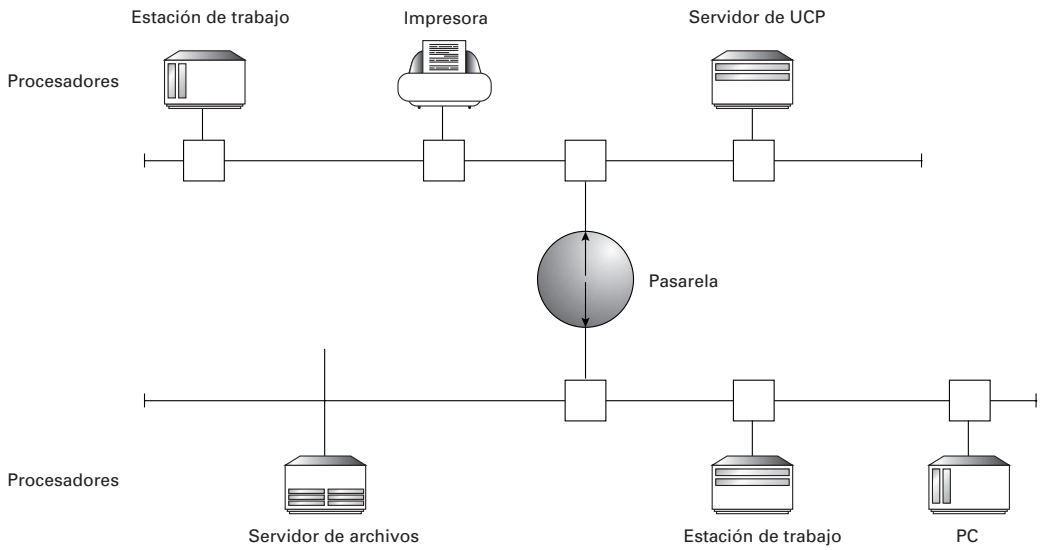
# FUNDAMENTACIÓN DE ARQUITECTURAS DE BASES DE DATOS

Existe una diversidad amplia en cuanto a las estructuras de bases de datos, nosotros trabajaremos las que exige la tendencia del mercado colombiano :

- Arquitecturas *Centralizadas*
- Arquitecturas *Cliente - Servidor*
- Arquitecturas de *Sistemas Servidores*
- Arquitecturas *Paralelas*
- Arquitecturas *Distribuidas*




Una base de datos es un contenedor organizado de información, en donde se puede realizar la actualizacion (registrar, modificar , eliminar y consulta) de dicha información. Al existir diferentes necesidades de acceso de la informacion por parte de sus usuarios, asi existen multiples formas (arquitecturas) de organizar la informacion. Las bases de datos pueden ser centralizadas, cliente / servidor, sistemas servidores, paralelas y distribuidas.

Todas las empresas de los diferentes sectores productivos, sin importar a que se dediquen, poseen una concordancia entre si, siendo esto el desarrollo de su negocio. Este concepto se conoce como **core** de negocio. Este core de negocio es a lo que se dedica la empresa, compañía o entidad, sin importar si es privada, estatal o incluso mixta. Este core de negocio puede ser único o compuesto, lo que significa que una compañía puede tener uno o varios core de negocio. Por ejemplo una compañía puede dedicarse a vender productos de paquetes de turismo y además a vender tiquetes aéreos. Esta compañía se encuentra en el sector turístico pero su core es de Servicios. Otra compañía puede fabricar refrescos y luego venderlos a almacenes de grandes superficies, supermercados, tiendas, kioscos etc. En este caso la compañía posee un core de negocio de industria y además de servicios. Es muy importante que usted ubique a la compañía en el sector y además defina el core(s) de negocio de esta.

Para el desarrollo de un core de negocio se incorporan uno o varios **procesos**, siendo un proceso la línea de trabajo que se debe desarrollar para ejecutar todo lo concerniente a la unidad productiva del negocio, el proceso esta constituido por **actividades**, una actividad es el conjunto de **tareas**(pasos) que se necesitan para ejecutar unas acciones que darán como resultado la elaboración del proceso, veamos un ejemplo.

***Compañía:*** **Almacen de Grandes Superficies.** ***Core de negocio:*** **Servicios.**

| Procesos | ① VENTAS | | ② COMPRAS | |
|---|---|---|---|---|
| Actividades | ❶ Facturación | ❷ Devolución | ❶ | ❷ Orden Compra |
| Devolución Tareas | Actualizar Inv.<br>Verificar Cupo.<br>Actualizar Saldo<br>Verificar Crédito | Verificar Fact.<br>Actualizar Inv.<br>Generar Cheque. | Verificar Faltantes | Actualizar Inv. |

Al momento de realizar las actividades de un proceso que tiene como objeto el desarrollo de una unidad de negocio, surgen las **necesidades** de los clientes, se debe entender como cliente el actor participativo de la unidad o core de negocio. Las necesidades son la problemática que existe en el desarrollo de las actividades de los procesos, los cuales exigen unas soluciones que permitan realizar un flujo normal de las tareas de estas actividades, siempre manteniendo un control optimo del tiempo, estas soluciones se conocen como **requerimientos**. Cabe anotar y corregir una definición errada y es que los requerimientos no son del cliente o establecidos por este, ya que el no tiene la capacidad de darlos, los requerimientos son desarrollados por personal calificado como los analistas o desarrolladores de software.

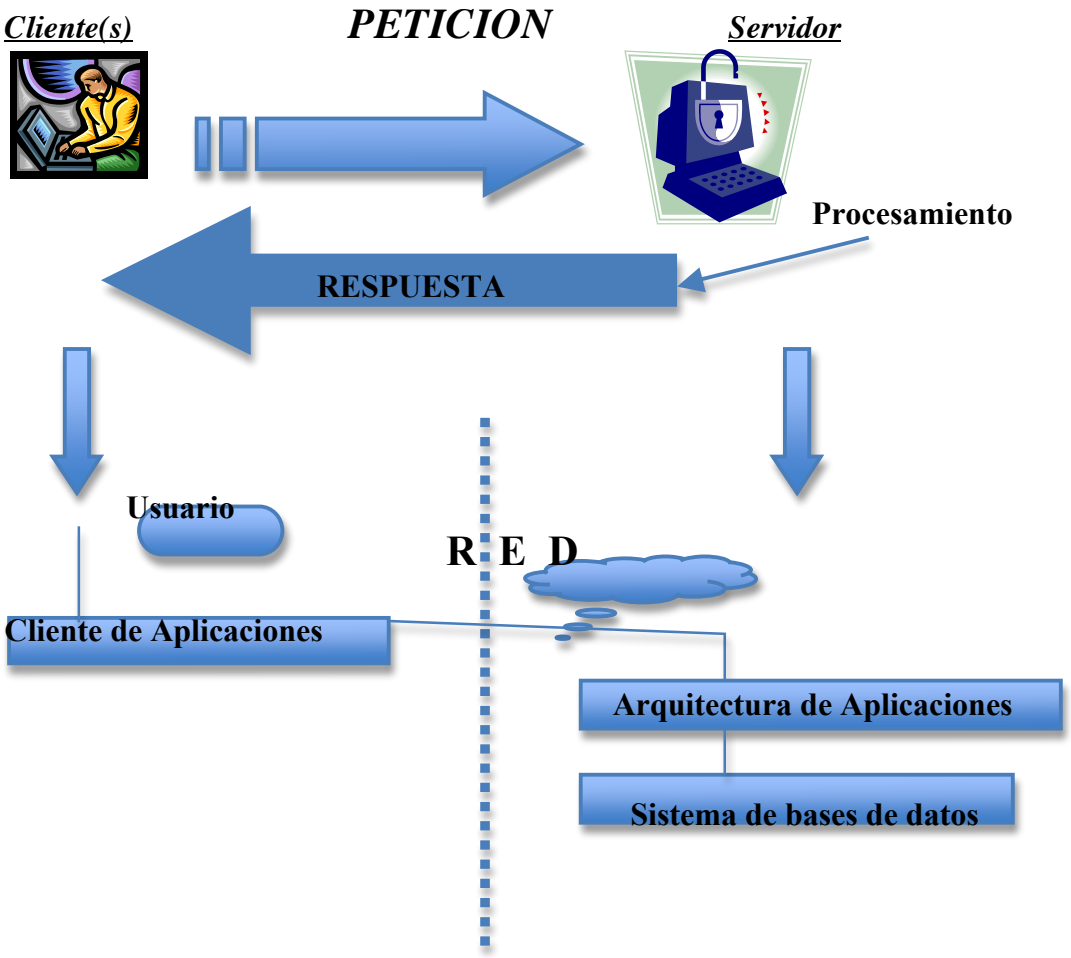
El activo fijo mas importante para una compañía es la información, ya que con el control adecuado de esta, podemos ser muy competitivos, también si perdemos control de la

información podemos estar en serios problemas. Podemos definir como **información** el conjunto de datos coherentes con los procesos del core de negocio al cual le estamos fabricando una solución. Un **dato** se puede definir como el contenido informático de la definición de un algo. Veamos un ejemplo, si estamos en una entidad educativa y uno de los requerimientos es asignar horarios a las salas o aulas de sistemas, la información es todo lo pertinente a esta tarea, ejemplo la información de un profesor, uno de sus datos seria su No de identificación (16.456.123), este seria el contenido informático de la definición (No de id.) de un algo (profesor).

Todas las compañías tienen una **interfaz** de trabajo que esta compuesta por un conjunto de criterios que me constituyen mi sistema de información. Los criterios que componen la interfaz son variados, miremos algunos.

- ✓ El lenguaje o los lenguajes de desarrollo
- ✓ El motor de base de datos
- ✓ Topologías de redes
- ✓ La metodología de desarrollo
- ✓ Sistema Operativo
- ✓ Arquitecturas
- ✓ Telecomunicaciones
- ✓ Etc

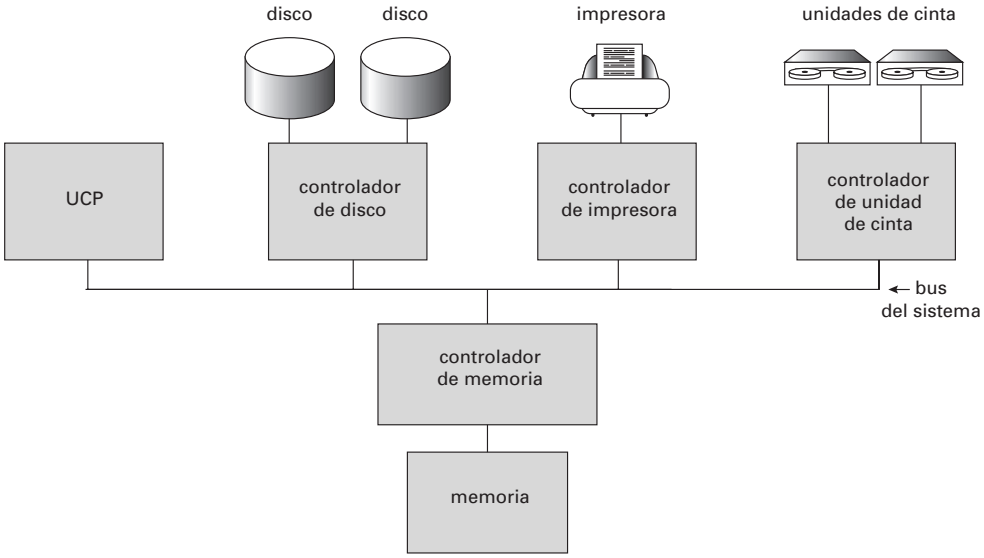
Muchas veces existen compañías que coinciden en sector productivo y core de negocio, esto no significa que coincidan en interfaz.
Cuando se trata de definir los criterios de trabajo con arquitecturas en sistemas de información, siempre escuchara usted dos básicos, Cliente y Servidor. Empecemos hablando del cliente, es importante aclarar que este termino cliente no es igual al explicado previamente. Un **cliente** en una arquitectura es aquel que realiza la petición, nunca es un actor siempre será un equipo o dispositivo(computador, datafono, celular, tableta, etc.). Un **servidor** es aquel proveedor de servicios de procesamiento, es decir recepciona las peticiones de uno o varios clientes, las procesa y da respuesta a estos. Por favor desvincule el criterio servidor del de un computador, no se imagine que un computador es un servidor, un servidor es la composición de muchos periféricos, entre ellos muchos procesadores, memorias, discos, telecomunicaciones y muchos dispositivos que usted aprenderá a conocer e identificar durante el proceso de formación.




La forma y manera de organizar la información depende directamente de la necesidad y del bosquejo del core de negocio, vamos a tomar prestado algunos procesos de unas unidades de negocio de nuestra sociedad para entender un poco como funcionan estas ***arquitecturas***. Pensemos en una unidad de negocio de grandes superficies(supermercados de grandes cadenas de almacenes), y utilicemos una de las actividades de uno de sus procesos, la facturación que se desarrolla en un punto de venta. Bueno ya nos estamos escenificando(es necesario conocer que hoy en día se desarrolla software con diseño basado en escenarios, estudios de casos y justificación económica). Antes de continuar con el ejemplo recuerde que una petición la ejecuta un cliente, y este no es mas que un periférico, cuando ingresa el PC(personal computer), ingresa un equipo con **CPU**(unidad central de proceso - también conocido como **UCP**), que va a tener la capacidad de desarrollar actividades de apoyo en proceso al servidor o servidores. Pero también existen periféricos "no inteligentes", que realizan peticiones, estos se conocen como terminal bruta , y estas son las que permitirán explicar la primera arquitectura, siendo esta la **Centralizada**. Una arquitectura se considera centralizada cuando existe un proveedor de servicios que se encarga de suministrar procesamiento de datos a las peticiones de un numero de terminales, las cuales tendrán como una única tarea servir de apoyo de conexión a un usuario frente al sistema de información, estos sistemas estarán administrados por un sistema operativo multiusuario. Pueden configurarse periféricos autónomos con capacidad de almacenamiento limitado con tecnologías básicas de administración de datos los cuales trabajan de manera individual, y posteriormente bajo un criterio denominado batch(al final del día) entregan la

información a un servidor, esto se realiza fuera de la línea del sistema de información, estos periféricos funcionan y son administrados por sistemas operativos monousuario. Para la primera alternativa de arquitectura centralizada se estarían utilizando terminales que usted puede identificarlas en los puntos de ventas de los grandes almacenes (cajas de punto de venta), mientras que la segunda alternativa usted podrá identificarlo en aquellos periféricos que los vendedores de tienda a tienda llevan para la captura de pedidos de gaseosas, cerveza, etc.

**Figura No 1 : Servidor para un sistema informático centralizado multiusuario.**




Como las computadoras personales son ahora más rápidas, más potentes y más baratas, los sistemas se han ido distanciando de la arquitectura centralizada. Los terminales conectados a un sistema central han sido suplantados por computadoras personales. De igual forma, la interfaz de usuario, que solía estar gestionada directamente por el sistema central, está pasando a ser gestionada, cada vez más, por las computadoras personales. Como consecuencia, los sistemas centralizados actúan hoy como **sistemas servidores** que satisfacen las peticiones generadas por los ***sistemas clientes***. Usted podrá identificarlos en sistemas de información empresarial donde se administran por intranet, por ejemplo sistemas de facturación, compras, contabilidad, etc. Donde las estaciones de trabajo son **PC.** O cuando usted registra sus datos personales en paginas de Internet para realizar cualquier actividad.

**Figura No 2 : Estructura general de un sistema cliente-servidor.**

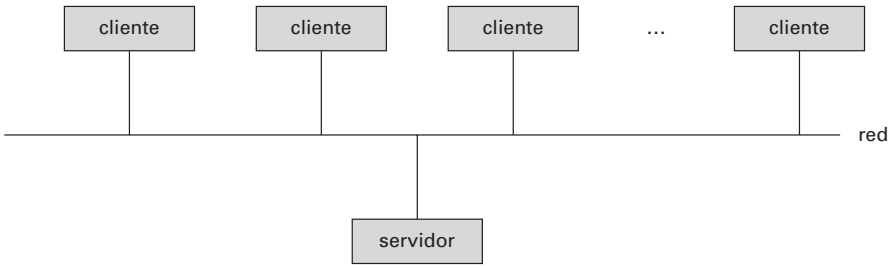
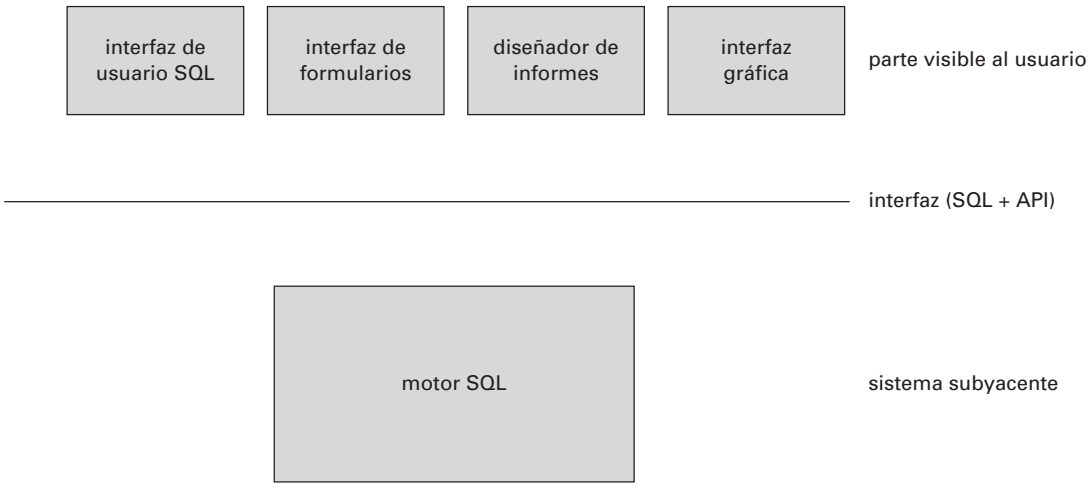



**Figura No 3 : Funcionalidades de la parte visible al usuario y del sistema subyacente.**




Como se muestra en la Figura 3, la funcionalidad de una base de datos se puede dividir a grandes rasgos en dos partes: la parte visible al usuario y el sistema subyacente. El sistema subyacente gestiona el acceso a las estructuras, la evaluación y optimización de consultas, el control de concurrencia(cantidad de usuarios conectados al tiempo a la base de datos) y la recuperación. La parte visible al usuario de un sistema de base de datos está formado por herramientas como formularios, diseñadores de informes y facilidades gráficas de interfaz de usuario. La interfaz entre la parte visible al usuario y el sistema subyacente puede ser SQL(lenguaje estructurado de consultas) o una aplicación.
Las normas como *ODBC* (Open Database Connectivity) y *JDBC(tecnología de acceso a los datos basado en java)*, se desarrollaron para hacer de interfaz entre clientes y servidores. Cualquier cliente que utilice interfaces ODBC o JDBC puede conectarse a cualquier servidor que proporcione esta interfaz.
En las primeras generaciones de sistemas de bases de datos, la carencia de tales normas hacía que fuera necesario que la interfaz visible y el sistema subyacente fueran proporcionados por el mismo distribuidor de software. Con el aumento de las interfaces estándares, a menudo diferentes distribuidores proporcionan la interfaz visible al usuario y el servidor del sistema subyacente. Las *herramientas de desarrollo de aplicaciones* se utilizan para construir interfaces de usuario; proporcionan herramientas gráficas que se pueden utilizar para construir interfaces ***sin programar***. Algunas de las herramientas de desarrollo de aplicaciones más famosas son PowerBuilder, Magic y Borland Delphi; Visual Basic también se utiliza bastante en el desarrollo de aplicaciones.
Además, ciertas aplicaciones como las hojas de cálculo y los paquetes de análisis estadístico utilizan la interfaz **cliente-servidor** directamente para acceder a los datos del servidor subyacente. De hecho, proporcionan interfaces visibles especiales para diferentes tareas.

Algunos sistemas de procesamiento de transacciones proporcionan una interfaz de **llamada a procedimientos remotos para transacciones** para conectar los clientes con el servidor. Estas llamadas aparecen para el programador como llamadas normales a procedimientos, pero todas las llamadas a procedimientos remotos hechas desde un cliente se engloban en una única transacción al servidor final. De este modo, si la transacción se cancela, el servidor puede deshacer los efectos de las llamadas a procedimientos remotos individuales.
Las arquitecturas de **sistemas servidores** pueden dividirse en servidores de transacciones y servidores de datos.

• Los sistemas **servidores de transacciones**, también llamados sistemas **servidores de consultas**, proporcionan una interfaz a través de la cual los clientes pueden enviar peticiones para realizar una acción que el servidor ejecutará y cuyos resultados se devolverán al cliente. Normalmente, las máquinas cliente envían las transacciones a los sistemas servidores, lugar en el que estas transacciones se ejecutan, y los resultados se devuelven a los clientes que son los encargados de visualizar los datos. Las peticiones se pueden especificar utilizando SQL o mediante la interfaz de una aplicación especializada.

• Los **sistemas servidores de datos** permiten a los clientes interaccionar con los servidores realizando peticiones de lectura o modificación de datos en unidades tales como archivos o páginas. Por ejemplo, los servidores de archivos proporcionan una interfaz de sistema de archivos a través de la cual los clientes pueden crear, modificar, leer y borrar archivos. Los servidores de datos de los sistemas de bases de datos ofrecen muchas más funcionalidades; soportan unidades de datos de menor tamaño que los archivos, como páginas, tuplas u objetos. Proporcionan facilidades de indexación de los datos, así como facilidades de transacción de modo que los datos nunca se quedan en un estado inconsistente si falla una máquina cliente o un proceso.

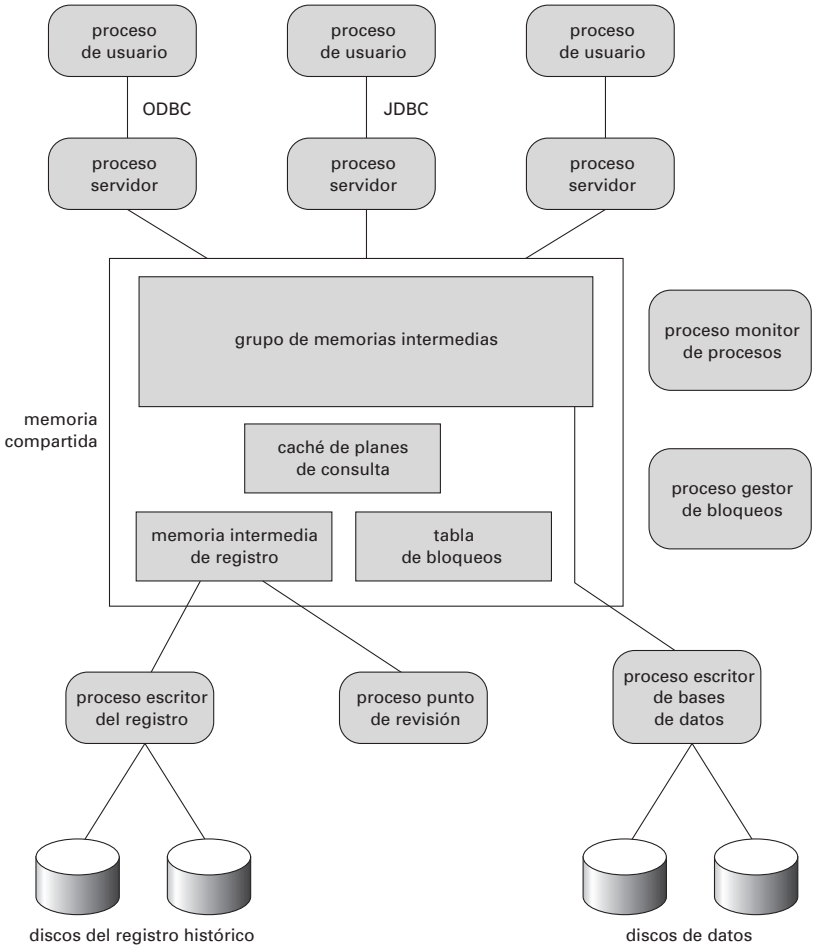



**Figura No 4 : Estructura de la memoria compartida y de los procesos.**

Las arquitecturas de sistemas **paralelos** mejoran la velocidad de procesamiento y de Entrada/Salida mediante la utilización de UCP y discos en paralelo. Cada vez son más comunes las máquinas paralelas, lo que hace que cada vez sea más importante el estudio de los sistemas paralelos de bases de datos. La fuerza que ha impulsado a los sistemas paralelos de bases de datos ha sido la demanda de aplicaciones que han de manejar bases de datos extremadamente grandes (del orden de terabytes, esto es, 1012 bytes) o que tienen que procesar un número enorme de transacciones por segundo (del orden de miles de transacciones por segundo). Los sistemas de bases de datos centralizados o cliente-servidor no son suficientemente potentes para soportar tales aplicaciones.
En el procesamiento paralelo se realizan muchas operaciones simultáneamente mientras que en el procesamiento secuencial, los distintos pasos computacionales han de ejecutarse en serie. Una máquina paralela de **grano grueso** consiste en un pequeño número de potentes procesadores; una máquina **masivamente paralela** o de **grano fino** utiliza miles de procesadores más pequeños. Hoy en día, la mayoría de las máquinas de gama alta ofrecen un cierto grado de paralelismo de grano grueso: son comunes las máquinas con dos o cuatro procesadores. Las computadoras masivamente paralelas se distinguen de las máquinas paralelas de grano grueso porque son capaces de soportar un grado de paralelismo mucho mayor. Ya se encuentran en el mercado computadoras paralelas con cientos de UCP y discos.
Para medir el rendimiento de los sistemas de bases de datos existen dos medidas principales:

(1) la **productividad**, número de tareas que pueden completarse en un intervalo de tiempo determinado, y (2) el **tiempo de respuesta**, cantidad de tiempo que necesita para completar una única tarea a partir del momento en que se envíe. Un sistema que procese un gran número de pequeñas transacciones puede mejorar la productividad realizando muchas transacciones en paralelo. Un sistema que procese transacciones largas puede mejorar el tiempo de respuesta así como la productividad realizando en paralelo las distintas sub-tareas de cada transacción.

**Arquitecturas paralelas de bases de datos**

Existen varios modelos de arquitecturas para las máquinas paralelas. En la Figura 5 se muestran algunos de los más importantes (en la figura, **M** quiere decir memoria, **P** procesador y los discos se dibujan como cilindros):
• **Memoria compartida.** Todos los procesadores comparten una memoria común (Figura 5a).
• **Disco compartido.** Todos los procesadores comparten un conjunto de discos común (Figura 5b). Algunas veces los sistemas de disco compartido se denominan **agrupaciones**.
• **Sin compartimiento**. Los procesadores no comparten ni memoria ni disco (Figura 5c).
• **Jerárquico**. Este modelo es un híbrido de las arquitecturas anteriores (Figura 5d).

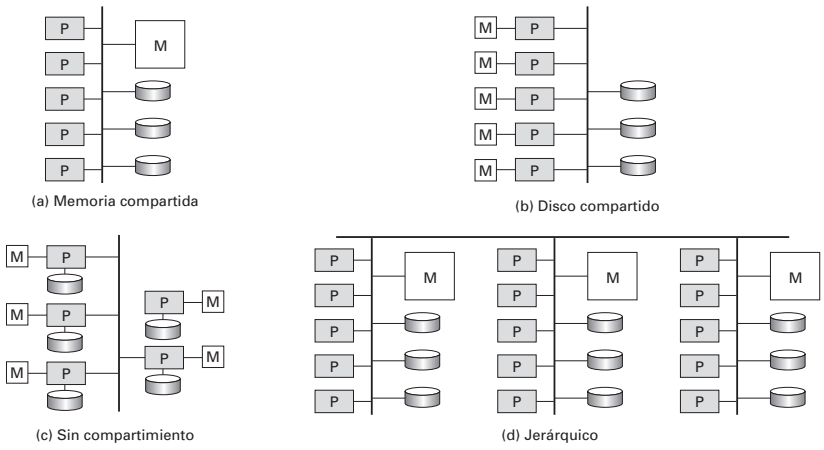



**Figura No 5 : Arquitecturas paralelas de bases de datos.**

En las arquitecturas de **sistema distribuido de bases** de datos se almacena la base de datos en varias computadoras. Varios medios de comunicación, como las redes de alta velocidad o las líneas telefónicas, son los que pueden poner en contacto las distintas computadoras de un sistema distribuido. No comparten ni memoria ni discos. Las computadoras de un sistema distribuido pueden variar en tamaño y función pudiendo abarcar desde las estaciones de trabajo a los grandes sistemas.
Dependiendo del contexto en el que se mencionen existen diferentes nombres para referirse a las computadoras que forman parte de un sistema distribuido, tales como **sitios** o **nodos**. Para enfatizar la distribución física de estos sistemas se usa principalmente el término **sitio**. En la Figura 6 se muestra la estructura general de un sistema distribuido.

Las principales diferencias entre las bases de datos paralelas sin compartimientos y las bases de datos distribuidas son que las bases de datos distribuidas normalmente se encuentran en varios lugares geográficos distintos, se administran de forma separada y poseen una interconexión más lenta. Otra gran diferencia es que en un sistema distribuido se dan dos tipos de transacciones, las locales y las globales. Una **transacción local** es aquella que accede a los datos del único sitio en el cual se inició la transacción. Por otra parte, una **transacción global** es aquella que, o bien accede a los datos situados en un sitio diferente de aquel en el que se inició la transacción, o bien accede a datos de varios sitios distintos.
Hay varias razones para construir sistemas distribuidos de bases de datos, incluyendo el compartimiento de los datos, la autonomía y la disponibilidad.
• **Datos compartidos.** La principal ventaja de construir un sistema distribuido de bases de datos es poder disponer de un entorno donde los usuarios puedan acceder desde una única ubicación a los datos que residen en otras ubicaciones. Por ejemplo, en un sistema de banca distribuida, donde cada sucursal almacena datos relacionados con dicha sucursal, es posible que un usuario de una de las sucursales acceda a los datos de otra sucursal. Sin esta capacidad, un usuario que quisiera transferir fondos de una sucursal a otra tendría que recurrir a algún mecanismo externo que pudiera enlazar los sistemas existentes.
• **Autonomía.** La principal ventaja de compartir datos por medio de distribución de datos es que cada ubicación es capaz de mantener un grado de control sobre los datos que se almacenan localmente. En un sistema centralizado, el administrador de bases de datos de la ubicación central controla la base de datos. En un sistema distribuido, existe un administrador de bases de datos global responsable de todo el sistema. Una parte de estas responsabilidades se delegan al administrador de bases de datos local de cada sitio. Dependiendo del diseño del sistema distribuido de bases de datos, cada administrador puede tener un grado diferente de **autonomía local**. La posibilidad de autonomía local es a menudo una de las grandes ventajas de las bases de datos distribuidas.
• **Disponibilidad.** Si un sitio de un sistema distribuido falla, los sitios restantes pueden seguir trabajando. En particular, si los elementos de datos están **replicados** en varios sitios, una transacción que necesite un elemento de datos en particular puede encontrarlo en varios sitios. De este modo, el fallo de un sitio no implica necesariamente la caída del sistema.
El sistema puede detectar el fallo de un sitio, y pueden ser necesarias acciones apropiadas para recuperarse del fallo. El sistema no debe seguir utilizando los servicios del sitio que falló. Finalmente, cuando el sitio que falló se recupera o se repara, debe haber mecanismos disponibles para integrarlo sin problemas de nuevo en el sistema.
Aunque la recuperación ante un fallo es más compleja en los sistemas distribuidos que en los sistemas centralizados, la capacidad que tienen muchos sistemas de continuar trabajando a pesar del fallo en uno de los sitios produce una mayor disponibilidad. La disponibilidad es crucial para los sistemas de bases de datos que se utilizan en aplicaciones de tiempo real. Que, por

ejemplo, una línea aérea pierda el acceso a los datos puede provocar la pérdida de potenciales compradores de billetes en favor de la competencia.

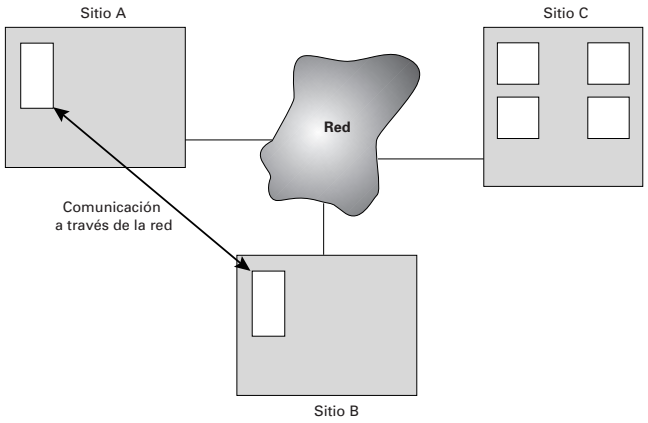



**Figura No 6 : Un sistema distribuido.**

Un ejemplo donde puede ver claramente los sistemas distribuidos es en el proceso de venta de tickets aéreos, donde la información se encuentran distribuida por todo el globo terráqueo en las diferentes aerolíneas y aeropuertos. Siendo cada uno de estos un nodo(sitio) del sistema distribuido. Pero el sistema de separación de sillas de vuelos, es un paralelo ya que su información es gigantesca, ya que el solo hecho de brindar escalas y cambio de un vuelo a otro exige esta arquitectura.

En este momento vamos a aprender lo que es realmente una base de datos, ya que acabamos de conocer las diferentes formas en que la información puede llegar a almacenarse, recuerde que el criterio que siempre estará de forma tacita es el tiempo de respuesta. Para adentrarnos en lo que son las capas de almacenamiento, debemos comprender lo que es una definición, las cuales se derivan de un criterio denominado concepto. Para el estudio del software, los **conceptos** son representaciones de la realidad, que nos ayudan a ambientarnos en los diferentes procesos, lo cual nos permite conocer algo. Este ***algo*** es la caracterización de lo que se conocerá mas adelante como una entidad. Veamos entonces, el kernel (corazón) de una base de datos es su información, la cual debe permitirme poder caracterizar un algo o algunos algo(s). Caracterizar es la capacidad de poder constituir una entidad (***es todo lo que se puede decir de algo***). Por lo tanto una entidad es un concepto plenamente caracterizable. Veamos un ejemplo:

Si estamos en un proceso donde se requiere caracterizar un paciente, y uno de sus criterios es la ubicación, tendremos que tener varios conceptos, o sea, ***realidades*** inmersas en esta definición. El criterio de la ubicación del paciente no es solo su dirección, también se incorporan características como: barrio, comuna, municipio y departamento. Pero algunos de estos conceptos no podrán ser entidades, serán características como todo concepto, pero no entidades.

Veamos el porque.

Recuerde la definición de entidad. Tendremos algo que decir de ***barrio*****?** R/ **Si…** podemos hablar de un código de barrio, también de un nombre de barrio, además del código de municipio del barrio. Como tenemos la capacidad de decir algo al respecto del barrio, entonces barrio es una entidad. Sucederá lo mismo con el concepto comuna**?** R/ **No…** no podemos decir nada al respecto de comuna, aunque usted podría pensar que si, que podríamos decir que la comuna esta compuesta por una serie de barrios, pero esto no es así ya que nosotros derivamos la composición de la comuna a razón de un criterio del barrio siendo este la comuna. Entonces

comuna aunque al igual que barrio es un concepto, no es una entidad ya que **no posee criterios propios.**

Bien ya comprendemos lo que es una entidad, ahora aprendamos a caracterizarla.

Pensemos en la entidad paciente, la cual tendrá atributos como id, nombres, apellidos, teléfono, dirección, celular, Rh, id_barrio. El conjunto de atributos(criterios que conforman una entidad), me permite definir el algo. El atributo se constituye en una característica cuando se le agrega el dato (contenido informatico) correspondiente, veamos:

| **Entidad : Paciente** | |
|---|---|
| **Atributo** | **Dato** |
| ID | 6.076.856 |
| Nombres | Pedro |
| Apellidos | Infante |
| Teléfono | 643 33 11 |
| Dirección | Calle 1 No 1 – 11 |
| Celular | 324 123 22 11 |
| Rh | O+ |
| Id_barrio | 0810 |
| **C A R A C T E R I Z A C I O N** | |

La entidad Paciente posee el atributo nombres al cual se le agrega el dato **Pedro** en este momento, **nombres → Pedro** se convierte en una característica.

El conjunto de características se denomina caracterización. Lo cual me permite constituir una entidad ya que me permite generar expresión de algo.

**L**a información se analiza, se cuantifica, se cualifica, se estructura, se normaliza se radica en entidades asociativas (entidad relación) para luego darle forma. Los datos que conforman la información van a incorporarse al interior de las entidades que conforman la base de datos, para eso esta información debe ser trabajada bajo los criterios de actualización, siendo estos :

**Definicion de CRUD**

1. **C**reate - (Registrar)
2. **R**ead - (Leer)
3. **U**pdate - (Modificar)
4. **D**elete - (Eliminar)

Estos criterios de actualización de base de datos se conocen como afectación **Transacional**, la cual afecta la base de datos de una manera continua, esta continuidad se conoce en el factor tiempo con el criterio - On Line (en línea - OL).
Las bases de datos sufren entonces una afectación en su contenido a razón de la actualización de datos, esta se conoce con la definición de OLTP (On Line Transaction Processing – Procesamiento Transacional En Línea). la definición de transacción, hace referencia a cada una de la afectaciones en registros que sufre o sufren las diferentes entidades que participan de la tarea de una actividad que desarrolla un proceso en una unidad de negocio. Ejemplo : Si

estamos realizando el ingreso de un nuevo paciente para ser atendido por urgencias, las entidades que se ven afectadas son, paciente, triage, consulta, formula, ordenes de servicio, etc.. no estamos causando una sola transacción, sino un conjunto de transacciones que se cuantifican de acuerdo al numero de registros causados en la base de datos, lo que significa que si esta actividad causo 14 registros, entonces la base de datos se afecto en 14 transacciones. Existe una técnica de administración del recurso de información al interior de la base de datos, esta técnica utiliza una herramienta que es un lenguaje, este lenguaje es conocido como SQL (Structured Query Language – Lenguaje de Consulta Estructurado), con el cual podemos realizar todo lo necesario para desarrollar el concepto de OLTP.

Ya conocemos lo que es una base de datos, ya entendemos que estas se construyen a razón de las realidades de las unidades de negocio y sus diferentes necesidades. También conocemos que la información que reposa en estas bases de datos nos permitirán tomar decisiones adecuadas con referencia a las diferentes actividades que se realizan cotidianamente, pero, existe una verdad absoluta y es que la compañía deberá de tomar decisiones no solo amparada en las situaciones de si misma, sino también teniendo en cuenta las diferentes tendencias y comportamientos de las realidades locales, regionales, nacionales e internacionales. Para esto, no solo nos deberemos basar en la información propia, sino también en información externa. Lo que nos obliga a tener en cuenta datos de otras compañías o tendencias. Al surgir estas necesidades, se incorporan requerimientos diferentes, que nos exige aprender a manipular información cuyos orígenes son heterogéneos(normalmente varias bases de datos OLTP), obligándonos a migrar a una base de datos homogénea separada, conocida como **almacén de datos**. Aquí los datos se organizan para facilitar consultas analíticas en lugar de procesamiento de transacciones. Debemos ser muy concientes que aquí toma fuerza el criterio **origen de datos**.

Veamos entonces :

Que una agencia de viajes colombiana, tiene la exigencia de atender a un cliente que tiene como necesidad, la obtención de un ticket de vuelo para Paris- Francia, esta compañía no cuenta con la información propia necesaria para satisfacer la demanda de este cliente, pero tiene la posibilidad de accesar a diferentes bases de datos de múltiples compañías aéreas del mundo, que tienen la información consignada en un gran almacén de datos, con el cual podrá conocer la gama de posibilidades para satisfacer la necesidad de este cliente. Por ejemplo, sabrá que puede vender dos vuelos de la siguiente forma.

Vuelo 1

Salida : 08:00 Cali, Colombia – Aeropuerto : Alfonso Bonilla Arago.
Llegada : 11:45 Miami, Estados Unidos - Aeropuerto : Miami International.
Línea Aérea : American Airlines AA920 Avión : Boeing 737-800.

Vuelo 2

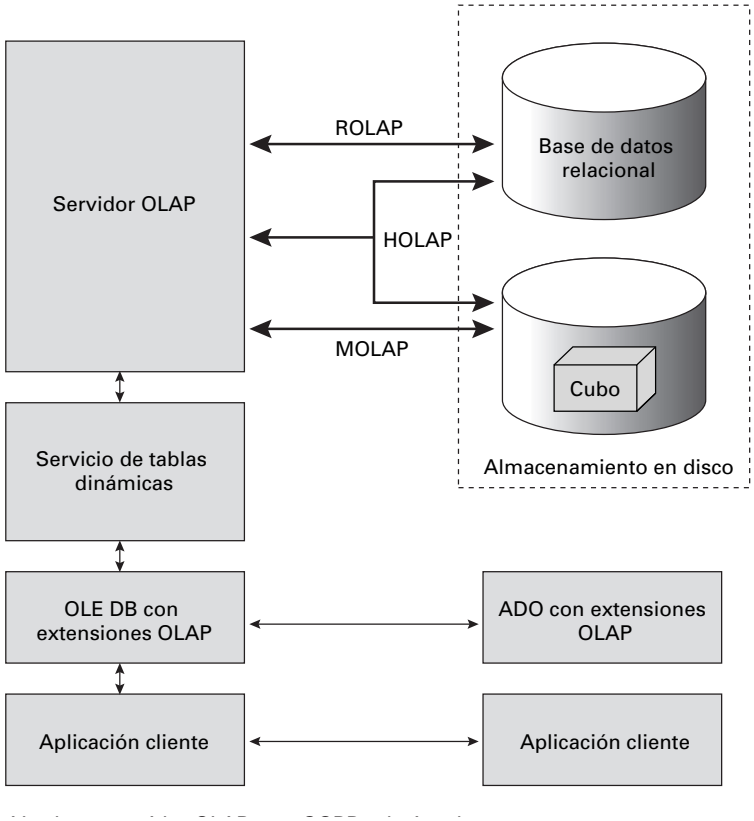
Salida : 11:45 Miami, Estados Unidos - Aeropuerto : Miami International.
Llegada : 09:00 + 1 día(s) Paris, Francia - Aeropuerto : Charles De Gaulle, Terminal 2A
Línea Aérea : British Airways BA1548 Avión : Boeing 767-300/300ER.

Y tengamos en cuenta que esta información no le pertenece a la agencia de vuelos, le pertenece a las aerolíneas American Airlines y British Airways. Pero la agencia de viajes no solo tiene acceso a consultar sus bases de datos, sino también a realizar afectaciones transacionales en línea en estas bases de datos ya homologadas para este proceso, ósea migradas a un gran almacén de datos. Esta forma de procesar información a razón del análisis en tiempo real de las necesidades existentes, se conoce como OLAP (Procesamiento analítico en línea).

Existe una gran diferencia de lo que conocemos como base de datos y un almacén de datos, la diferencia radica en su organización. Mientras que en la base de datos la informaron se organiza en registros, en el almacén de datos se organiza en cubos multidimensionales con información resumida, precalculada con el objetivo de proporcionar respuestas eficientes a consultas analíticas complejas. Debido a esto el objeto principal de OLAP es el cubo, el cual consiste en un origen de datos, dimensiones, medidas y divisiones. Un almacén de datos puede soportar muchos cubos distintos. Las consultas multidimensionales, en los cubos devuelven objetos de tipo conjunto de datos.

Los primeros sistemas de OLAP utilizaban arrays de memoria multidimensionales para almacenar los cubos de datos y se denominaban sistemas **OLAP multidimensionales (Multidimensional OLAP, MOLAP)**. Posteriormente, los servicios OLAP se integraron en los sistemas relacionales y los datos se almacenaron en las bases de datos relacionales. Estos sistemas se denominan sistemas **OLAP relacionales (Relational OLAP, ROLAP)**.

Los sistemas híbridos, que almacenan algunos resúmenes en la memoria y los datos básicos y otros resúmenes en bases de datos relacionales, se denominan sistemas **OLAP híbridos (Hybrid OLAP, HOLAP)**.




ahora bien ya conoce usted la diferencia existente entre las bases de datos y los almacenes de datos, lo fundamental es que los almacenes de datos trabajan con base a modelos, los cuales obedecen a una serie de patrones. Los tipos de patrones que los tomadores de decisiones intentan identificar incluyen asociaciones, secuencias, agrupamientos y tendencias.

- ✓ Las asociaciones son modelos que ocurren al mismo tiempo. Por ejemplo, una persona que compra cereal normalmente compra leche para acompañar el cereal.
- ✓ las sucesiones o secuencias son modelos de acciones que tienen lugar durante un periodo. Por ejemplo, si una familia compra una casa este año, probablemente comprarán muebles (un refrigerador o lavadora y secadora] el próximo año.
- ✓ El agrupamiento es el modelo que se desarrolla entre un grupo de personas. Por ejemplo, clientes que tienen un código postal particular podrían tender a comprar un automóvil particular.
- ✓ Las tendencias son modelos que se observan durante un periodo. Por ejemplo, los clientes podrían cambiar de comprar bienes genéricos a productos de marca.

Aquí les comparto un articulo publicado por *Moisés Naim el 20 de octubre de 2012* en el periódico el país de Cali.

**"P**ara Franklin D. Roosevelt fue la radio. Y para John F. Kennedy, la televisión. Para la primera elección de Barack Obama fue Internet y, en particular, Facebook. Es sabido que, en cada una de esas elecciones, una nueva tecnología contribuyó a la victoria del candidato que mejor la supo aprovechar. ¿Cuál será la innovación tecnológica que tendrá más peso en determinar el ganador de las próximas elecciones en EEUU? La respuesta es ***Data Mining***, la minería de datos, y más concretamente el ***microtargeting*** o la microsegmentación.

Entre los expertos existe el consenso de que, en este campo, el equipo de la campaña electoral del presidente lleva una gran ventaja al de Romney. Es el arma secreta de Obama, y sus principales asesores están convencidos de que, en una elección tan reñida como esta, la superioridad en el uso de estas tecnologías va a ser el factor determinante en su reelección.

La minería de datos es una rama de las ciencias de la información que utiliza complejos algoritmos y métodos estadísticos para identificar los patrones que puedan existir en las enormes bases de datos que hoy en día se acumulan gracias a las nuevas tecnologías. Se trata de convertir esa información en conocimiento útil para la toma de decisiones. En el mundo de la empresa privada, el ***Data Mining*** se usa hace tiempo y con gran sofisticación. Cuando usted entra en Internet y aparece una publicidad, es probable que su contenido resulte del uso de estas tecnologías. El mensaje específico es seleccionado de una lista de posibles anuncios, y la máquina escoge cuál enviarle a partir de un cálculo que se nutre de información sobre quién es usted (mujer, 37 años, casada con hijos, vive en la ciudad X, barrio Y), qué le gusta (ha comprado esto y aquello), qué hace (visita regularmente las páginas A y B en la Red) y la información extraída de una base de datos de personas con sus mismas características, gustos y hábitos. Todo esto revela los patrones más comunes sobre las motivaciones que determinan una decisión de compra en su segmento. Así, la publicidad que usted recibe, apunta a sus motivaciones, posibilidades y deseos. Esto es el *microtargeting*: apuntar micrométricamente no a un mercado, al público o a los votantes, sino a segmentos muy específicos dentro de esas categorías.

En el mundo de la política estas tecnologías se habían utilizado menos, pero ahora se han vuelto indispensables.

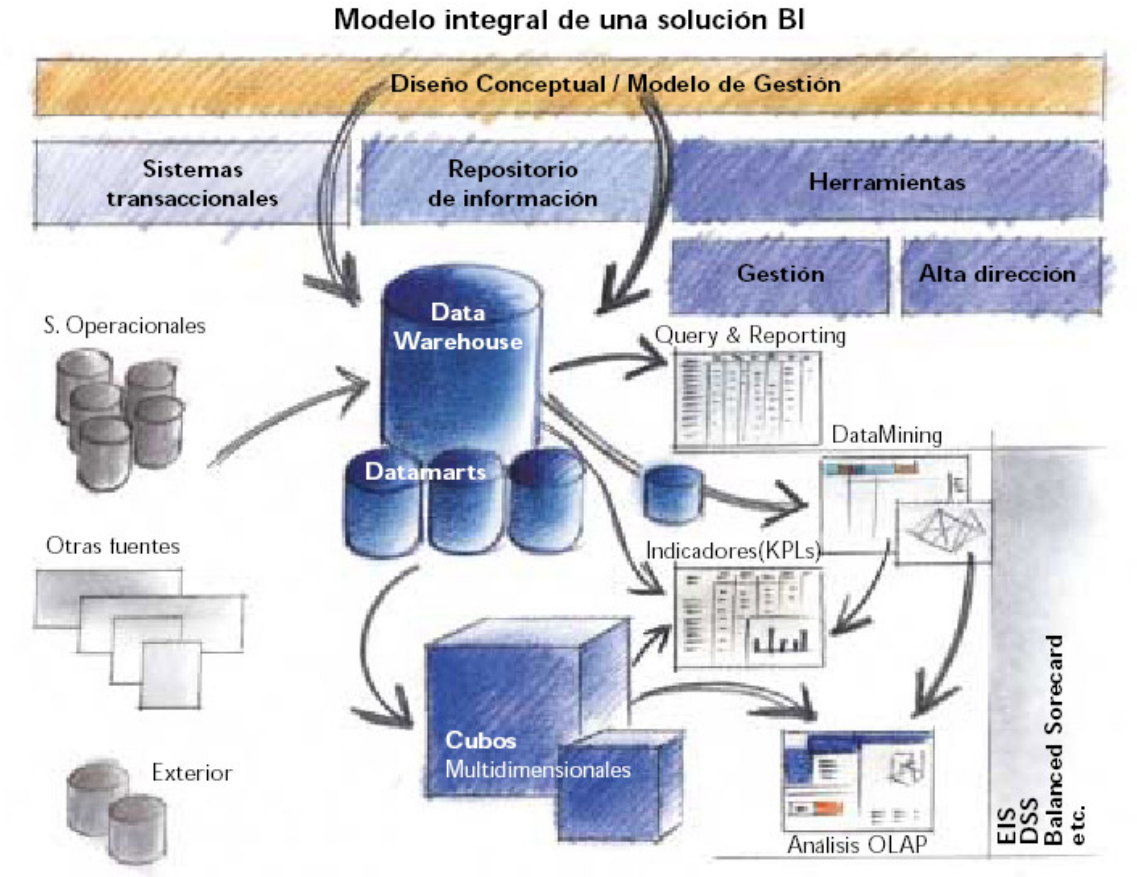
La ventaja de Obama en este campo se remonta a las elecciones primarias de 2007 y luego a su campaña presidencial de 2008. Su candidatura atrajo a un número sin precedentes de jóvenes, novatos en política pero magos en el uso de Internet. Terry McAuliffe, quien fuera el jefe del Partido Demócrata, me dijo: ***"Obama tiene a la mejor gente del mundo en el uso de Internet para campañas políticas. Lo sé porque lo sufrí en carne propia: ¡yo dirigí la campaña de Hillary en las primarias contra Obama! Eran extraordinarios. Y esa tecnología y esa gente ni siquiera son del partido. Son de la organización de Obama".***

Muchos de ellos son empleados a tiempo completo y provienen de empresas como Google, Facebook o Amazon. Actualmente, Harper Reed, antiguo hacker y exitoso vendedor de camisetas por Internet, dirige la operación de ***Data Mining*** de Obama. No da entrevistas y sus actividades se mantienen en secreto. Pero ha montado la más ambiciosa y eficiente estructura tecnológica para saber a quiénes acudir, qué decirles y qué pedirles (su voto, una donación, los votos de sus amigos y familiares, hacer llamadas telefónicas, un coche para llevar a la gente a votar, etcétera). De hecho, la tecnología les permite enviar mensajes distintos a dos personas de la misma familia y que viven en la misma casa.

En contraste, la campaña de Romney, que también hace un amplio uso de estas tecnologías, depende más de empresas privadas cuyos servicios el candidato utilizó con éxito en sus tiempos de empresario.

La tasa de desempleo, el dinero del que los candidatos disponen para su campaña y los SuperPacs, los comités de acción política que pueden dedicar enormes sumas de dinero a favor —o en contra— de Obama o Romney; los debates, la personalidad de los candidatos y su oferta electoral son algunos de los factores que van a influir en quién será el próximo presidente de EE UU. Y esta lista es aún más larga. Pero la capacidad para convertir información masiva y desordenada en conocimiento que aporta votos estará muy en el tope de esa lista."

Los almacenes de datos expresan la representación y la esencia de la tendencia a lo que hoy día se conoce como ***Business Intelligence (BI)***, lo cual suele definirse como la transformación de los datos de la compañía en conocimiento para obtener una ventaja competitiva (Gartner Group). Desde un punto de vista más pragmático, y asociándolo directamente a las tecnologías de la información, podemos definir Business Intelligence como el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada (interna y externa a la compañía) en información estructurada, para su explotación directa (reporting, análisis OLAP...) o para su análisis y conversión en conocimiento soporte a la toma de decisiones sobre el negocio.




**Componentes de una solución BI.**

**Diseño conceptual de los sistemas**. Para resolver el diseño de un modelo BI, se deben contestar a tres preguntas básicas: cuál es la información requerida para gestionar y tomar decisiones; cuál debe ser el formato y composición de los datos a utilizar; y de dónde proceden esos datos y cuál es la disponibilidad y periodicidad requerida. En otras palabras, el diseño conceptual tiene diferentes *momentos* en el desarrollo de una plataforma BI: En la fase de *construcción* del datawarehouse y datamarts, primarán los aspectos de estructuración de la información según potenciales criterios de explotación. En la fase de *implantación* de herramientas de soporte a la alta dirección, se desarrolla el análisis de criterios directivos: misión, objetivos estratégicos, factores de seguimiento, indicadores clave de gestión o KPIs, modelos de gestión... en definitiva, información para el qué, cómo, cuándo, dónde y para qué de sus necesidades de

información. Estos momentos no son, necesariamente, correlativos, sino que cada una de las etapas del diseño condiciona y es condicionada por el resto.

**Construcción y alimentación del datawarehouse y/o de los datamarts**. Un datawarehouse es una base de datos corporativa que replica los datos transaccionales una vez seleccionados, depurados y especialmente estructurados para actividades de query y reporting. Un datamart (*o mercado de datos*) es una base de datos especializada, departamental, orientada a satisfacer las necesidades específicas de un grupo particular de usuarios (en otras palabras, un datawarehouse departamental, normalmente subconjunto del corporativo con transformaciones específicas para el área a la que va dirigido).

## 1. BIBLIOGRAFIA.

✓ **FUNDAMENTOS DE BASES DE DATOS cuarta edición.**
Abraham Silberschatz - Bell Laboratories.
Henry F. Korth - Bell Laboratories.
S. Sudarshan - Instituto Indio de Tecnología, Bombay.

✓ **ANALISIS Y DISEÑO sexta edición.**
KENNETHE. KENDALL - Rutgers University School of Business-Camden Camden, New Jersey.
JULIEE. KENDALL - Rutgers University School of Business-Camden Camden, New Jersey.

✓ **ANALISIS Y DISEÑO DE SISTEMAS DE INFORMACION segunda edición.**
JAMES A. SENN. – Georgia State University.