**Benchmark**

Proceso sistemático y continuo que permite evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones. Estas mediciones son las que van a definirnos cuál es el rendimiento de lo que estamos evaluando y así permitir la toma de decisión correspondiente. Permite:

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

Etapas

* *Determinar el elemento de estudio*: Elegir los factores y variables claves. Seleccionar las opciones disponibles en el mercado.
* *Preparar el entorno de prueba*: Recopilar requerimientos del tipo de Benchmark a realizar. Realización de tareas previas a la etapa de ejecución. Preparación del ambiente en donde se realizara la prueba.
* *Realizar benchmark*: Someter elementos a las pruebas. Medir las respuestas para cada variable analizada y determinar la diferencia respecto a lo que se compara.
* *Analizar resultados*: Descartar elementos estudiados que en base a los resultados, no cumplen con las necesidades. Realizar informes para mostrar los resultados obtenidos. Determinar si se requiere recalibrar el benchmark y reiniciar. Desarrollar planes de acción.

Tipos

* *Sintéticos*: Mide el rendimiento de un componente individual del hardware, normalmente llevando el componente escogido a su máxima capacidad.
  + *Whetstone:* Provee una medida muy razonable de rendimiento de monoprocesadores de aritmética flotante. (Unidad = MWIPS)
  + *Dhrystone:* Para medir la velocidad del sistema en cuanto a rendimiento no numérico. Mide la capacidad del procesador. (Unidad = DPS)
* *Aplicaciones*: Simulan una carga de trabajo para medir el comportamiento global del equipo. Tienen gran cantidad de código y de requerimientos de almacenamiento de datos.
* *Test de Bajo nivel*: Miden directamente el rendimiento de los componentes. Ej: reloj d CPU
* *Test de Alto nivel*: Están más enfocados a medir el rendimiento de la combinación componente/controlador/SO de un aspecto específico del sistema. Ej: E/S con archivos

Más conocidos (faltan varios)

* *HINT*: Resolver un problema matemático cuya solución se puede ir mejorando. (QUIPS)
* *TPC*: (TPC-C) Mide los resultados en transacciones por minuto. Se simula la carga de trabajo y solicitan la ejecución de una mezcla de cinco tipos de transacciones: New-Order, Payment, Order-Status, Delivery, Stock-Level. Las transacciones simuladas deben cumplir:
  + *Atomicidad*: Las transacciones se confirmen o se aborten. Si falla, no a medias.
  + *Consistencia*: Solo se ejecutan aquellas operaciones que no van a romper la reglas.
  + *Aislamiento*: una operación no puede afectar a otras.
  + *Durabilidad*: Una vez realizada la operación, no se deshace aunque falle el sistema.

**Almacenamiento**

* *Centralizado*: Un único sistema central.
* *Sistemas de almacenamiento accedidos de manera directa (DAS)*: Accesible mediante controladoras e interfaces conectadas en forma directa a los servidores implementando protocolos de comunicación específicos.
* *Red de almacenamiento de datos (SAN)*: ‘’ ‘’ ‘’ ‘’ ‘’ ‘’
* *Red IP de almacenamiento (NAS)*: Almacenamiento accesible por medio de redes IP de transferencia de datos.
* *Servicios de almacenamiento en la nube*

Disponibilidad (Availability): Es la posibilidad de poder continuar dando servicios de storages en el evento de una falla de hardware o software.

Rendimiento (Performance): Es la métrica usada para definir la velocidad de un sistema de almacenamiento. Existen 3 métodos de medición: I/OxSeg, MBxSeg, response time (ms).

Soluciones de resguardo offline

* *Almacenamiento de datos en cintas magnéticas*: Tipes Drives manuales, semi-automaticos, automáticos, virtuales.
* *RAID*
  + *RAID 0*: Los datos se dividen en pequeños segmentos y se distribuyen entre varios discos. Lectura y escritura rápida. No tolerancia a fallos.
  + *RAID 1*: Discos adicionales sobre los que se realiza una copia bloque a bloque. Lectura rápida. Tolerancia a fallos. Costoso.
  + *RAID 3*: Dedica un disco al almacenamiento de información de paridad. Altas tasas de transferencia y confiabilidad. Rendimiento de transacción pobre.
  + *RAID 5*: Utiliza bloques distribuidos para el almacenamiento de información de paridad que le permite ofrecer tolerancia a fallas. Escrituras simultáneas.
  + *RAID 6*: Implementa doble paridad usando dos discos. Permite la falla de dos discos de manera simultánea. Costoso.

Plan de contingencia y recuperación

*Recovery Time Objective (RTO)*: Tiempo que pasará una infraestructura antes de estar disponible.

*Recovery Point Objective (RPO)*: Cuantos datos está dispuesta a perder la organización antes de estar disponible.

(unos cuadros que no se entienden una chota)

Estrategias de protección de datos

* *Copias de resguardo en discos locales y externos*: Rápido, integración con apps y BDD. Costoso. No transportable a otro datacenter. Si la falla se produce en el dc no recuperable.
* *Copias de resguardo periódicas en cinta*: Bajo costo. Transportable a otro dc. Demora más.
* *Replicación de datos en sitio externo*: Reguardo fuera de dc. No continuar operación.
* *Replicación de datos en dc externo implementado como sitio de contingencia*: Continuar operación. Rapidez. Costo alto.

Medidas preventivas: evitar ocurrencia. de detección: para detección. correctivas: recuperar la op.

**Tipos de computadores**

* *Mainframe*: Computadora central destinada a procesos críticos a gran escala con gran confiabilidad y capacidad de procesamiento. Es una potente herramienta para dar soporte en línea a miles de usuarios. Tiene la capacidad de ser reparado sin detener los servicios. Muy costosa. Incompatible con otras plataformas. Decenas de procesadores.
* *Supercomputadora*: Es una computadora con capacidad de cálculo muy superiores a una PC común. Se usan para aplicaciones especializadas que requieren enormes cantidades de cálculos matemáticos. Costosa, consume mucha energía. Miles de procesadores.

Paralelismo *Funcional*: Procesos, hilos, redundancia. *De datos*: Vectores, matrices.

Métricas de rendimiento

* *MIPS*: Millones de instrucciones por segundo.
* *MFLOPS*: Millones de operaciones en punto flotante por segundo.

Servidores

* *Tipo torre*: Su infraestructura es una PC básica. Empresas pequeñas y medianas
* *En rack*: Requieren infraestructura específica para su instalación y funcionamiento. Para dc
* Blade: Ideales para reducir espacio y disminuir el consumo de energía.

Cluster

Grupo de computadoras individuales (denominadas “nodo”) trabajando en forma conjunto bajo una solución de software y conectividad que se ponen al servicio del procesamiento de una determinada tarea.

*Componentes*: Nodos, almacenamiento, SO, conexiones red, protocolos de comunicación, soft.

*Características*: Rendimiento (procesa grandes volúmenes), Disponibilidad (si falla un nodo no impide operatividad), Eficiencia (aprovecha recursos en su totalidad), Escalabilidad (nuevos nodos)

*Tipos de Cluster*

* *LB-C (de balanceo de carga)*: Rutea la totalidad de la carga de trabajo por medio de un servidor, el cual la distribuye entre los nodos de procesamiento activo.
* *HP-C (de alta performance)*: Explota el potencial del procesamiento en paralelo entre múltiples computadoras. Es el más indicado para funciones complejas.
* *HA-C (de alta disponibilidad)*: Diseñado para mantener computadoras redundantes que pueden convertirse en sistemas de backup en caso de una falla del sistema.

Grid computing: Permite la organización de un “pool” de múltiples recursos. Los recursos pueden ser accedidos por una gran cantidad de usuarios. No es un cluster, pero puede serlo o contenerlo.

Cloud Computing: Permite disponibilidad de implementación y crecimiento inmediata. Dependencia de los proveedores. Exposición de datos sensibles ante terceros.

* *IaaS (Infraestructure as a Service)*: Procesamiento, Cache, Almacenamiento, Streaming.
* *PaaS (Platform as a Service)*: Desarrollo de aplicaciones.
* *SaaS (Software as a Service)*: Hosting de aplicaciones en la nube.
* *MSP (Managed Service Provider)*: Backup/Restore, Monitoreo.

**Software base**

Sistema Operativo: Es un conjunto de software de sistema que controla como las aplicaciones acceden al hardware y la manera en que los usuarios acceden a recursos.

*Compuesto por*: Gestor de procesos, gestor de MP, gestor de Memoria Secundaria, gestor de E/S. Sistema de archivos, sistema de protección, sistema de comunicaciones. Programas de sistema.

*Clasificación*

* *Administración de tareas*:
  + *Monotarea*: Solamente puede ejecutar un proceso en un momento dado.
  + *Multitarea*: Es capaz de ejecutar varios procesos de forma alternada.
* *Administración de usuarios*:
  + *Monousuario*: Si sólo permite ejecutar los programas de un usuario a la vez.
  + *Multiusuario*: Permite que varios usuarios ejecuten simultáneamente, accediendo a la vez a los recursos. Utilizan métodos de protección de datos.
* *Manejo de Recursos*:
  + *Centralizado*: Permite utilizar los recursos de una sola computadora.
  + *Distribuido*: Permite utilizar los recursos de más de una computadora a la vez.
* *Manejo de eventos*:
  + *Tiempo Real*: multitarea, con intercambio de procesos basado en prioridades y eventos externos.
  + *Tiempo compartido*: intercambian tareas en función de interrupciones de reloj.

*Criterios de selección*

* *Aplicaciones*: Software que se requiere utilizar. Disponibilidad de otras aplicaciones. Disponibilidad de herramientas de gestión y su facilidad de extensión.
* *Estabilidad*: Evolución del núcleo y las interfaces. BUG, Instalación y actualización.
* *Confiabilidad*: Consumo del sistema base. Optimización del uso de recursos (Deadlocks).
* *Seguridad*: Reporte y resolución de BUG. Administración de credenciales y permisos.
* *Escalabilidad*: Capacidad de clusterización. Modularidad y personalización.
* *Usabilidad*: Facilidad de uso, de aprendizaje y de automatización de tareas. Interfaz GUI.
* *TCO*
  + *Disponibilidad de RRHH capacitados*:
    - SO privativos: Disponibilidad Guru 🡪 costoso, Disp. novatos 🡪 económico
    - SO no privativos: Aprendizaje más lento. Riqueza técnica superior.
  + *Licienciamiento y soporte*:
    - SO privativos: Licenciamiento pago (WINDOWS, UNIX)
    - SO no privativos: Licenciamiento gratuito (BSD, LINUX)
  + *Soporte*: Comercial (pago) o de la comunidad (voluntario).

Tipos de licenciamiento

* *Software propietario*: Utiliza una licencia de código cerrado. Código fuente inaccesible.
* *Software libre*: (COPYLEFT) El trabajo realizado se mantiene disponible de forma libre.

L0) Ejecutar el programa, para cualquier propósito. L1) cambiarlo para que haga lo que se requiera (OPEN SOURCE). 2L) Redistribuir copias. L3) Mejorar y publicar sus mejoras.

Open source: Software que se pone a disposición con el código fuente, y algunos otros derechos.

Freeware: Software propietario que puede ser utilizado sin costo de licencia.

**Bases de datos**

Que se busca con los SGBD: Abstracción, Independencia, Redundancia minima, Consistencia, Seguridad, Integridad, Respaldo y Recuperación, Control de concurrencia, Tiempo de Respuesta.

Aspectos funcionales

* *Distribución y replicas*: La replicación es un conjunto de herramientas para copiar y distribuir datos y objetos de una BDD a otra y sincronizarlas para mantener la coherencia.
* *Concurrencia y niveles de aislamiento*: (MVCC) Provee de cada usuario conectado con la base de datos una “foto” de la BDD para esa persona. Ningún cambio realizado será visualizado por otros usuarios hasta que la transacción haya sido confirmada.
* *Tipos de Índices*:
  + *Índices Hash*
  + *Árbol B*: Utilizado en la mayoría de los índices de BDD.
  + *Índices bitmap*: Consumen menor almacenamiento que Árbol B. En un índice bitmap se crea un mapa de bits para cada valor clave del índice.
  + *Árbol R-/R+*: Similar al Árbol B pero optimizado para acceder a información multidimensional. Generalmente utilizado para coordenadas geográficas.
  + *Índice parcial*: Se aplica solo a aquellas filas que cumplen una determinada condición, permitiendo generar un índice para un propósito más específico que un índice común, evitando indexar filas innecesarias.
  + *Índices de expresión*: Se basan en el indexado de una función o expresión aplicada sobre las columnas del registro, y es eso lo que se indexa. Ej.: por upper(col1).
  + *Fulltext*: Permite la búsqueda de n palabras combinadas.
* *Backups*: (no te explica una mierda, solo dice que tipos de backups tiene c/ BDD)
* *Vistas materializadas*: Una vista es una consulta guardada en la BDD, que permite reutilizarse como si reutilizamos una consulta ya hecha. En este caso lo que se guarda no es el resultado de la consulta, sino las relaciones utilizadas para volver a ejecutarla.
* *Optimización de consultas*: Proceso de selección del plan de evaluación de las consultas más eficiente de entre las muchas estrategias generalmente disponibles para el procesamiento de una consulta dada.
* *Particionamiento*: Técnica para dividir tablas grandes en tablas pequeñas más manejables, y permite que se manipulen desde el punto de vista de la aplicación de una manera uniforme. Mejoran la performance. Los hay por rango y por hash.

Comparación de rendimientos: **TPC-H** es un punto de referencia apoyo a las decisiones que consiste en una cadena de consultas ad-hoc…

Flashback: Característica para recuperar de errores humanos. Permite a los usuarios ver el estado del dato, trasladado a un instante anterior sin requerir de cambios estructurales a la BDD.

Bases de datos no SQL

* *Ventajas*: Resuelven problemas de escalabilidad horizontal, Manejo de enorme cantidades de datos, Fácil de escalar, Excelentes tiempos de respuesta en almacenamiento y recuperación.
* *Desventajas*: No cumplen ACID, Utilizan mayormente memoria en vez de disco como la principal ubicación de escritura, Falta de madurez, Falta de estándares, Aplicación para BI.

Clasificación

* *Clave-Valor*: Los datos se almacenan en pares del tipo clave-valor.
* *Column Family*: Permiten almacenar claves mapeadas a valores y esos valores agrupados en múltiples familias de columnas siendo cada columna un mapa de datos.
* *Basadas en documentos*: La base de datos almacena y recupera documentos que pueden estar en XML, JSON o BSON.
* *Basadas en grafos*: Permiten almacenar entidades y relaciones entre esas entidades. Tanto los nodos como las relaciones tienen sus propiedades asociadas.

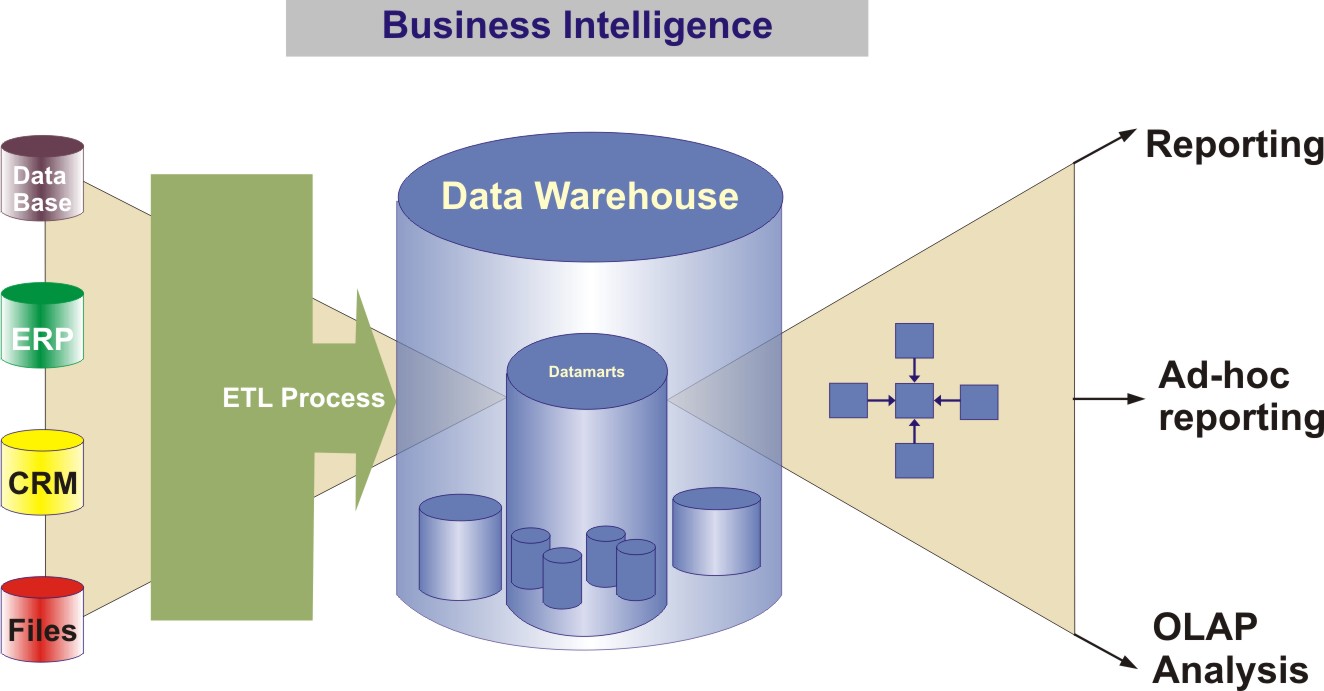
Business Intelligence (BI)

Abarca las tecnologías de datawarehousing los procesos en el ‘back end’, consultas, informes, análisis y herramientas para mostrar información y procesos en el ‘front end’.

*Objetivo*: asistir en la toma de decisiones y, posteriormente descubrir conocimiento desconocido para la organización hasta el momento.

*Beneficios*:

* *Tangibles*: Reducción de costes y tiempo, generación de ingresos.
* *Intangibles*: Tener disponible la información para la toma de decisiones.
* *Estratégicos*: Todos aquellos que nos facilitan la formulación de la estrategia.



**Virtualización de sistemas**

Es un modo de abstraer aplicaciones y sus componentes subyacentes del hardware que lo soporta y presentar una vista lógica o virtual de esos recursos.

*Objetivos*: Mayor nivel de performance, escalabilidad, confiabilidad/disponibilidad, agilidad. Creación de un dominio unificado de seguridad y gestión.

Vistas: La virtualización puede crear una vista artificial donde varias computadoras individuales se muestran como una sola o donde una computadora se muestra como varias individuales.

Capas

Virtualización de acceso: Hardware y software que permite que prácticamente cualquier dispositivo acceda a cualquier aplicación y viceversa, sin que ninguno tenga mucho conocimiento del otro. Se mete en la comunicación entre dispositivos.

Virtualización de aplicaciones: Software que permite que las aplicaciones corran en diferentes sistemas operativos y plataformas de hardware. Existen 2 formas:

* *Del lado del cliente*: Se crea un ambiente protegido que posibilita que las aplicaciones estén aisladas de otras y del SO base.
* *Del lado del servidor*: La colocación de aplicaciones del lado del servidor en un ambiente virtual. Aplicaciones que son incompatibles pueden funcionar juntas.

Virtualización de Procesamiento: Hardware y software que oculta la configuración de hardware a los SO o aplicaciones. Tipos:

* *Un sistema único se muestra como varios*: El software de máquina virtual permite que todo el software que constituye un sistema se encapsule en un archivo de máquina virtual. La virtualización del SO permite que muchas aplicaciones corran bajo un solo SO y darle a cada una un ambiente aislado y protegido.
* *Varios sistemas se muestran como uno solo*: Los monitores de procesamiento paralelo permiten que varias máquinas ejecuten las mismas aplicaciones o componentes con el objetivo de reducir el tiempo de procesamiento. Los monitores de carga de trabajo permiten que varias instancias de una misma aplicación puedan correr simultáneamente en varias máquinas.
* *Varios sistemas se muestran como uno solo*: Los monitores de alta disponibilidad/tolerancia a fallas/recuperación de desastres permiten que las personas utilicen servicios de protección contra fallas. La virtualización de memoria o memoria cache distribuida permite que varios sistemas compartan sus memorias internas.

Virtualización de Red: Hardware y software que presenta una vista la red que difiere de la vista física. O hacer que varias redes se presenten como una sola. Funciones:

* *Ruteo*: El tráfico dirigido a clientes o servidores remotos se envía de una red a otra hasta alcanzar el destino.
* *Network Address Translation (NAT)*: Una dirección de red puede ser realmente usada por un número de clientes o servidores. Osea, IP única para un sector.
* *Network Isolation*: los clientes y servidores solamente pueden ver (y por ello comunicarse con) sistema específicos.

Virtualización de almacenamiento: Hardware y software que oculta dónde están los sistemas de almacenamiento y qué tipo de dispositivos realmente almacenan aplicaciones y datos. Funciones:

* *Viabilidad de sistemas distribuidos de archivos*: Los dispositivos de almacenamiento remoto se arman para que parezcan directamente conectados al sistema.
* *Creación de volúmenes artificiales de almacenamiento*: Varios dispositivos de almacenamiento se unen para crear la imagen de un dispositivo único mucho mayor.
* *Creación de arreglos de volúmenes de almacenamiento*: Los datos y las aplicaciones se pueden distribuir sobre un número de dispositivos y servidores de almacenamiento para mejorar la performance global.
* *Viabilidad de mayor control sobre el espacio de almacenamiento*: los dispositivos se pueden segmentar en varios “file systems”, lo que permite una utilización más eficiente del almacenamiento.
* *Viabilidad de compartir dispositivos de almacenamiento entre sistemas incompatibles*: SOs usan mecanismos diferentes para almacenar y recuperar aplicaciones y datos. La virtualización permite que todos ellos compartan los dispositivos de almacenamiento y los archivos que contienen.

Seguridad de ambientes virtuales: Se trata de tecnología de software que controla el acceso a diferentes elementos en un ambiente virtual y previene usos no autorizados o maliciosos

Gestión de ambientes virtuales: Software que permite que varios sistemas sean provistos y gestionados como si fueran un recurso individual. Si los ambientes de vuelven más complejos, esta capa se vuelve más importante. Funciones:

* Creación y provisión ambientes virtuales o componentes de los mismos
* Monitoreo y control de su ejecución
* Análisis del registro de ejecución para encontrar problemas
* Optimización del uso de ambientes o componentes

Big data: Combinación de virtualizaciones de procesamiento y almacenamiento. Permiten gestionar conjunto muy grandes de datos y dispositivos de almacenamiento.

Virtualización de servidores: Se utiliza aplica al uso de tecnología de máquina virtual o virtualización de sistema operativo para permitir que un solo sistema físico soporte múltiples cargas.

(FALTAN)

Clusters

Virtualización de escritorio

Sistemas de alta performance

Procesamiento extremo de transacciones

**Arquitectura de Software**

Es el conjunto de estructuras que componen al software. En sí, son elementos de tecnología, relaciones y propiedades entre ellas. Las estructuras contienen elementos y sus relaciones. La arquitectura trata de omitir ciertos detalles internos de cada uno de estos elementos, para abstraerse de su dificultad. Es importante para:

* El usuario: que la rapidez, de la disponibilidad y de que sea confiable.
* El cliente: que sea implementada conforme al calendario y al presupuesto.
* El Project Manager: que permita que los equipos trabajen independientemente.
* El arquitecto: que todos los puntos anteriores funcionen correcta y sincronizadamente.

Requerimientos

* *Requerimientos Funcionales*: Estos definen, qué tiene que hacer el sistema, y cómo debe actuar en tiempo de ejecución.
* *Requerimientos de calidad del sistema*: Estos definen, las características de los requerimientos funcionales, a lo largo de todo el producto. Ej: rapidez para hacer un ReqF.

Restricciones: La restricción es una decisión de diseño que debe tomarse. Ej.: elección de lenguaje.

Atributos de calidad

* *Disponibilidad*: Minimizar las interrupciones de servicio y mitigar las posibles fallas. Hay 3 tipos de tácticas: detectar, recuperar y prevenir fallas.
* *Interoperabilidad*: 2 o más sistemas pueden intercambiar y comprender información importante vía interfaces.
* *Adaptabilidad*: Posibilidad de cambio al menor costo y riesgo.
* *Performance*: Habilidad que tienen los sistemas para contestar a los eventos que ocurren en tiempo y forma. Hay 2 tipos de tácticas: disminuir demanda o aumentar gestión de rec.
* *Seguridad*: Habilidad que tienen los sistemas, para proteger datos e información de un acceso no autorizado. Hay 2 tipos de tácticas: Detectar o resistir ataques (revocar o lock).
* *Capacidad de prueba y testeo*: Facilidad con la que el software puede hacer para demostrar sus fallos a través de pruebas.
* *Usabilidad*: Cuán fácil es para el usuario, ejecutar una tarea deseada y cuál es el tipo de usuario que soporta. Comprende el aprendizaje, eficiencia, confianza, etc.

Tácticas y patrones de arquitecturas

Son los caminos para realizar un buen diseño de estructuras que pueden ser reutilizadas.

Los patrones de arquitectura establecen relaciones entre: Contexto, problema y solución.

Las definiciones de patrones son estrictas pero toleran pequeñas modificaciones.

Tipos de patrones

Requerimientos de arquitectura significativos (ASR): son aquellos que tienen un efecto profundo sobre la arquitectura. La arquitectura bien podría ser diferente en ausencia de tal requisito

**Métricas**

Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo. Se usan para: Evaluar el proyecto, Seguir riesgos potenciales, Predecir problemas críticos, Ajustar el flujo y las tareas, Evaluar la habilidad del equipo en controlar la calidad.

Indicador: Métrica o combinación de métricas que da una visión profunda del proceso y proyecto de software o el producto. Se recopilan medidas y desarrollan métricas para obtener indicadores.

Métricas de software

* *Medidas directas*: Líneas de código producidas (LDC), Velocidad de ejecución, Memoria utilizada, Defectos en un periodo de tiempo.
* *Medidas indirectas*: Funcionalidad, Calidad, Complejidad, Facilidad de mantenimiento

Métricas orientadas al tamaño: (Errores, defectos, $, paginas documentadas) por KLDC (miles LDC)

Métricas orientadas a la función (Errores, defectos, $, páginas documentadas) por PF

Puntos de Función

Se basa en el diseño lógico del sistema y en características externas.

*Objetivos*: Medir independientemente de la tecnología, Proporcionar factor de normalización para comparar software.

*Elementos*:

* *Entradas (External Input)*: Entran desde fuera de la aplicación.
* *Salidas (External Output)*: Datos que salen de la aplicación.
* *Consultas (External Query)*: Combinación de E/S que se obtiene de una búsqueda.
* *Archivos internos (Internal Logic File)*: Grupo de datos relacionados, mantenidos y utilizados dentro de la aplicación
* *Archivo de interface externa (External Interface File)*: Grupo de datos relacionados, mantenido por otra aplicación.

Para cada parámetro externo se indica su complejidad (baja, media o alta) en base al número de campos, cantidad de ficheros que referencia y el número de registros (en ficheros).

Medida de calidad del software

* *Corrección*: Grado en que lleva a cabo su función (defectos por KLDC).
* *Facilidad* *de mantenimiento*: Facilidad para corregir un error o realizar un cambio.
* *Integridad*: Habilidad para resistir ataques (accidentales e intencionados).
* *Facilidad* *de uso*: Que tan amigable es con el usuario.

Herramientas de estimación**:** Juicio experto, datos históricos de proyectos similares. Estimaciones individuales. Paramétrica (Definir una formula en función del esfuerzo por unidad de trabajo).

PERT (Por 3 valores): Calcula la duración de cada actividad con promedios ponderados de estimaciones. La duración del proyecto es la suma de la duración de las tareas del camino crítico.