# Arquitectura de Software
# Conceptos útiles para diagramas

## 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, sino que se "alquila" 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 provistas y desplegadas casi de manera automática, pudiendo 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. | **Responsabilidad** | Del proveedor. |
| Muy elevado. | **CapEx** | Casi nulo. |
| No está atado a la demanda, sino a la capacidad instalada. | **OpEx** | Dinámico, ajustado a la demanda. |
| Baja. Cualquier cambio implica altos costos y demoras en el tiempo. | **Flexibilidad para ampliar/disminuir capacidades** | Alta. |
| Manejo propio. | **Seguridad** | Imposible manejar toda la seguridad. |
| Manejo propio. Se evitan problemas de compatibilidad entre SWs. | **Actualizaciones de SW** | 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 · *Infraestructure 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*.

<table>
<tr><th></th><th>On-Premise</th><th>IaaS</th><th>CaaS</th><th>PaaS</th><th>SaaS</th><th></th></tr>
<tr><td rowspan="10">Responsabilidad del Usuario</td><td colspan="5">Uso</td><td rowspan="10">Responsabilidad del Proveedor</td></tr>
<tr><td colspan="4">Aplicaciones</td><td>Aplicaciones</td></tr>
<tr><td colspan="4">Datos</td><td>Datos</td></tr>
<tr><td colspan="3">Ejecución</td><td colspan="2">Ejecución</td></tr>
<tr><td colspan="2">SO</td><td colspan="3">SO</td></tr>
<tr><td>Virtualización</td><td colspan="4">Virtualización</td></tr>
<tr><td>Servidores</td><td colspan="4">Servidores</td></tr>
<tr><td>Almacenamiento</td><td colspan="4">Almacenamiento</td></tr>
<tr><td>Redes</td><td colspan="4">Redes</td></tr>
</table>

# 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** |
|---|---|---|
| **Descripción* Funciones Básicas y Principio de Trabajo** | • Servidor centralizado.<br>• De propósito general.<br>• Almacena grandes DB.<br>• Atiende miles de usuarios en forma simultánea.<br>• Focalizada en la performance de las DB masivas.<br>• Ataca problemas limitados por la confiabilidad. | • Posee gran cantidad de núcleos.<br>• Su fortaleza reside en la capacidad de cálculos complejos en punto flotante.<br>• Para procesar semejante cantidad de datos, se requieren suficientes memoria y almacenamiento.<br>• Ataca problemas limitados por la capacidad de cálculo.<br>• Son *trajes a medida*. |
| **Velocidad** | Millones de instrucciones por segundo. | Miles de millones de operaciones en punto flotante por segundo. |
| **Usos** | - | 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.**

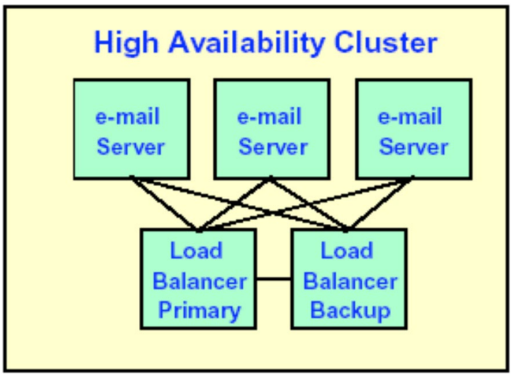
- Parte cada petición en "subpeticiones" y distribuye éstas entre los servidores.
- Los servidores trabajan en paralelo cada uno atendiendo su subpetición.

## 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.

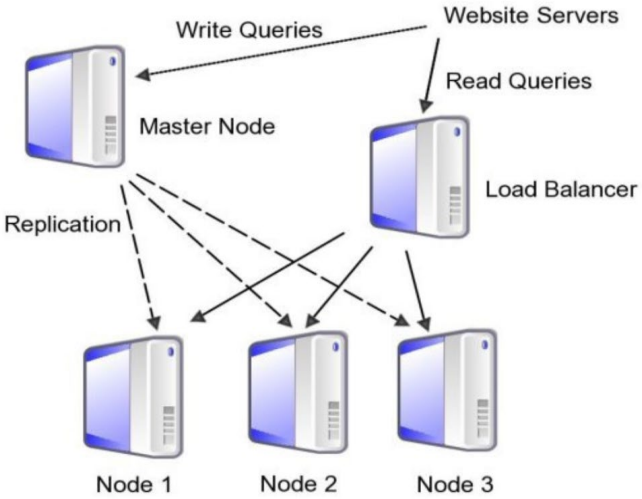
En el ejemplo se tienen 2 balanceadores de carga (*load balancer primary* y *load balancer backup*), los cuales distribuyen el trabajo entre servidores de mail:




## 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 disrupción.

## 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**[1] → 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:*

| | **EJEMPLO 0** | **EJEMPLO 1** | **EJEMPLO 2** |
|---|---|---|---|
| **Amenaza** | 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] |
| **Vulnerabilidad** | 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 auditorías de SW y respaldo de datos regularmente. [BAJA] |
| **Activo** | Servidores ubicados en la sala. [CRÍTICO] | Sitio web. [CRÍTICO] | Archivos en un repositorio compartido. [MEDIO] |
| **Impacto** | 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] |
| **Probabilidad de Ocurrencia** | La última falla del sistema ocurrió el mes pasado. [ALTA] | Se detectó una sola DDoS en los últimos 2 años. [MEDIA] | [MEDIA] |
| **Riesgo** | Pérdida de $30M. (ALTO) | Pérdida de $3M por cada hora caída. (MEDIO) | (BAJO) |
| **Recomendación de Control** | Comprar un nuevo sistema de aire acondicionado por un costo de $1.000.000. | Monitorear el *firewall*. | Continuar el monitoreo de cambios de permisos de usuarios y respaldos. |

[1] 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.

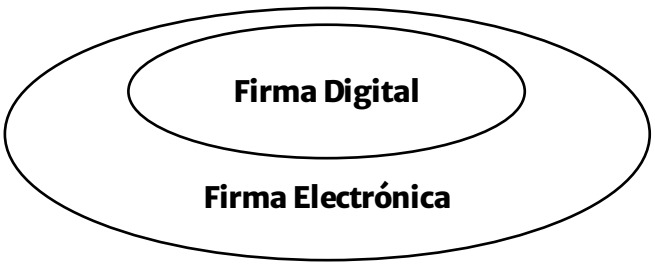
Herramientas y debilidades:

- **Inteligencia Artificial** → manipulación tendenciosa a través de *fake news* y *deepfakes*.
- **Tecnología Móvil de 5ta 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.**<br>• Se basa en esquemas de clave pública y clave privada. |
| **FIRMA ELECTRÓNICA** |
| **Concepto legal que apunta a asegurar la identidad de alguien.**<br>• 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).