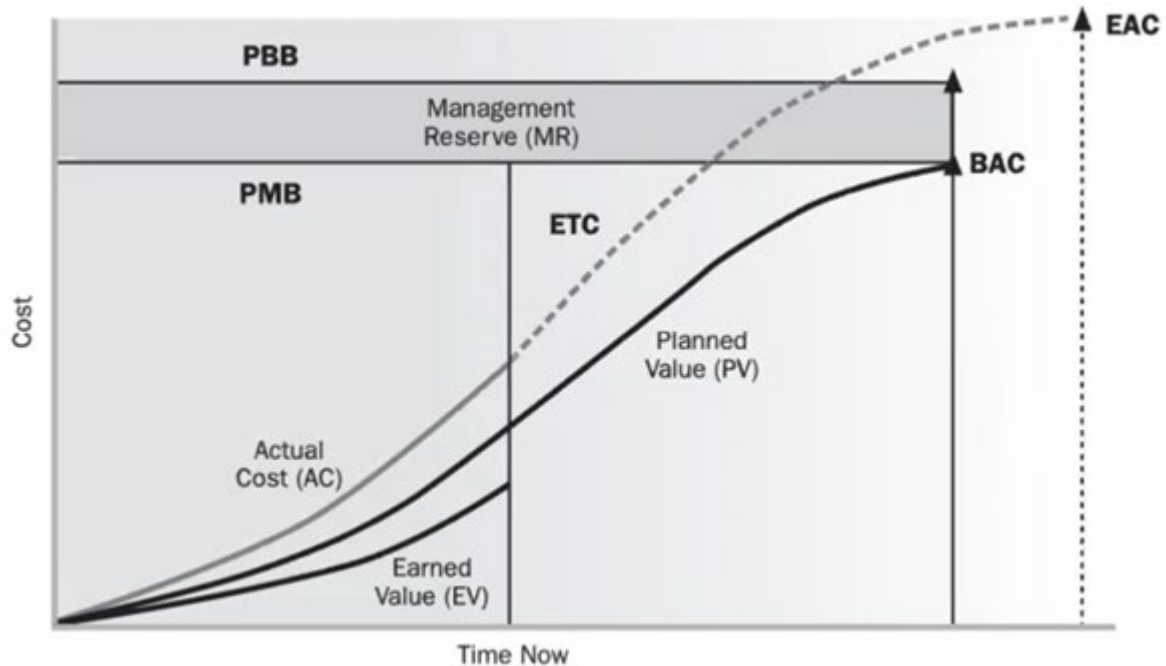
- Relación entre el costo para terminar el trabajo pendiente y el presupuesto restante.
- Representa la eficiencia que es necesaria mantener para cumplir el plan.

$$TCPI = \frac{BAC - EV}{BAC - AC}$$

Ahora bien, si hay un nuevo presupuesto y distinto al planificado originalmente, entonces:

$$TCPI = \frac{BAC - EV}{EAC - AC}$$

## Ejemplo/



Para medir el **avance** (según cronograma), se usan SV y SPI:

Resulta  $SV < 0 \Leftrightarrow 0 < SPI < 1$ , por lo tanto:

- estamos atrasados respecto del cronograma.
- hicimos menos de lo esperado.

Para medir el **rendimiento y costo** (según el presupuesto), se usan CV y CPI:

Resulta  $CV < 0 \Leftrightarrow 0 < CPI < 1$ , por lo tanto:

- estamos por encima del presupuesto.
- estamos gastando más de lo esperado.

## Enfoque EVM Ágil

- Se toman marcos temporales reducidos.
  - La incertidumbre avanza conforme pasa el tiempo, por lo que no tiene sentido hacer un plan hasta el final del proyecto.
  - Los marcos temporales no deben ser muy largos ni muy cortos.
- El tamaño del *backlog* del proyecto se mide en valores relativos → el tamaño estimado para el proyecto será la suma de todos los ítems agrupados por sprints.
  - Al proyecto se lo va entregando en sprints.
  - Cada sprint tendrá cierta cantidad de puntos.
  - Para hacer la conversión costos-puntos, se divide el presupuesto entre la cantidad de puntos totales → así se determina cuánto cuesta cada punto.  
Conociendo cuánto cuesta 1 release y conociendo cuántos puntos va a entregar, puedo determinar cuánto cuesta 1 punto (según el presupuesto).  
Sabendo cuánto cuesta 1 punto, puedo determinar cuánto cuesta cualquier cantidad de puntos.
  - La relación entre punto y costo debe ser constante para todos los ítems del *backlog* incluidos en el release.

# EVM Tradicional vs EVM Ágil

EVM TRADICIONAL		EVM ÁGIL
\$.	<b>Unidad de Medida</b>	Puntos y \$.
Presupuesto del proyecto.	<b>BAC</b>	Presupuesto del <i>release</i> .
PV para cada período del proyecto.	<b>Baseline (Línea de Base)</b>	Cantidad de puntos (y su equivalente presupuestario) que deben completarse en cada <i>sprint</i> .
Costo presupuestario del trabajo que espera realizarse para un momento del proyecto.	<b>PV</b>	Cantidad de puntos (y su equivalente presupuestario) que deben completarse al finalizar un <i>sprint</i> .
Costo presupuestario del trabajo realizado para un momento del proyecto.	<b>EV</b>	Cantidad de puntos (y su equivalente presupuestario) que realmente se completaron al finalizar un <i>sprint</i> .
Costo real del trabajo realizado acumulado para un momento del proyecto.	<b>AC</b>	Costo real acumulado de los puntos completados al finalizar un <i>sprint</i> .
Tasa de avance lograda en comparación con el cronograma original. $SPI = \frac{EV}{PV}$	<b>SPI</b>	Tasa de avance lograda en comparación con el cronograma original. $SPI = \frac{\text{Cant. Puntos Entregados}}{\text{Cant. Planeada de Puntos}}$
Cuánto se obtiene por unidad de costo comparado con el estimado originalmente. $CPI = \frac{EV}{AC}$	<b>CPI</b>	Cuánto se obtiene por unidad de costo comparado con el estimado originalmente. $CPI = \frac{\text{Costo Planeado por Punto}}{\text{Costo Real por Punto}}$

# Gestión de Abastecimiento – Benchmark – MEP

## Gestión de Abastecimiento o Compras

### GESTIÓN DE ABASTECIMIENTO o COMPRAS

Uso de los recursos que disponemos de manera efectiva y eficaz para poder mejorar el proceso de compra de los bienes/servicios que necesita la organización para su funcionamiento.

### Proceso de Abastecimiento/Compra – Pasos

5. Definición de Requerimientos de Compra:

- Debemos ser muy precisos para comprar algo que satisfaga nuestras necesidades.
- Se traducen las necesidades de un usuario en requerimientos para los proveedores.

6. Selección del Mecanismo de Compra:

- Se determina de qué forma adquiriremos dicho bien o servicio.
- Los mecanismos de compra están definidos por las leyes de compras públicas de cada jurisdicción y/o por reglamentos de compras internos de cada organización.
- Los mecanismos de compra más comúnmente utilizados consisten en:
  - Convenios/Acuerdos Marco → pensado para compras habituales donde se especifica un convenio con todos los aspectos generales de una compra y, después, en cada instancia, se agregan aspectos particulares de esa compra.
  - Licitación Pública o Privada → se llama a varios proveedores para que oferten un producto/servicio.
  - Trato Directo → contacto directo con el proveedor.

7. Llamado y Recepción de Ofertas → contacto con los proveedores, donde les pedimos que nos oferten para recibir sus propuestas.

8. Evaluación de las Ofertas Recibidas → se analizan y se elige la mejor [→ ver **MEP**].

9. Adjudicación de Ofertas:

- Se cierra la evaluación y se decide a quién se comprará.
- Se deben formalizar, documentar y comunicar los acuerdos administrativos.

10. Recepción del Producto/Servicio:

- Se controla que todo lo pautado se cumpla.
- La conformidad del comprador permite que el proveedor pueda cobrar.

11. Seguimiento y Monitoreo de la Compra:

- Se evalúan los proveedores.
- Se revisan periódicamente fechas de término y renovación de contratos.
- Se aclaran los mecanismos de garantías de compra.
- Se ordena información relevante para futuras compras.

# Benchmark

## BENCHMARK

Evaluación de desempeño, manifestada con métricas, de algo que nos interesa.

- Siempre implica una comparación.
- Sirve para comprar elementos a través de características claves para la solución.
- Permite obtener un resultado objetivo, con la mejor relación costo/beneficio.
- Comprueba si los elementos estudiados se adecúan a las necesidades.
- Ventaja → permite obtener muy buena información sobre algo que nos interesa, que nos puede llevar a comprar ese algo o a recomendarlo
- Desventaja → es costoso<sup>5</sup>.

## Proceso de Benchmarking

1. Determinar el elemento de estudio:
  - a. Determinar qué se someterá a estudio.
  - b. Elegir factores y variables claves.
  - c. Seleccionar las opciones disponibles que ofrece el mercado.
2. Preparar el entorno de prueba:
  - a. Recopilar requerimientos del tipo de *benchmark*.
  - b. Realizar tareas previas a la etapa de ejecución.
3. Realizar el *benchmark*:
  - a. Someter el elemento a pruebas.
  - b. Tomar muestras de las respuestas de las distintas variables analizadas.
  - c. Realizar comparaciones y obtener resultados.
4. Analizar resultados de la medición:
  - a. Descartar elementos que no cumplen con las necesidades.
  - b. Informar resultados.
  - c. Determinar si se requiere recalibrar el *benchmark*.
  - d. Desarrollar planes de acción.

## [TPC]TransactionProcessingPerformanceCouncil

El TPC es una organización que define, mediante simulaciones de ejecución, *benchmarks* de medición de procesamiento de transacciones en bases de datos con un alto grado de sofisticación (reduciendo al mínimo la interferencia subjetiva de medición).

---

<sup>5</sup> Para las grandes decisiones, las organizaciones no realizan *benchmarks*, sino que compran uno publicado por profesionales y, en base a eso, aproximan. Si bien no es lo mismo, resulta más barato.

# MEP • Método de Evaluación de Propuestas

## Paso 1: Armar la TABLA DE REQUERIMIENTOS

Primero, se analiza la información de quien quiere comprar.

Luego, se arma una tabla con 4 columnas ([1] y [2] obligatorias; [3] y [4] no obligatorias):

1. Columna INDISPENSABLE/OBLIGATORIO → se listan los requerimientos obligatorios; aquellos que deben cumplirse sí o sí para que una propuesta pueda ser evaluada.
  - Una propuesta será evaluada únicamente si cumple con todos estos requerimientos.
  - Si una propuesta no cumple con un requerimiento de esta columna, no será tenida en cuenta para su evaluación.
2. Columna PREFERIDO/DESEABLE → se enumeran listan los requerimientos deseados; aquellos que marcan diferencias con el resto de las propuestas, aquellos que *suman puntos*.
  - Hay requerimientos que pueden estar en esta columna y también en la anterior → estos requerimientos no solamente son obligatorios, sino que marcan una diferencia.
  - *Ejemplo 1: un SLA para incidentes de cierta prioridad → no es lo mismo que sea solamente menor o igual a 5 horas [obligatorio] a que sea menor a 3 horas [deseable], por ejemplo.*
  - *Ejemplo 2: el costo total → no es lo mismo que sea sólo menor a \$1.000.000 [obligatorio] a que sea menor a \$600.000 [deseable], por ejemplo.*
3. Columna NO DESEADO → se listan los requerimientos no deseados; aquellos que *restan puntos*.
4. Columna NO CONSIDERADO → se enumeran las características irrelevantes; aquellas que, de cumplirse, no le suman absolutamente nada a la propuesta.

## Paso 2: Armar la TABLA DE PESOS RELATIVOS

Se consideran solamente los requerimientos enumerados en la columna PREFERIDO/DESEABLE.

Se agrupan tales requerimientos por características (físicas, técnicas, de funcionamiento, comerciales, etc.) y se agrega una característica COSTO al final de todo.

La tabla muestra cuánto pesa cada ítem de acuerdo a nuestro nivel de satisfacción.

Luego, se arma una tabla con 4 columnas:

1. Columna ÍTEM → se tiene una fila por cada ítem, incluidas las características que los agrupan.
  - El ítem COSTO debe estar sí o sí, y sin desagregar.
2. Columna NIVEL 1 (N1) → se indica el porcentaje de peso relativo asignado (sobre 100%), en forma arbitraria, a cada característica (no a cada ítem).
  - Los valores de la columna deben sumar 100%.
  - El valor del ítem COSTO debe ser alrededor del 20% (siempre menor al 25%), de manera que el costo no se lleve puesto al resto de las características o ítems.
3. Columna NIVEL 2 (N2) → se indica el porcentaje de peso relativo asignado (sobre 100%), en forma arbitraria, a cada ítem perteneciente a una misma característica.
  - Para una misma característica, los valores de esa columna deben sumar 100%.
4. Columna NIVEL 3 (N3) → se indica el porcentaje de peso relativo de cada ítem (sobre 100%).
  - Los valores de la columna deben sumar 100%.



### **Paso 3: Armar la TABLA DE VALORACIÓN DE ATRIBUTOS**

Los atributos son las alternativas ofrecidas por el mercado para los ítems a evaluar.

La tabla tiene una columna ÍTEM (con sus respectivos agrupamientos introducidos por la TABLA DE PESOS RELATIVOS), donde cada ítem “se abre” con todos los valores posibles.

La asignación de valores deberá estar entre 0 y 100, siendo:

- 0 para el atributo que cumpla mínimamente con el requerimiento.
- 100 para lo que más satisfacción nos daría.

Existen distintos tipos de atributos:

- Atributos Mutuamente Excluyentes → se excluyen entre sí.
  - Dada una lista de N atributos posibles, una alternativa solamente puede tener 1.
  - Símil *RadioButton*.
- Atributos Binarios → sólo se tienen los extremos.
  - Es todo o nada: 100 o 0.
  - Son un caso particular de los mutuamente excluyentes.
- Atributos Aditivos → no se excluyen entre sí, sino que pueden ir sumándose entre ellos.
  - Dada una lista de N atributos posibles, una alternativa puede tener hasta N atributos.
  - Símil *CheckButton*.
  - Sumadas todas las opciones, el resultado debe ser 100.
- Atributos Definidos por una Función Lineal → atributos donde la satisfacción no varía de forma escalonada, sino de forma continua y de manera lineal.
  - La función lineal permite una variación continua de la preferencia en función del atributo.
  - *Ejemplo: el ítem Costo varía de forma continua: a mayor costo, menor satisfacción y viceversa (la pendiente de la función será negativa).*

*Para armar la función lineal del Costo, primero debemos saber si el presupuesto máximo está definido:*

- *Si el presupuesto máximo está definido, entonces será el CM<sup>6</sup>. Luego, se estimará un Cm<sup>7</sup> razonable.*
- *Si el presupuesto máximo no está definido, entonces habrá que hacer una investigación de mercado para establecer precios máximo y mínimo: para obtener el CM, se le agrega un 5% al precio máximo recabado; para obtener el Cm, se le resta un 5% al precio mínimo recabado.*

*Conociendo CM y Cm, se arma la función lineal planteando lo siguiente:*

$$f(x) = ax + b \Rightarrow \left\{ \begin{array}{l} f(\text{CM}) = 0 \\ f(\text{Cm}) = 100 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} a \cdot \text{CM} + b = 0 \\ a \cdot \text{Cm} + b = 100 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} a = \dots \\ b = \dots \end{array} \right\} \Rightarrow \boxed{f(x) = ax + b}$$

- *Otros ejemplos: magnitudes físicas como masas, longitudes, alturas, superficies, volúmenes, etc.*

---

<sup>6</sup> CM = costo máximo con margen de seguridad.

<sup>7</sup> Cm = costo mínimo con margen de seguridad.

## Paso 4: Armar la TABLA DE PONDERACIÓN DE PROPUESTAS

Se aplica cada propuesta<sup>8</sup> recibida a la **TABLA DE VALORACIÓN DE ATRIBUTOS**, obteniéndose ponderaciones para cada ítem de cada propuesta.

Para cada propuesta, se arman 3 columnas:

1. **Columna ATRIBUTO** → similar a la **TABLA DE VALORACIÓN DE ATRIBUTOS**, pero ahora se asigna el atributo de dicha propuesta en particular.
2. **Columna VALOR** → similar a la **TABLA DE VALORACIÓN DE ATRIBUTOS**, pero ahora se asigna el valor correspondiente de acuerdo a cada atributo de dicha propuesta en particular.
3. **Columna PONDERACIÓN** → se colocan los valores ponderados de atributo de cada ítem.

Finalmente:

- La sumatoria de los valores de cada ponderación corresponde al puntaje que obtiene cada propuesta, siendo la mejor (es decir, la que mayor satisfacción brindará) aquella que mayor puntaje obtenga.
- Si la diferencia de puntaje final entre dos alternativas es muy pequeña (una diferencia de 2 puntos, por ejemplo), no significa que una sea mejor que la otra: esa estrecha diferencia está sujeta tanto a la subjetividad de quien realizó la evaluación como al error del método. Considerando esa diferencia, ambas propuestas son prácticamente equivalentes [→ ver VPP]

### [VPP] Valor de Punto de Ponderación

#### VALOR DE PUNTO DE PONDERACIÓN

Cantidad de dinero que uno debería estar dispuesto a pagar (como máximo) por un incremento de 1 punto de ponderación.

Si el costo tiene un peso relativo del, supongamos, 20%, el costo puede variar entre 0 y 20 puntos de ponderación y además le corresponde una variación entre  $C_m$  y  $CM$ , por lo que se podrá determinar cuánto dinero representa 1 punto de ponderación.

$$VPP = \frac{CM - C_m}{\text{Peso Relativo del Costo en la TABLA DE PESOS RELATIVOS}(\sim 20\%)}$$

Si se quiere mejorar un ítem de una propuesta recibida o bien agregar algo que la propuesta recibida no lo tiene, podemos calcular cuánta plata estaríamos dispuestos a pagar por esa mejora. Para eso, se debe obtener la cantidad de puntos de ponderación que representa esa mejora (ese “salto”, es decir, la diferencia entre el valor ofertado y el valor al que se aspira llegar) y multiplicar dicha cantidad por el VPP.

---

<sup>8</sup> Propuesta, oferta y alternativa refieren al mismo concepto.

# Big Data

## BIG DATA

- Conjuntos extensos de datos, principalmente en las características de volumen, variedad, velocidad y/o variabilidad, que requieren una arquitectura escalable para su almacenamiento, manipulación y análisis eficientes.
- Conjunto de datos cuyo tamaño supera la capacidad de las herramientas típicas de SW de base de datos para capturar, almacenar, administrar y analizar.

## Las 5 'V'

- Volumen → grandes volúmenes de datos.
- Variedad → diversos formatos de datos procedentes de diferentes fuentes de información.
- Velocidad → altas velocidades de acumulación y procesamiento de datos.
- Veracidad → precisión e integridad en la generación y procesamiento de los datos.
- Valor → valor propio de los datos recolectados y analizados para el negocio.

## Usos

- UX (Experiencia de Usuario).
- Desarrollo de Productos.
- Fraude y conformidad.
- Eficiencia operacional.
- Mantenimiento predictivo.

## Desafíos

- Lidiar con el crecimiento de los datos.
- Generar conocimiento en forma oportuna.
- Reclutar y retener talento de *Big Data*.
- Integrar diferentes fuentes de datos.
- Validación de datos.
- Seguridad.

## Tecnologías y Productos

- |                            |                                  |
|----------------------------|----------------------------------|
| • Inteligencia Artificial. | • R.                             |
| • <i>Blockchain</i> .      | • Data Lakes.                    |
| • Bases de Datos NoSQL.    | • Predictive Analytics.          |
| • Python.                  | • Big Data Governance Solutions. |
| • Hadoop.                  | • Big Data Security Solutions.   |
| • Spark.                   | • Prescriptive Analytics.        |

## Seguridad

### SEGURIDAD EN BIG DATA

Conjunto de acciones de protección de datos y de procesos de análisis frente a factores que podrían comprometer su confidencialidad e integridad.

- Requiere combinaciones de herramientas de seguridad y procesos inteligentes para monitorear la seguridad a lo largo de la vida de la plataforma.
- Se busca que los datos se enruten a través de un circuito establecido no vulnerable.
- Opera sobre datos de entrada, datos almacenados y datos de salida.
- Tecnologías:
  - Cifrado → protección de datos en tránsito y en reposo.
  - Control de Acceso a Usuarios → configuración de acceso con roles y usuarios.
  - Detección y Prevención de Intrusiones.
  - Seguridad Física → debe considerarse siempre.
  - Gestión Centralizada de Claves → registro de uso, entrega de claves bajo demanda, abstracción de la administración de claves respecto de su uso, etc.
- Los responsables de la seguridad son los DBAs, los programadores, las áreas de calidad y las áreas de *compliance*.

## Data Engineering, Data Science, Data Analytics<sup>9</sup>

### DATA ENGINEERING

Ingeniería que se dedica a superar los cuellos de botella en el procesamiento de datos y los problemas de manejo de datos para aplicaciones que usan *Big Data*.

### DATA SCIENCE

Campo multidisciplinario centrado en encontrar información procesable a partir de grandes conjuntos de datos tanto sin procesar como estructurados.

- Focalizada en encontrar respuestas a las cosas que no sabemos que no sabemos.
- Se usan varias técnicas para obtener respuestas, mediante análisis predictivo, ciencias de la computación, estadísticas y machine learning para analizar conjuntos masivos de datos en un esfuerzo por establecer soluciones a problemas que aún no fueron pensados.

### DATA ANALYTICS

Centrado en procesar y realizar análisis estadísticos en conjuntos de datos existentes, con el objetivo de mostrar dicha información con una herramienta adecuada.

- Se busca crear métodos para capturar, procesar y organizar datos que permitan tanto descubrir información procesable sobre problemas actuales como establecer la mejor manera de presentar estos datos.
- Busca resolver problemas disparados por preguntas cuyas respuestas aún no conocemos, pudiendo generar resultados que pueden conducir a mejoras inmediatas.

---

<sup>9</sup> Data Engineering nutre a Data Science y Data Analytics, quienes permiten la toma de decisiones.

# Blockchain

## BLOCKCHAIN

- Red P2P que no depende de entidades centralizadas para llegar a un consenso.
- Tecnología que permite tener un registro distribuido donde cada par tiene su propia copia de la información, cuya validez y veracidad se establece por consenso entre los pares.

## Conceptos Técnicos

- P2P → protocolo de red de comunicación entre pares.
- Algoritmo de Hash → función que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija.
- Criptografía asimétrica → sistema que utiliza dos claves (una clave pública para cifrar y una clave privada para descifrar) para el envío de datos.
- Proof of Work o Prueba de Trabajo → algoritmo de consenso basado en operaciones matemáticas complejas y en fuerza bruta, utilizado para la confirmación de las transacciones y la generación de nuevos bloques.
- Consenso → que toda la red esté de acuerdo con el resultado de una prueba.

## Principios de Blockchain

- Integridad en la Red → la integridad está cifrada y distribuida en todas las etapas del proceso y no depende de cada individuo.
- Poder Distribuido → el sistema distribuye el poder por una red de iguales sin que haya ningún punto de control → las partes no pueden apagar el sistema por sí solas.
- El Valor como Incentivo → el sistema alinea los incentivos de los *stakeholders* y sus intereses.
- Seguridad → se garantiza la confidencialidad y la autenticidad de todas las actividades.
- Privacidad → se elimina la necesidad de conocer la verdadera identidad de los pares.
- Preservación de Derechos → los derechos de propiedad están garantizados.
- Inclusión → la economía funciona mejor cuando funciona para todos.

## Pasos

1. Un usuario solicita una transacción.
2. Se crea un bloque que representa esa transacción.
3. El bloque se difunde a todos los nodos de la red.
4. Dichos nodos validan el bloque y, así, también validan la transacción.
5. El bloque se añade a la cadena.
6. La transacción se verifica y se ejecuta → la transacción ya no se podrá modificar.

## Usos

- Cualquiera donde se quiera asegurar que la información no se altere.
- Contratos.
- Libros contables y libros de registros.
- *Tokenización* de activos.
- Registros automotores.
- Historias clínicas.
- Escrituras de propiedades.

## Ventajas

- Naturaleza distribuida → no hay ningún nodo privilegiado, ningún nodo puede apagar la red, no tiene un punto único de falla.
- Estabilidad → es muy poco probable que los bloques confirmados sean revertidos.
- Sistema *Trustless* → no requiere de confianza entre terceros.
- Incorruptible → es casi imposible alterar la información contenida en los bloques.
- Transparencia → cualquiera puede consultar las transacciones en el registro y verificarlas.
- Trazabilidad → la información se puede rastrear de manera sencilla y su historial se comprueba constantemente.
- Libre de Errores → los resultados siempre son comprobados y correctos.

## Desventajas

- Ataques del 51% → si un número suficiente de nodos se hace con el control de más del 50%, será la mayoría, pudiendo así crear *nuevas verdades* y *una nueva realidad*.
- Claves privadas → perder la clave privada implica no tener acceso a la información.
- Ineficiente → las *blockchains*, en especial las que usan *proof of work*, usan mucha energía debido al gran procesamiento por la fuerza bruta empleada.
- Almacenamiento → la cantidad de información de cada copia de la *blockchain* es inmensa.
- Apertura → cualquiera puede consultar datos presentes/históricos de cualquier otro.

## Posibles Riesgos o Ataques<sup>10</sup>

- Doble gasto → una misma moneda digital puede gastarse más de una vez.
- Redes fantasmas → red impostora donde se realicen operaciones falsas.

---

<sup>10</sup> El protocolo de *blockchain* soluciona ambos ataques.

## BaaS

### BaaS • BLOCKCHAIN AS A SERVICE

- *Blockchain* pensado como un servicio.
  - Redes basadas en la nube por parte de terceros para empresas en el negocio de la creación de aplicaciones *blockchain*.
- Permiten la integración con contratos inteligentes.
  - Permiten la integración con plataformas que aseguran identidad.
  - Permiten poder trabajar con distintas implementaciones de *blockchain*.
  - Ofrecen mecanismos de consenso basados en la identidad.

## BFA • Blockchain Federal Argentina

BFA es una plataforma multiservicios abierta y participativa pensada para integrar servicios y aplicaciones sobre *blockchain* dentro de Argentina.

Características:

- Sin criptomoneda → la plataforma no está pensada para criptomonedas.
- Modelo liviano → no usa fuerza bruta (el minado) para obtener recompensas.
- Permisiónada → el consenso se logra porque los integrantes se conocen (no son anónimos).
- Transacciones gratuitas → las transferencias no tienen costo.
- Almacenamiento en off-chain → no se almacena información per se dentro de la *blockchain*, sino los hashes de esa información.
- Software libre → el SW se basa en una implementación abierta.

# Arquitectura de Software

## Arquitectura de Software

### ARQUITECTURA DE SOFTWARE

Elementos de tecnología que se relacionan entre sí para que los productos, los sistemas de información y los servicios de tecnología funcionen de la mejor forma posible para que el usuario final los use.

- Se ocupa de las interfaces (que separan lo privado de lo público), centrándose en la comunicación e interacción entre esos elementos.
- Se abstrae de los detalles internos y de cómo funcionan internamente esos elementos

### Interesados en la SA

- Usuario Final → quien usará el sistema de información o producto SW.
  - Interesado en que los sistemas/productos funcionen de la mejor manera posible (se consideran criterios como rapidez, disponibilidad y confiabilidad) para que justamente el usuario final los quiera usar.
- Cliente → quien nos contrató, no necesariamente es el usuario final.
  - Interesado en que se implemente una arquitectura que respete calendario y presupuesto previamente seleccionados.
- Project Manager (PM) → quien define cómo se lleva adelante un proyecto.
  - Interesado en que los equipos trabajen en forma independiente interactuando con disciplina.
- Arquitecto → encargado de comunicar qué arquitectura se implementará.
  - Interactúa con el usuario final, el cliente y el PM.
  - Interesado en dejar conforme a los tres protagonistas mencionados.
- Administradores de Bases de Datos (DBA).
- Desarrolladores.
- Testers.
- Gente de seguridad.

### Elección de la SA – Decisiones de Diseño:

- ¿Procesamiento distribuido o no?
- ¿SW dividido en capas? ¿Cuántas? ¿Qué patrones se usan?
- ¿Comunicación sincrónica o asincrónica?
- ¿Dependemos del sistema operativo que tenemos?
- ¿Dependemos del HW que tenemos?



## Elección de la SA – Contextos y/o Aspectos:

- Técnico → conocimiento.
- Negocio → imposiciones de reglas de negocio que tiene la organización (como por ejemplo los convenios para usar ciertas tecnologías).
- Ciclo de Vida del proyecto → ¿encaja el ciclo de vida de un producto con la SA deseada?
- Profesional → ¿encajan las características del equipo de trabajo con la SA deseada?

## Atributos de Calidad

### ATRIBUTOS DE CALIDAD

Propiedades o medidas de testeo que permiten indicar qué tan bien funciona un sistema y cómo satisfacer las necesidades de los interesados.

- Atributos de Calidad referidos a los requerimientos:
  - Requerimientos Funcionales → qué hará el sistema de información o producto SW.
  - Requerimientos NO Funcionales → caracterizar las funcionalidades.
  - Restricciones de Negocio → reglas de negocio y convenios que tiene la organización.
- Disponibilidad → minimizar las interrupciones del servicio y mitigar posibles fallas.
  - Tácticas: detección, recuperación y prevención de fallas.
- Interoperabilidad → capacidad que tienen dos elementos dentro de la SA para poder relacionarse entre sí vía interfaces.
- Adaptabilidad → capacidad de no sentir resultados de cambios (de plataforma, de sistema operativo, etc.), costos o riesgos.
- Variabilidad → adaptación al contexto.
- Performance → referido al tiempo y a la habilidad.
- Seguridad → referido a la detección de ataques y a cómo resistirlos.
  - Detección de intrusos, denegación de servicios, verificación de integridad, autenticación de actores, límites de acceso, encriptación de datos, etc.
- Usabilidad → cuán fácil es para el usuario ejecutar una tarea deseada.
- Testeabilidad → buena parte del costo de una buena ingeniería en el desarrollo de los sistemas es absorbida por las pruebas.
- Portabilidad → capacidad para adaptarse a los cambios de plataforma.
- Desarrollo Distribuido → diseño del SW.
- Escalabilidad → capacidad de agregar más recursos.
- Monitoreo → controlar e investigar el sistema mientras trabaja.
- Comerciabilidad → si lo que terminamos construyendo se adaptó a lo que quería el negocio.

## Patrones de Arquitectura

### PATRONES DE ARQUITECTURA

Conjunto de soluciones de diseño que se dan bajo contextos y problemas similares.

## Requerimientos de Arquitectura Significativos (ASR)

- Revisar documentación previa sobre qué arquitecturas se implementaron.
- Entrevistar a los interesados en el negocio por qué se eligieron esas arquitecturas.
- Interactuar con el usuario final para entender los objetivos del negocio

## Características de la SA en Proyectos Ágiles

Existe una combinación de arquitecturas que se basan en proyectos ágiles y arquitecturas de paradigmas estructurados, no siempre hay que caer en lo que ofrece el mercado como solución.

No obstante, algunas características son:

- División del proyecto en intervalos de tiempo relativamente cortos (sprints).
- Entregas de SW entre semanas y meses.
- Interacción continua y fluidez de la comunicación entre el equipo de trabajo y el cliente.
- Alta satisfacción del cliente cuando se entrega una versión.

## Gestión y Gobierno de la SA

- Evaluación:
  - El arquitecto debe interesarse por la gestión de proyectos.
  - PM y arquitecto deben trabajar en conjunto por la perspectiva de la organización.
  - A mayor complejidad de proyectos, más útil es la implementación de una arquitectura.
- Planificación:
  - Si bien la planificación sucede constantemente, 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 que puedan 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 *tests* de integración.
  - System Test Manager → testeo de sistema y *testing* de aceptación.
  - Product Manager → representan el marketing.

# Arquitectura Cloud – Cloud Computing

## ARQUITECTURA CLOUD – CLOUD COMPUTING

Modelo que permite el acceso ubicuo (desde cualquier lugar y en diferentes condiciones), conveniente y bajo demanda a través de la red a un conjunto compartido de recursos informáticos configurables (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.

## Características

La arquitectura cloud brinda mayor flexibilidad que otras arquitecturas.

- Servicios a demanda → el servicio se presta de acuerdo a lo solicitado.
  - Se pueden solicitar más o menos recursos en cualquier momento.
  - Una vez hecho el pedido, se los puede obtener inmediatamente.
  - No se es propietario de lo que se necesita, sino que es una suerte de alquiler → se usa algo brindado por alguien cuya capacidad es prácticamente infinita comparado con una organización individual.
  - Es una novedad que sea bajo demanda → antes era un problema que nos exigieran algo que no podíamos dar, generando así otros problemas (como la necesidad de ampliar el presupuesto o diferentes demoras producto de esas necesidades).
- Acceso único de red → varios sistemas pueden acceder al sistema mediante una única URL.
- Acceso amplio a través de la red → al servicio se accede a través de la red, por lo que la conectividad es esencial.
  - Sin conectividad, no hay servicio.
  - Para consumir el servicio como se desea, la capacidad de red debe ser muy buena.
- Pool de recursos → da la sensación de que los proveedores brindan servicios casi ilimitados.
- Conjunto compartido de recursos → los recursos se consumen de una manera distinta respecto de un *datacenter*.
  - La cantidad de recursos a compartir es extraordinariamente grande.
  - En un gran *datacenter*, cada recurso está identificado de manera física.
  - En *cloud*, los recursos están virtualizados y el cliente obtiene el equivalente a esos recursos → no se sabe dónde están ubicados físicamente esos recursos, ni interesa.
- Independencia de ubicación → se desconocen las ubicaciones de los proveedores.
- Elasticidad rápida → las capacidades pueden ser aprovisionadas y desplegadas casi de manera automática, permitiendo escalar rápidamente aumentando o disminuyendo según la demanda.
- Servicios medidos → se paga por lo que se consume, siempre de acuerdo a lo contratado.

## Cloud Privado vs Cloud Público

- Cloud Privado → arquitectura privada a nuestra organización, que no se vea para afuera.
- Cloud Público → arquitectura pública, para que sea compartida con el mundo.

## Cloud vs On\_Premise

La responsabilidad de mantener la calidad, en tanto seguridad, *performance* y disponibilidad, recaen sobre distintos actores:

- Arquitectura cloud → el responsable es el proveedor.
- Arquitectura on-premise → nosotros somos los responsables.
- Esquema Híbrido → combinación entre arquitecturas *on-premise* y *cloud*.

DataCenter On_Premise		Cloud
Nuestra.	<b>Responsabilidad</b>	Del proveedor.
Muy elevado.	<b>CapEx</b>	Casi nulo.
No está atado a la demanda, sino a la capacidad instalada.	<b>OpEx</b>	Dinámico, ajustado a la demanda.
Baja. Cualquier cambio implica altos costos y demoras en el tiempo.	<b>Flexibilidad para ampliar/disminuir capacidades</b>	Alta.
Manejo propio.	<b>Seguridad</b>	Imposible manejar toda la seguridad.
Manejo propio. Se evitan problemas de compatibilidad entre SWs.	<b>Actualizaciones de SW</b>	El SW está siempre actualizado, siempre se tiene la última versión. Pueden generar problemas de compatibilidad con otros SWs usados en la organización.
Si el acceso es local, es independiente de la conexión a Internet.	...	Usa estándares/normas internacionales que regulan los servicios prestados.

- Cuando la disponibilidad de los recursos es casi infinita, el enfoque (de los decisores de IT) para brindar soluciones cambia drásticamente → hay soluciones *cloud* que son impensables si se trabaja *on-premise*.
- No es fácil determinar los servicios *cloud* que se contratan → la sobrecontratación es un riesgo, ya que se puede terminar contratando servicios que no se necesitan.
- Para aquellos negocios que tienen perfectamente identificadas situaciones de concurrencia muy por encima de la media, los servicios de *cloud computing* son ideales.
- Desde el punto de vista del área de IT, es necesario no sólo mantenerse actualizado respecto de nuevas ofertas de servicios *cloud* sino también de los enfoques respecto de la utilización de esos recursos y del diseño de soluciones.

## Modelos de Despliegue

- Private Cloud → la infraestructura se proporciona para el uso de una sola organización.
- Community Cloud → la infraestructura se proporciona para la organización, proveedores y socios comerciales que forman una comunidad en su conjunto.
- Public Cloud → la infraestructura se proporciona para uso abierto, público en general.
- Hybrid Cloud → combinaciones de los modelos anteriores.

## Modelos de Servicio Cloud Ofrecidos

- SaaS · Software as a Service → el SW está instalado en la infraestructura del proveedor del servicio (no en las dependencias del cliente) y el cliente usa dicho SW desde afuera.
- PaaS · Platform as a Service → el cliente ya no contrata algo tan encapsulado como el SW a usar sino toda una plataforma (para desarrollo, despliegue y mantenimiento).
- IaaS · Infrastructure as a Service → el cliente contrata la infraestructura (sea procesamiento, almacenamiento, redes u otros recursos informáticos).
- CaaS · Container as a Service → el proveedor ofrece herramientas para construir y desplegar aplicaciones basadas en contenedores que resulten seguras y escalables.
- BPaaS · Business Process as a Service → el proveedor ofrece procesos de negocio, tales como gestión bancaria, publicidad, marketing, administración y finanzas y soporte a clientes.
- DBaaS · Data Base as a Service → el proveedor ofrece la utilización de DB, desentendiendo al usuario de detalles como la infraestructura subyacente, la instalación y las actualizaciones.
- FaaS · Function as a Service → el proveedor ofrece la ejecución de código en respuesta a eventos sin la infraestructura compleja típica de la construcción y despliegue de aplicaciones basadas en microservicios.
- BaaS · Blockchain as a Service → el proveedor ofrece infraestructura y herramientas al usuario para crear y mantener aplicaciones *blockchain*.

		On-Premise	IaaS	CaaS	PaaS	SaaS
Responsabilidad del Usuario	Uso					
	Aplicaciones					Aplicaciones
	Datos					Datos
	Ejecución				Ejecución	
	SO			SO		
	Virtualización	Virtualización				
	Servidores	Servidores				
	Almacenamiento	Almacenamiento				
	Redes	Redes				
		Responsabilidad del Proveedor				

# Arquitecturas Monolíticas

- La aplicación se construye como una unidad, un *todo*, una sola pieza, un monolito.
- Todos los módulos ejecutan dentro de un mismo proceso y sobre un mismo HW.
- Forma típica que adquieren:
  - Interfaz de Usuario del lado del cliente → formada por páginas HTML y código JavaScript que se ejecutan en el navegador web del cliente.
  - Base de Datos → generalmente es una BD relacional, pero puede ser otra.
  - Aplicación del lado del servidor → un único monolito que recibe solicitudes HTTP del cliente, ejecuta una lógica de negocio, recupera y actualiza datos de la BD y arma las vistas HTML que serán enviadas al navegador web.
- Los cambios en la aplicación están muy acoplados:
  - Todo cambio implica construir y desplegar una nueva versión de toda la aplicación.
  - Todo escalado requiere que escale toda la aplicación
  - Todo cambio condiciona la frecuencia de las entregas.
- Si bien pueden ser muy exitosas estas arquitecturas, suelen generar cierta frustración en los responsables, sobre todo a medida que la aplicación crece.

## Ventajas

- Fáciles de desarrollar.
- Fáciles de testear → no significa que sean fáciles de corregir.
- Fáciles de desplegar → no significa que las tareas previas al despliegue sean sencillas, porque el armado de dichas tareas puede llevar bastante tiempo.
- Fáciles de escalar horizontalmente, replicando en distintos servidores.

## Desventajas

- A medida que la aplicación crece:
  - Aumenta la complejidad del monolito → sigue siendo una sola pieza.
  - Aumenta la dificultad de mantener la modularidad inicial.
  - Aumenta la dificultad de incorporar nuevas tecnologías → es muy difícil cambiar la tecnología de un sector sin cambiar todo.
  - Disminuye la fiabilidad → un error en cualquier módulo puede potencialmente afectar toda la aplicación.
- El escalado es completo → se replica toda la aplicación.

## Arquitectura Monolítica vs Arquitectura de Microservicios (MSA)

Arquitectura Monolítica	Arquitectura de Microservicios (MSA)
Pone toda la funcionalidad en un solo proceso.	Pone cada funcionalidad en un servicio separado.
Escala replicando el monolito en múltiples servidores.	Escala distribuyendo estos servicios entre servidores, replicando según sea necesario.

# Arquitectura de Microservicios (MSA)

## MICROSERVICIOS (MSA)

Estilo de arquitectura en el que una aplicación se desarrolla como un conjunto de pequeños servicios que:

- Ejecutan en su propio proceso y se comunican con otros microservicios vía mecanismos ligeros.
- Se construyen en torno a capacidades de negocio.
- Pueden desplegarse de forma independiente mediante procesos automatizados.
- Poseen una mínima gestión centralizada.
- Pueden escribirse en diferentes lenguajes de programación y utilizar diferentes tecnologías de almacenamiento de datos.

## Ventajas de MSA

- Facilita el despliegue continuo de aplicaciones grandes y complejas → la aplicación grande y compleja en realidad son aplicaciones chicas combinadas entre sí.
- Facilita el mantenimiento.
- Facilita la incorporación de nuevas tecnologías.
- Permite el despliegue y escalado independiente.
- Permite trabajar con equipos de desarrollo autónomos.

## Desventajas de MSA

- Aumento significativo de la complejidad propia de un sistema distribuido.
- Requiere implementar mecanismos de comunicación entre servicios y el manejo de fallos.
- El *testing* de interacción es, en general, más complejo.
- Aumenta la complejidad de implementación, gestión y monitoreo.
- Se dificulta la detección de errores en tiempo de ejecución.
- Complejidad de la arquitectura de persistencia de datos particionada → son muy comunes las transacciones de negocio que requieren actualizaciones en repositorios pertenecientes a múltiples servicios.

## ¿Cuándo usar una MSA?

- Cuando el aprovisionamiento de infraestructura es rápido y, en lo posible, automatizado.
- Cuando hay herramientas adecuadas para el monitoreo correcto.
- Cuando hay mecanismos que permitan un despliegue rápido.

## ¿Cuándo NO usar una MSA?

- Si la aplicación es pequeña y no se espera que escale, **MSA** puede resultar una solución complicada y cara → una arquitectura monolítica es una solución mejor.

## Características principales de MSA

- **Componentización a través de servicios** → las MSA usan bibliotecas, pero su forma principal de crear componentes de su propio SW es segmentarlos en servicios.
  - **Componente** → unidad de SW independiente, se puede reemplazar y actualizar.
  - **Bibliotecas** → componentes que corren dentro del mismo proceso → están vinculadas a un determinado SW que se invocan por medio de llamadas a funciones en memoria.
  - **Servicios** → componentes que corren fuera del proceso.
    - Se comunican vía peticiones a *web services* o RPC, por ejemplo.
    - Que el servicio corra en un proceso aparte lo dota de independencia → si se cae el servicio, no se cae todo el monolito → no tiene un único punto de falla.
- **Organizada en torno a funcionalidades de negocio** → el hecho de partir las funcionalidades en servicios puede generar dudas.
  - Los cambios simples pueden llevar a que un proyecto entre equipos tome tiempo y requiera aprobación presupuestaria.
  - Los equipos suelen intentar optimizar su rendimiento introduciendo lógica en la parte sobre la que tienen control directo.
- **Productos, no proyectos:**
  - **Enfoque de Proyecto:**
    - Tiene un objetivo determinado.
    - Se desarrollará en un marco temporal (tendrá fechas de inicio y fin planificadas).
    - El objetivo es entregar una pieza de SW que se considerará terminada en algún momento.
    - Al finalizar, el SW se asignará a una organización de mantenimiento y el equipo que lo creó se disuelve.
  - **Enfoque de Producto** → *somos los padres de la criatura, debemos hacernos cargo.*
    - Un equipo debe poseer un producto durante toda su vida útil.
    - Si nosotros lo construimos, entonces nosotros nos encargamos de él.
    - El equipo de desarrollo tiene un conocimiento profundo del producto → hay una relación más intensa con el negocio, que en general da mejores soluciones.
    - Se concibe al SW en una relación continua en la que se busca cómo este puede ayudar a sus usuarios a mejorar sus capacidades en el negocio.
    - Mientras la solución está disponible, no termina: termina cuando la retiremos.
- **Smart endpoints and dumb pipes**
  - La lógica de la aplicación está en los servicios y no en los mecanismos de comunicación.
  - Los mensajes son coreografiados utilizando protocolos simples.



- **Gobierno descentralizado**

- Los microservicios no están obligados a ninguna estandarización de plataformas.
- Cada microservicio elige cuál es la tecnología que mejor se adapta a sus necesidades.

- **Gestión de Datos descentralizada**

- Descentralización del modelo conceptual → se segmenta un dominio complejo en múltiples dominios delimitados y mapea las relaciones entre ellos.
- Descentralización de las decisiones de persistencia de datos → cada servicio administra su propia DB.
- **Teorema de CAP** → en un sistema distribuido, donde las partes deben comunicarse entre sí, existen 3 características cuyo cumplimiento no se puede asegurar en forma permanente: puede ser que las 3 características se cumplan la mayor parte del tiempo, pero *no siempre; siempre* solamente 2:
  - [C] **Consistencia** → si distintos usuarios se conectan a distintas partes del sistema, todos deben obtener la misma lectura.
  - [A] **Disponibilidad** → si un usuario se conecta al sistema, siempre obtiene una respuesta, aunque no se garantiza que esa respuesta esté actualizada.
  - [P] **Tolerancia al Particionamiento** → el sistema sigue funcionando incluso si ha sufrido particiones (por lo general, producto de una catástrofe).

Una buena y muy utilizada estrategia, aprovechando que las particiones son muy poco frecuentes, es asegurarse la consistencia y la disponibilidad permanentes. Respecto de la tolerancia al particionamiento, podemos prepararnos para, si sucede lo inevitable (una catástrofe que particione al sistema), recuperarnos rápidamente:

- Pueden generarse particiones, ya que son inevitables.
- Se tienen mecanismos para detectar tempranamente las particiones y, una vez recuperada la conectividad, se trabaja para corregir rápidamente las consecuencias de las particiones (un *merge* entre las dos particiones).

- **Automatización de infraestructura.**

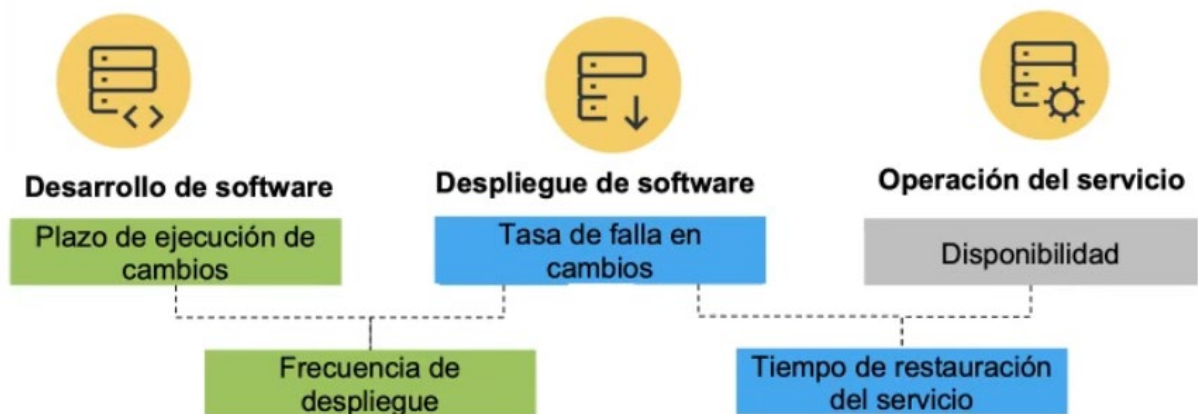
- **Diseño tolerante a fallas** → se preparan soluciones que contemplan escenarios de fallas para lidiar con ellas y mantenerse operativo.

- Las comunicaciones sobre redes son, por naturaleza, poco confiables.
- Las aplicaciones se diseñan para ser resilientes y manejar errores, no solamente para prevenirlos.
- Monitoreo en tiempo real de la aplicación, tanto de elementos de arquitectura como de métricas relevantes del negocio.

- **Diseño evolutivo:**

- Los servicios evolucionan buscando reducir al mínimo el impacto de los cambios en sus consumidores.
- Los servicios se diseñan buscando el mínimo acoplamiento posible al contrato de sus proveedores.
- El uso de servicios como componentes posibilita planeamientos de despliegues más granulares.

## Métricas de MSA



- Plazo de Ejecución de Cambios → mide el tiempo que va desde que empieza el análisis de los requerimientos hasta que todo se despliega y quede disponible para su uso.
- Tasa de Falla en Cambios → mide cuántos despliegues son exitosos respecto del total de despliegues realizados.
- Disponibilidad → mide qué tan disponible está un servicio.
- Frecuencia de Despliegue → lo ideal es que se las entregas se realicen muy seguido (varias veces por día), con una muy baja tasa de errores.
  - Es la combinación entre el plazo de ejecución de cambios y la tasa de falla en cambios.
- Tiempo de Restauración del Servicio → mide el tiempo que se tarda en recuperar el servicio, en volverlo a disponibilizar.
  - Es la combinación entre la tasa de falla en cambios y la disponibilidad.

### Aspectos de la performance de la entrega de software:

	ÉLITE	ALTO	MEDIO	BAJO
Frecuencia de Despliegue	Bajo Demanda. Varios despliegues por día.	1 vez por día ~ 1 vez por semana	1 vez por semana ~ 1 vez por mes	1 vez por mes ~ 1 vez por semestre
Plazo de Ejecución de Cambios	Menos de 1 día	1 día ~ 1 semana	1 semana ~ 1 mes	1 mes ~ 6 meses
Tiempo de Restauración de Servicio	Menos de 1 hora	Menos de 1 día	Menos de 1 día	1 semana ~ 1 mes
Tasa de Falla en Cambios	0% ~ 15%	0% ~ 15%	0% ~ 15%	46% ~ 60%

# Procesamiento de Datos

## Infraestructura de Procesamiento de Datos – Lo que se espera de ella

- **Confiabilidad** → dada por la estabilidad del sistema.
  - Refiere a que, al ponerla en funcionamiento, responda de manera exitosa.
- **Rendimiento** → predictibilidad del tiempo de obtención del resultado.
- **Sustentabilidad Económica** → que se tenga presupuesto suficiente.
  - Nuestra astucia será cuánto más podemos hacer con el presupuesto que hay.
  - Es válido solicitar aumentos de presupuesto (también es parte de la buena gestión).

## Unidades de Procesamiento de Datos – Lo que se espera de ella

- **Confiabilidad.**
- **Disponibilidad** → que pueda funcionar de manera continua y regular.
  - Un sistema no sirve de nada que no está en condiciones de ser utilizado.
- **Tolerancia a Fallas** → que, ante fallas, el servicio siga operando y no se vea afectado.
- **Escalabilidad** → capacidad de agregar/reducir recursos ante un cambio en la demanda.
  - Hay que estar preparados para cuando aumenta la demanda.
  - La idea es gastar en función de lo que se necesita.
- **Compatibilidad** entre el HW y los controladores disponibles.
- **Administración Remota.**
- **Mantenimiento En Caliente** → que, ante mantenimientos, el sistema siga funcionando.
  - Los sistemas de más alta disponibilidad siguen funcionando cuando se mantienen, aun si esos mantenimientos no son planificados.

## Mainframe vs Supercomputadora

	Mainframe	Supercomputadora
<b>Descripción*</b> <b>Funciones Básicas</b> <b>y Principio de Trabajo</b>	<ul style="list-style-type: none"><li>• Servidor centralizado.</li><li>• De propósito general.</li><li>• Almacena grandes DB.</li><li>• Atiende miles de usuarios en forma simultánea.</li><li>• Focalizada en la performance de las DB masivas.</li><li>• Ataca problemas limitados por la confiabilidad.</li></ul>	<ul style="list-style-type: none"><li>• Posee gran cantidad de núcleos.</li><li>• Su fortaleza reside en la capacidad de cálculos complejos en punto flotante.</li><li>• Para procesar semejante cantidad de datos, se requieren suficientes memoria y almacenamiento.</li><li>• Ataca problemas limitados por la capacidad de cálculo.</li><li>• Son <i>trajes a medida</i>.</li></ul>
<b>Velocidad</b>	Millones de instrucciones por segundo.	Miles de millones de operaciones en punto flotante por segundo.
<b>Usos</b>	-	Ciencia, Industria y Defensa.

# Virtualización, Hiperconvergencia, MVs, Contenedores, Grid Computing

## VIRTUALIZACIÓN

Recursos HW y SW que se muestran como SW, brindándole al usuario una vista distinta de la realidad subyacente.

- **Aplicaciones** → servidores virtuales, aplicaciones de seguridad, almacenamiento distribuido y/o remoto, HW de redes, redes virtuales, sistemas de *cluster* de servicios.
- **¿Por qué es importante?**
  - Optimiza el uso de recursos.
  - Aumenta la velocidad de despliegue muy significativamente.
  - Aumenta la disponibilidad de los servicios.
  - Disminuye tiempos de parada por mantenimiento de HW.
  - Permite delegar la gestión de los recursos.

## HIPERCONVERGENCIA

Servidores que vienen con una capa de SW que permiten administrar los recursos HW.

## MÁQUINA VIRTUAL (VM)

Computadora que tiene su propio sistema operativo y sus propias aplicaciones.

- Sobre la infraestructura corre el sistema operativo y sobre él se monta el SW de virtualización (un hipervisor), el cual permite administrar las distintas VMs.
- La infraestructura de la VM está virtualizada.

## CONTENEDOR

SW de virtualización que permite administrar pequeñas aplicaciones dentro del sistema operativo sobre el cual está montado el SW en cuestión.

- Sobre la infraestructura corre el sistema operativo y sobre él se monta el SW de virtualización (un contenedor).
- No siempre se necesitan → en general, complementan las VMs.

## GRID COMPUTING

Cluster de uso general donde los recursos utilizados, que son ociosos (o residuales) y distantes (están distribuidos geográficamente, no es que están en el mismo lugar), son administrados por un SW de virtualización.

- Se parece a la hiperconvergencia → se tienen recursos en equipos distintos y se busca que sean vistos de otra manera, no necesariamente “como uno solo”.

## Clusters

### CLUSTER

Conjunto de recursos donde cada uno es un nodo, evitando así un punto único de falla que aparece al trabajar con un único nodo:

- Una falla en ese único componente desencadena una falla en el sistema completo.
- El sistema no puede funcionar sin ese único componente.

### [HA-C] CLUSTER DE ALTA DISPONIBILIDAD

Sistema que aumenta la disponibilidad del servicio ofrecido a través de la redundancia de nodos agrupados.

- Garantiza la disponibilidad del servicio mientras exista al menos 1 nodo operativo.
  - Todos los nodos hacen exactamente lo mismo.
  - Hay un SW de virtualización que orquesta los nodos.
- Si un nodo falla, el sistema de gestión del *cluster* transfiere el servicio activo a otro nodo.
- El nivel de redundancia está dado por la cantidad de nodos.
- El nivel de redundancia determinará la cantidad de fallas simultáneas admisibles sin pérdida de servicio.
- La implementación es compleja.

### [LB-C] CLUSTER DE BALANCEO DE CARGA

Sistema que provee escalabilidad basada en la distribución de la carga de trabajo entre los nodos activos del sistema.

- El balanceador de carga distribuye el trabajo entre los distintos servidores de aplicación.
  - Los servidores no hacen lo mismo → no atienden las mismas peticiones.
- Se busca gestionar la capacidad → la idea es que los servidores reciban una carga pareja, más allá de la capacidad de cada servidor.
- Si un servidor de aplicación falla, el resto de los servidores debe absorber el trabajo.
- Si en un momento crítico (servidores trabajando al 100% habiendo peticiones en espera) se agrega otro servidor, aumenta la *performance*.
- Al trabajar con un único **LB-C**, hay un punto único de falla → la solución es agregar un segundo **LB-C**.

### [HP-C] CLUSTER DE ALTA PERFORMANCE

Sistema pensado específicamente para explotar el potencial del procesamiento en paralelo entre múltiples computadoras.

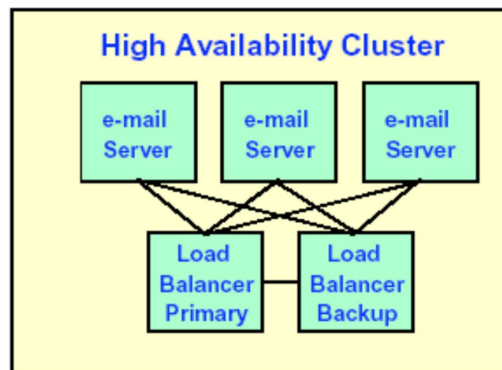
- Parte cada petición en “subpeticiones” y distribuye éstas entre los servidores.
- Los servidores trabajan en paralelo cada uno atendiendo su subpetición.

## Clusters Combinados – Ejemplo: LB-C y HA-C sin punto único de falla

Para evitar que el balanceador de carga sea un punto único de falla, se agrega un segundo balanceador de carga.

- Si se cae un balanceador de carga, el sistema sigue funcionando (gracias al backup).
- Ahora, no le pase nada a ese porque si ése se cae, el sistema deja de funcionar.

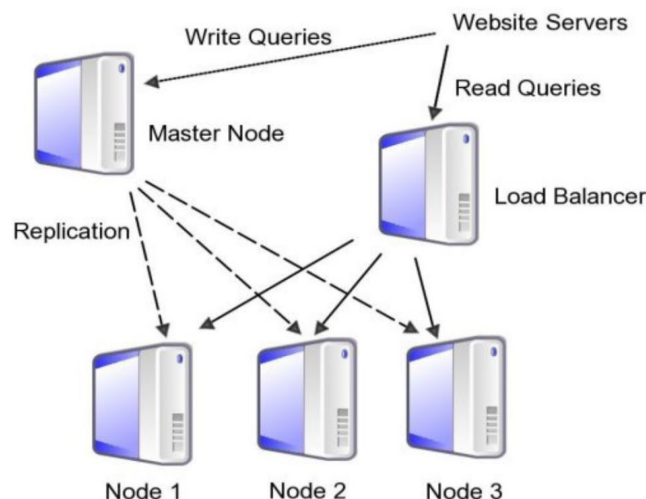
En el ejemplo se tienen 2 balanceadores de carga (*load balancer primary* y *load balancer backup*), los cuales distribuyen el trabajo entre servidores de mail:



## Clusters Combinados – Ejemplo: HP-C y los tipos de peticiones

En el ejemplo se tiene una granja de servidores que recibe solicitudes y las separa en solicitudes de escritura en DB (un INSERT, un UPDATE o un DELETE) y solicitudes de lectura (un SELECT).

Las solicitudes se dividen para evitar un desbalanceo de carga entre las cantidades de una y las de otra, de manera que no se llegue a un cuello de botella.



Si se reciben solicitudes de escritura (léase *write queries*), se actualiza la DB. Luego, el motor de la DB (léase *master node*) replica en otros 3 servidores (léase: *node 1*, *node 2* y *node 3*).

Si se reciben solicitudes de lectura (léase *read queries*), las toma el balanceador de carga y éste las distribuye entre los 3 servidores (léase: *node 1*, *node 2* y *node 3*).

# Persistencia de Datos

## PERSISTENCIA DE DATOS

Capacidad de almacenar cualquier tipo de información de manera que perdure en el tiempo.

- Persistencia Volátil → los datos no necesitan ser almacenados tras su procesamiento.
- Persistencia NO Volátil → los datos deben perdurar tras su procesamiento.
- Persistencia Políglota → varias formas de persistencia en una misma solución.

En base de datos, una transacción es un conjunto de instrucciones que se ejecutan como una unidad de trabajo, en forma atómica, de manera indivisible.

Toda transacción debe cumplir con las propiedades ACID...

## Propiedades ACID

- [A] Atomicidad → las transacciones se ejecutan en forma atómica, como un todo.
- [C] Consistencia → las transacciones no van a “romper” la BD.
- [I] Aislamiento → se pueden definir distintos niveles de aislamiento (*isolation levels*) para cada transacción, independientemente de cada una.
- [D] Durabilidad → si la transacción se ejecuta correctamente, persiste.

## Sistemas de Persistencia Volátil

- Sistema de Caché → memoria para el almacenamiento de información de rápido acceso.
  - Necesita inteligencia para determinar, en base a información estadística, cuáles son aquellas cosas que más utilizarán en el futuro, para que luego se puedan guardar allí.
  - Si algo se usa muy seguido, se trae a la *caché* y la próxima vez que se lo requiera, no se lo irá a buscar a disco sino a la *caché*.
- Sistema MemCaché → auxiliar del código que tiene que almacenar información.
  - Ubicado detrás de los servidores, a la par de la DB y de los discos.
  - Las decisiones de uso de caché son tomadas por el servidor.
- Sistema Varnish → pensado para servidores web cuyo objetivo es cachear contenido de uso frecuente.
  - Ubicado entre los servidores y el balanceador de carga.
  - Las decisiones de uso de caché son tomadas por el propio sistema Varnish.
- Sistema REDIS (Remote DIctionary Server) → esquema de almacenamiento volátil que tiene la opción de persistir en forma no volátil, usando un esquema no relacional: *key-value*.
  - Puede tener un esquema tolerante a fallas, con un *cluster* de esquema *master-slave*, donde la replicación de los nodos *masters* en los nodos *slave* es asincrónica.
  - Es extremadamente veloz → el tiempo de respuesta es muy bajo.
  - Puede escalar bastante, llegando a atender millones de solicitudes por segundo.

## Sistemas de Persistencia NO Volátil

- **Datos Estructurados** → guardan o respetan una estructura de datos que permite, más allá del posicionamiento físico, almacenarlos o recuperarlos de manera predefinida.
  - **Lenguaje SQL** → lenguaje estructurado de tratamiento de datos para interactuar con motores de DB relacionales.
- **DB SQL** → DB que implementan modelos relacionales estrictos con el objetivo de garantizar la consistencia de los datos a partir de relaciones.
  - Son las DB más maduras, dada su antigüedad y extensa utilización.
- **DB NoSQL** → DB cuyo modelo no busca garantizar la consistencia de los datos a partir de relaciones, sino que tienen por objetivo soportar modelos flexibles que no requieran estructuras rígidas propias del modelo relacional.
  - Hay varios esquemas, como: clave-valor (*key-value*), familia de columnas, basadas en documentos, basadas en grafos, etcétera.
  - Muchas de ellas están preparadas para sistemas distribuidos.
  - **Elastic Search** → motor de búsqueda y análisis de documentos (a partir de relaciones de términos) y, a la vez, DB (ya que recibe información, la almacena y la indexa):
    - Monitorea (*logs*, infraestructura en esquemas *cloud*) en tiempo real.
    - Es sumamente rápido.
- **Persistencia de Objetos** → modelo de persistencia distribuida de archivos implementado generalmente en servicios de *cloud* pública.
  - Tiene gran escalabilidad.
  - *Ejemplo: Amazon S3 (Amazon Simple Storage Service).*
- **Persistencia de Archivos Distribuidos** → sistema distribuido de archivos para manejo de grandes volúmenes de datos.
  - Son de rápido acceso.
  - Tienen alta disponibilidad → tienen un alto nivel de redundancia.
  - *Ejemplo: Apache Hadoop, el cual es un framework de procesamiento y almacenamiento de información.*
- **CDN (Content Delivery Network)** → sistema distribuido y escalable de entrega de contenidos basado en minimizar el costo de red entre el punto de distribución y el usuario.
  - En Internet, dependiendo de la ubicación del usuario, la *performance* de las aplicaciones suele ser pobre (sobre todo aquellas que demandan un gran ancho de banda) → el usuario recibe un servicio que no es bueno.

Una solución a eso son los CDN → una suerte de memoria caché ubicada entre los usuarios e Internet que almacena la información que más frecuentemente utilizan los mencionados usuarios.
  - Cuando un usuario realiza una solicitud al servidor original, esa petición se delega a otro servidor más cercano que puede ofrecer mejor *performance* que el original.



## Sistemas de Persistencia Políglota

- Key-Value para funcionalidades de carrito de compra e inicio de sesión.
- DB relacionales para funcionalidades que requieren transacciones.
- DB con grafos para necesidades de navegación entre distintos conceptos.

Aplica a **MSA** porque puede suceder que ...

- ... varios microservicios usen un mismo tipo de persistencia, compartiendo una misma fuente de datos.
- ... un mismo microservicio use varios tipos de persistencia, requiriendo múltiples fuentes de datos.
- ... varios microservicios utilicen varios tipos de persistencia, requiriendo múltiples fuentes de datos → es común que se requiera una capa de persistencia políglota.

## Equipos y Sistemas de Almacenamiento: DAS, SAN y NAS

- DAS (Direct Attached Storage) → método tradicional donde se conecta el dispositivo de almacenamiento directamente al servidor (PC), como si fuera un disco duro externo.
  - Las aplicaciones hacen las peticiones de datos directamente al sistema de archivos.
  - El acceso a los archivos es a bajo nivel → a nivel de bloque de datos.
  - VENTAJA → es la menos costosa.  
VENTAJA → es la que tiene las mayores velocidades de transmisión de información.
  - DESVENTAJA → la escalabilidad es limitada → sólo se puede acceder al dispositivo de almacenamiento desde la PC al que está físicamente conectado. Al ser un recurso compartido, quienes quieran usarlo deben ponerse de acuerdo.
- SAN (Storage Area Network) → conjunto de dispositivos que crean una red enfocada en el intercambio de datos a través de una red de alta velocidad, equipos de interconexión (como *switches*) y discos duros donde almacenar los datos.
  - Las aplicaciones hacen las peticiones de datos directamente al sistema de archivos.
  - El acceso a los archivos es a bajo nivel → a nivel de bloque.
  - VENTAJA → muy altas velocidades de transmisión de información.  
VENTAJA → muy buen rendimiento.
  - DESVENTAJA → más fáciles de administrar.  
DESVENTAJA → altos costos.
- NAS (Network Attached Storage) → método de almacenamiento dedicado para compartir información entre servidores o PCs dentro de una red local o Internet (vía HTTP).
  - Se puede trabajar con la información desde varios equipos en forma simultánea.
  - Las aplicaciones hacen las peticiones de datos a los sistemas de archivos de manera remota y el almacenamiento es local al sistema de archivos
  - El acceso a los archivos es a alto nivel → a nivel de archivo.
  - VENTAJA → de uso sencillo y bajo costo.
  - DESVENTAJA → no tiene la performance de una SAN.

## Disponibilidad de los Datos

- **Disponibilidad** → que la información esté accesible para quien la quiera utilizar. Un dispositivo ofrece disponibilidad si los usuarios pueden acceder al mismo y el dispositivo funciona de acuerdo a lo esperado. Cualquier acción que altere eso afecta la disponibilidad.

Elementos que aumentan la disponibilidad:

- Redundancia de fuentes.
  - Redundancia de controladores.
  - Redundancia de *switches*.
  - Redundancia de discos → sistema RAID.
- **Performance** → métrica usada para medir la velocidad de un sistema de almacenamiento, como la cantidad de información transmitida/recibida por unidad de tiempo (la más común) o el tiempo de respuesta.
  - **Dispositivos de Almacenamiento Offline** → muy útiles para la restauración.

Características:

- Integridad → reemplazan la información que sí está *online*.
- La idea es mantener a estos dispositivos en un lugar distinto de donde están los entornos productivos → si sucede un desastre masivo, estarán a salvo.

Un ejemplo son las cintas magnéticas, donde:

- La lectura y la escritura son secuenciales → si solamente nos preocupa grabar (escribir), es muy rápido; si necesitamos ver las grabaciones (leer), no.
- Pueden ser de operación manual, semiautomática o automática.
- Hay “cintas magnéticas virtuales”, que en realidad son discos.

# Seguridad en la Información

La información es un activo, algo valioso para una organización → debe protegerse adecuadamente.

## SEGURIDAD DE LA INFORMACIÓN

Refiere a las medidas preventivas y reactivas de los sistemas tecnológicos y las organizaciones que tienen como objetivo mantener 3 pilares:

- **[C] Confidencialidad** → que la información la pueda ver sólo *quien y como* corresponda.
- **[I] Integridad** → que la información se mantenga igual si no se realizan cambios; y si se hacen, deben realizarse aplicando ciertas reglas.
- **[A] Disponibilidad** → la información debe estar disponible (según lo establecido en el SLA) para aquellas personas autorizadas a acceder a esa información, sin ninguna interrupción.

## Sistemas y Dispositivos

- **Firewall** → dispositivo de seguridad que regula el tráfico entrante y saliente. En base a ciertas reglas, decide si conceder/denegar cierto tráfico específico.
  - Permite tener redes internas seguras, controladas y con cierto grado de confianza.
  - Puede ser un dispositivo que viene con un SW embebido o bien un servicio (SaaS) dentro de una nube pública o privada.
- **IDS (Sistema de Detección de Intrusos)**
  - Analiza lo que atravesó el firewall y, en base a las reglas que tiene almacenada, analiza los paquetes y eventualmente levanta alertas → no acciona, sólo avisa.
  - Ubicado detrás del firewall, conectado al switch de entrada/salida de la red
- **IPS (Sistema de Protección de Intrusos)**
  - A diferencia del IDS, el IPS sí toma acciones, pudiendo filtrar paquetes.
  - Ubicado entre el firewall y el switch de entrada/salida de la red.
- **WAF (Web Application Firewall)**
  - Similar al *firewall*, pero con información específica de la capa de aplicación (OSI).
- **VPN (Red Privada Virtual)** → red privada dentro de la red pública.
  - La comunicación es segura → las direcciones IP de origen se disfrazan.
  - Un SW cliente VPN evita fallas de seguridad en los accesos e impide utilizar ciertas herramientas que pueden poner en riesgo la seguridad de toda una organización.
  - Se usa en trabajo remoto → el acceso a redes internas de una organización es seguro.
- **SIEM** → sistemas que reúnen gestión de seguridad en la información con gestión de eventos.
  - Una parte analiza la ocurrencia de ciertos eventos y la otra analiza cuáles de ellos pueden representar algo significativo respecto de la seguridad en la información.
  - Monitorea (detecta) y, de acuerdo a lo que examina, actúa (protege)
- **Actualizaciones de Seguridad** → cualquier sistema (firmwares, SO, DB, aplicaciones, etc.) puede presentar fallos de seguridad, por lo que para evitar que cualquier atacante explote esas vulnerabilidades resulta clave mantener actualizados a todos los sistemas involucrados.

## Conceptos relacionados

- **Evento** → ocurrencia identificada que indica una posible violación de la política de seguridad, pudiendo ser relevante para la seguridad de la información.
- **Incidente** → evento o serie de eventos de seguridad de la información inesperados o no deseados que tiene/n una probabilidad significativa de afectar al negocio.
- **Identidad** → refiere a asegurar que una persona es quien dice ser que es.
- **Autenticación** → refiere a los permisos que tiene una persona (a sabiendas de quién es).

## Gestión de la Seguridad de la Información (SGSI)

La **gestión de la seguridad de la información (SGSI)** es un proceso continuo que busca establecer y mantener (a lo largo del tiempo) programas, controles y políticas que tengan como finalidad conservar la confidencialidad, integridad y disponibilidad de la información.

## Plan de Respuesta a incidentes de seguridad

- **Fases:**
  - Acción inmediata para detener o minimizar su impacto.
  - Investigación temprana del incidente.
  - Restauración de los recursos afectados.
  - Reporte del incidente por los canales apropiados.
- **Componentes:**
  - Equipo de expertos en seguridad.
  - Estrategia legal revisada y aprobada, referida a obligaciones.
  - Soporte financiero de la organización.
  - Soporte ejecutivo de la gerencia superior de la compañía o áreas afectadas.
  - Recursos físicos.

## Seguridad de Datos – Acciones y Herramientas

- Análisis de vulnerabilidades en códigos fuente y aplicaciones.
- Separación de ambientes de desarrollo, de *testing*, de preproducción y de producción.
- Tests → *tests* unitarios, *tests* de integración, *tests* de regresión, etc.
- Control y auditoría de acceso mediante registros ocultos, para que nadie los altere.

## Seguridad Física

- *Backups* → administración de respaldos de información.
- Disponibilidad en sí misma.
- Gestión de centros de cómputos principales y secundarios.

## Seguridad Lógica

- Restringir el acceso a los programas y archivos.
- Asegurar que los usuarios tengan todo lo que necesitan para trabajar y sólo lo que necesitan.
- Que la información transmitida sea recibida solamente por el destinatario deseado.
- Que la información recibida sea la misma que ha sido enviada.
- Que existan canales alternativos secundarios de transmisión entre diferentes puntos.
- NO Repudio:
  - NO Repudio de Origen → asegurarse de que el mensaje fue enviado por quien dice ser el emisor.
  - NO Repudio de Destino → asegurarse de que el mensaje fue recibido por quien dice ser el receptor.
- Tipos de Usuario:
  - Propietario.
  - Administrador.
  - Usuario principal.
  - Usuario de explotación.
  - Usuario de auditoría.

## Norma ISO/IEC 27000

La norma ISO/IEC 27000 es una familia de normas orientadas a seguridad de la información.

- ISO/IEC 27001 → 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 → guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información.
- ISO/IEC 27005 → guía de gestión de riesgos específica para seguridad de la información.
- En otros puntos se hace referencia a:
  - Requisitos de la empresa para el control de acceso.
  - Controles criptográficos.
  - Áreas seguras → controles de acceso físico.
  - Procedimientos y responsabilidades operativas, protección contra malware, respaldo, registros y monitoreo, control del SW en producción, gestión de vulnerabilidades técnicas.

## Norma ISO/IEC 31000

La norma ISO/IEC 31000 es una familia de normas que aplican un enfoque de gestión de la seguridad de la información basado en riesgos.

- **Amenaza** → posible causa de un incidente no deseado que puede dañar un sistema.
- **Vulnerabilidad** → debilidad de un activo que puede ser explotada por una/s amenaza/s.
- **Activo**<sup>11</sup> → información o bien infraestructura o personal que contienen información.
- **Riesgo** → exposición a la amenaza → efecto de la incertidumbre sobre los objetivos.
- **Control** → medidas que pueden tomarse para atenuar o eliminar el riesgo.
- **Impacto** → resultado de la materialización del riesgo.

*Cuando las vulnerabilidades de activos o controles quedan expuestas a una o más amenazas, se genera un riesgo el cual, si se materializa, genera un impacto.*

Ejemplo de amenazas, vulnerabilidades y activos:

	EJEMPLO0	EJEMPLO1	EJEMPLO2
<b>Amenaza</b>	Falla del sistema por sobrecalentamiento de la sala de servidores. [ALTA]	Interferencia humana maliciosa mediante denegación de servicio (DDoS). [ALTA]	Acción humana accidental: se borraron archivos. [ALTA]
<b>Vulnerabilidad</b>	El sistema de aire acondicionado tiene 10 años de antigüedad. [ALTA]	El firewall tiene una configuración adecuada y buena mitigación de DDoS. [BAJA]	Los permisos están adecuadamente confirmados, se realizan regularmente auditorías de SW y respaldo de datos. [BAJA]
<b>Activo</b>	Servidores ubicados en la sala. [CRÍTICO]	Sitio web. [CRÍTICO]	Archivos en un repositorio compartido. [MEDIO]
<b>Impacto</b>	Todos los servicios no estarán disponibles durante, al menos, 3 horas. [CRÍTICO]	Los recursos del sitio web no estarán disponibles. [ALTO]	Los datos podrían perderse, pero casi con seguridad se podrían recuperar de un respaldo. [BAJO]
<b>Probabilidad de Ocurrencia</b>	La última falla del sistema ocurrió el mes pasado. [ALTA]	Se detectó una sola DDoS en los últimos 2 años. [MEDIA]	[MEDIA]
<b>Riesgo</b>	Pérdida de \$30M. (ALTO)	Pérdida de \$3M por cada hora caída. (MEDIO)	(BAJO)
<b>Recomendación de Control</b>	Comprar un nuevo sistema de aire acondicionado por un costo de \$1.000.000.	Monitorear el <i>firewall</i> .	Continuar el monitoreo de cambios de permisos de usuarios y respaldos.

<sup>11</sup> No definido por la norma.

## Otras normas

- **COBIT 5** → normativa más general, centrada en la gestión de servicios de IT que incluye aspectos de seguridad de información.
  - Alinea la seguridad de la información con los objetivos de la organización.
- **ITIL** → normativa más general, centrada en la gestión de servicios de IT que incluye aspectos de seguridad de información.
- **PCI** → se preocupa de proteger los datos sensibles del poseedor de la tarjeta de crédito, principalmente la confidencialidad y la integridad.

## Ciberseguridad

Los ciberataques son considerados uno de los principales riesgos dentro del mundo de los negocios digitales. El aumento de las comunicaciones hace más probable la ocurrencia de ciberataques.

### Herramientas y debilidades:

- **Inteligencia Artificial** → manipulación tendenciosa a través de *fake news* y *deepfakes*.
- **Tecnología Móvil de 5<sup>ta</sup> generación (5G)** → deberá modernizarse la infraestructura tecnológica, para poder combatir el déficit de cobertura, seguridad y confiabilidad.
- **Computación Cuántica** → podría reducir drásticamente el tiempo necesario para resolver problemas matemáticos en los que actualmente se apoyan las técnicas de cifrado.
- **Computación en la nube** → el hecho de que las empresas vuelquen en *cloud* cada vez más información personal crea potenciales riesgos a la privacidad y la seguridad de los datos.

**Servicios de Autenticación** → se le delega la responsabilidad de autenticación a otros servicios.

## Firma Electrónica y Firma Digital

### **FIRMA DIGITAL**

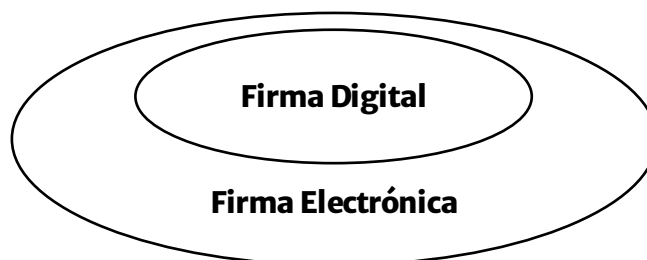
Resultado de aplicar a un documento digital un procedimiento matemático (vía criptografía) que requiere información de conocimiento exclusivo del firmante.

- Se basa en esquemas de clave pública y clave privada.

### **FIRMA ELECTRÓNICA**

Concepto legal que apunta a asegurar la identidad de alguien.

- Una forma de implementarla es con una firma digital.



*Todas las firmas digitales son firmas electrónicas,  
pero no todas las firmas electrónicas son firmas digitales.*

# Redes de Datos

## Importancia

- Nos permiten intercambiar datos accediendo a sistemas remotos.
- La disponibilidad de acceso a los sistemas informáticos remotos depende de la conectividad. Si no se puede acceder al servicio, es como si el servicio no existiera.
- La disponibilidad de las redes depende de medios físicos (cables, antenas, microondas y activos de red) y lógicos (protocolos y accesos).

## Servicios Disponibles de Conectividad

- Enlaces de acceso a Internet.
- Enlaces punto-a-punto (P2P), como LAN o VPN.
- Enlaces punto-a-multipunto.
- MPLS (Multi-Protocol Label Switching) → sistema que permite mejorar la eficiencia de las redes acelerando el encaminamiento de los paquetes de datos, combinando ventajas tanto del control de enrutamiento como de la conmutación rápida.
  - Es ideal para conectar sucursales.
- Enlaces por Fibra Óptica → el más eficiente, con altísimas velocidades de transmisión.
  - No está libre de incidentes → para evitar cortes de cables, en lugar de hacer un cableado subterráneo, se hacen tendidos.
- Enlace por Radiofrecuencia → permite unir dos puntos con línea de vista distantes a muchos kilómetros sin necesidad de cables, aunque utilizando torres y antenas.
  - No está libre de inclemencias climáticas.
- Enlace Satelital → llega prácticamente a cualquier locación, aunque no son buenos los tiempos de respuesta.
  - No está libre de inclemencias climáticas.

## Alta Disponibilidad en Redes

La alta disponibilidad se logra con varios nodos en *cluster* donde, ante una caída del nodo principal, un nodo secundario lo reemplaza y realiza la misma tarea que hacía el nodo principal. La disponibilidad prácticamente no se ve afectada porque se evita el punto único de falla.

La alta disponibilidad se puede aplicar en servidores, balanceadores de carga (*load balancers*) y distintos dispositivos de red (como *switches*, por ejemplo).