Sistemas distribuidos

**** Panorama ****

1. Introducción
2. Definición
3. Características
4. Evolución
5. Cliente-Servidor
6. Protocolo
7. Middleware
8. Objetos distribuidos
9. Base de datos distribuida
10. Desarrollo WEB
11. Tecnologías Inalámbricas
12. Ventajas de los Sistemas Distribuidos
13. Desventajas de los Sistemas Distribuidos
14. Desafíos
15. Aplicaciones
16. Conclusiones
17. Referencias

1.- INTRODUCCIÓN

La computación desde sus inicios ha sufrido muchos cambios, desde las grandes computadoras que permitían realizar tareas en forma limitada y de uso un tanto exclusivo de organizaciones muy selectas, hasta las actuales computadoras ya sean personales o portátiles que tienen las mismas e incluso mayores capacidades que los primeros y que están cada vez más introducidos en el quehacer cotidiano de una persona.

Los mayores cambios se atribuyen principalmente a dos causas, que se dieron desde las décadas de los setenta:

1. El desarrollo de los microprocesadores, que permitieron reducir en tamaño y costo a las computadoras y aumentar en gran medida las capacidades de los mismos y su acceso a más personas.
2. El desarrollo de las redes de área local y de las comunicaciones que permitieron conectar computadoras con posibilidad de transferencia de datos a alta velocidad.

Es en este contexto que aparece el concepto de "Sistemas Distribuidos" que se ha popularizado tanto en la actualidad y que tiene como ámbito de estudio las redes como por ejemplo: Internet, redes de teléfonos móviles, redes corporativas, redes de empresas, etc.

Este documento presenta una panorámica de los aspectos relevantes que están involucrados en los Sistemas Distribuidos".

2.- SISTEMAS DISTRIBUIDOS (definición).

"Sistemas cuyos componentes hardware y software, que están en computadoras conectadas en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo preestablecido".

3.- CARACTERÍSTICAS.

- **Concurrencia**.- Esta característica de los sistemas distribuidos permite que los recursos disponibles en la red puedan ser utilizados simultáneamente por los usuarios y/o agentes que interactúan en la red.
- **Carencia de reloj global**.- Las coordinaciones para la transferencia de mensajes entre los diferentes componentes para la realización de una tarea, no tienen una temporización general, está más bien distribuida en los componentes.
- **Fallos independientes de los componentes.-** Cada componente del sistema pudiera fallar de manera independientemente, y los demás continuar ejecutando sus acciones. Esto permite el logro de las tareas con mayor efectividad, pues el sistema en su conjunto continua trabajando.

4.- EVOLUCIÓN.

Procesamiento central (Host).- Refiere a uno de los primeros modelos de computadoras interconectadas, llamados centralizados, donde todo el procesamiento de la organización se llevaba a cabo en una sola computadora, normalmente un Mainframe, y los usuarios empleaban sencillas computadoras personales.

Algunos problemas de este modelo son:

- Cuando la carga de procesamiento aumentaba se tenía que cambiar el hardware del Mainframe, lo cual es más costoso que añadir más computadores personales clientes o servidores que aumenten las capacidades.
- El otro problema que surgió son las modernas interfases gráficas de usuario, las cuales podían conllevar a un gran aumento de tráfico en los medios de comunicación y por consiguiente podían colapsar a los sistemas.

Grupo de Servidores.- Otro modelo que entró a competir con el anterior, también un tanto centralizado, son un grupo de computadoras actuando como servidores, normalmente de archivos o de impresión, poco inteligentes para un número de minicomputadores que hacen el procesamiento conectados a una red de área local.

Algunos problemas de este modelo son:

- Podría generarse una saturación de los medios de comunicación entre los servidores poco inteligentes y los minicomputadores, por ejemplo cuando se solicitan archivos grandes por varios clientes a la vez, podían disminuir en gran medida la velocidad de transmisión de información.

La Computación Cliente Servidor.- Este modelo, que predomina en la actualidad, permite descentralizar el procesamiento y recursos, sobre todo, de cada uno de los servicios y de la visualización de la Interfaz Gráfica de Usuario. Esto hace que ciertos servidores estén dedicados sólo a una aplicación determinada y por lo tanto ejecutarla en forma eficiente.

5.- CLIENTE-SERVIDOR

Definición:

Sistema en donde el cliente es una máquina que solicita un determinado servicio y se denomina servidor a la máquina que lo proporciona. Los servicios pueden ser:

- Ejecución de un determinado programa.
- Acceso a un determinado banco de información.
- Acceso a un dispositivo de hardware.

La presencia de un medio físico de comunicación entre las máquinas, es un elemento primordial, y dependerá de la naturaleza de este medio la viabilidad del sistema.

Categorías de Servidores:

A continuación se presenta una lista de los servidores más comunes:

- **Servidores de archivos.-** Proporciona archivos para clientes. Si los archivos no fueran tan grandes y los usuarios que comparten esos archivos no fueran muchos, esto sería una gran opción de almacenamiento y procesamiento de archivos. *El cliente solicita los archivos y el servidor los ubica y se los envía.*
- **Servidores de Base de Datos.-** Son los que almacenan gran cantidad de datos estructurados, se diferencian de los de archivos pues la información que se envía está ya resumida en la base de datos. Ejemplo: *El Cliente hace una consulta, el servidor recibe esa consulta (SQL) y extrae sólo la información pertinente y envía esa respuesta al cliente.*
- **Servidores de Software de Grupo.-** El software de grupo es aquel, que permite organizar el trabajo de un grupo. El servidor gestiona los datos que dan soporte a estas tareas. Por ejemplo: almacenar las listas de correo electrónico. *El Cliente puede indicarle, que se ha terminado una tarea y el servidor se lo envía al resto del grupo.*
- **Servidores WEB.**- Son los que guardan y proporcionan páginas HTML. *El cliente desde un browser o navegador hace un llamado de una página (link) y el servidor recibe el mensaje para después enviar la página solicitada.*
- **Servidores de correo.-** Gestiona el envío y recepción de correo de un grupo de usuarios (el servidor no necesita ser muy potente). El servidor sólo debe utilizar un protocolo de correo.
- **Servidor de objetos.-** Permite almacenar objetos que pueden ser activados de manera remota. *Los clientes pueden ser capaces de activar los objetos que se encuentren en el servidor.*
- **Servidores de impresión.-** Gestionan las solicitudes de impresión de los clientes. *El cliente envía la solicitud de impresión, el servidor recibe la solicitud y la ubica en la cola de impresión, ordena a la impresora que lleve a cabo las operaciones y luego avisa a la computadora cliente que ya acabo su respectiva impresión.*
- **Servidores de aplicación.-** En el pasado refería a un servidor que se dedicaba a una única aplicación. Era básicamente una aplicación a la que podían acceder los clientes. En la actualidad refiere más a un servidor Web con capacidad de procesamiento, por lo que suele ser a la vez servidor Web con algunas funciones de lógica de negocio.

Componentes de Software:

Se distinguen tres componentes básicos de software:

- **Presentación.-** Tiene que ver con la presentación al usuario de un conjunto de objetos visuales y llevar a cabo el procesamiento de los datos producidos por el mismo y los devueltos por el servidor.
- **Lógica de aplicación.-** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos.-** Esta compuesta de los archivos que contienen los datos de la aplicación.

Arquitecturas Cliente / Servidor

A continuación mostramos las arquitecturas cliente-servidor más populares:

- **Arquitectura Cliente-Servidor de Dos Capas.-** Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

    - Cuando se requiera poco procesamiento de datos en la organización.
    - Cuando se tiene una base de datos centralizada en un solo servidor.
    - Cuando la base de datos es relativamente estática.
    - Cuando se requiere un mantenimiento mínimo.

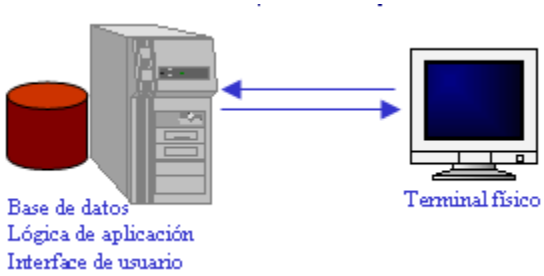
- **Arquitectura Cliente-Servidor de Tres Capas-** Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos. Normalmente esta arquitectura se utiliza en las siguientes situaciones:

    - Cuando se requiera mucho procesamiento de datos en la aplicación.
    - En aplicaciones donde la funcionalidad este en constante cambio.
    - Cuando los procesos no están relativamente muy relacionados con los datos.
    - Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
    - Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
    - Esta muy adecuada para utilizarla con la tecnología orientada a objetos.

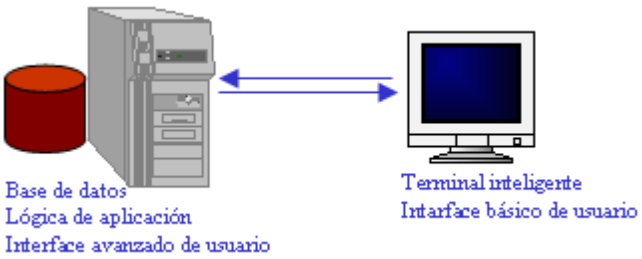
Clasificación de los sistemas cliente servidor:

A continuación mostramos la clasificación de de los sistemas cliente/servidor de acuerdo al nivel de abstracción del servicio que ofrecen:

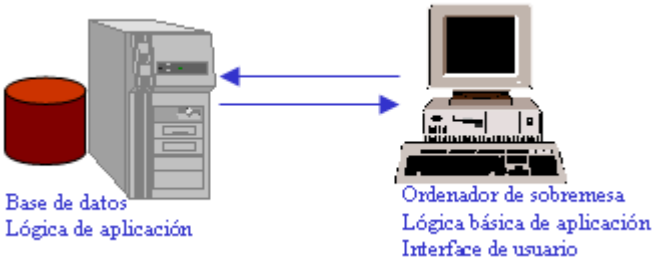
1. **Representación distribuida.-** La interacción con el usuario se realiza en el servidor, el cliente hace de pasarela entre el usuario y el servidor.



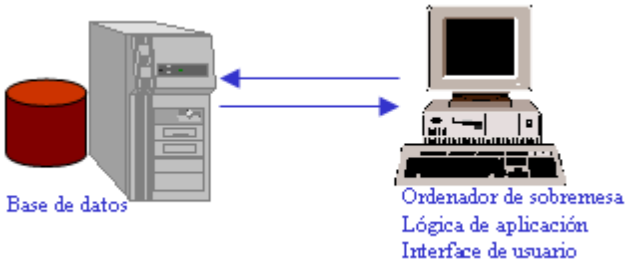
2. **Representación Remota.-**La lógica de la aplicación y la base de datos se encuentran en el servidor. El cliente recibe y formatea los datos para interactuar con el usuario.



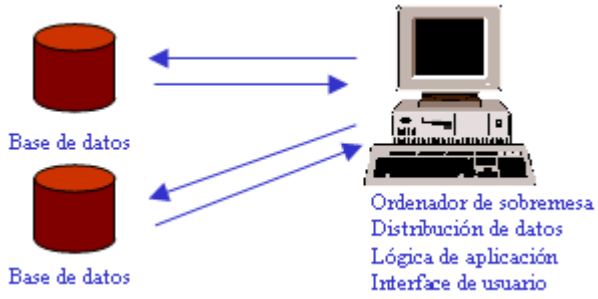
3. **Lógica Distribuida.-** El cliente se encarga de la interacción con el usuario y de algunas funciones triviales de la aplicación. Por ejemplo controles de rango de campos, campos obligatorios, etc. Mientras que el resto de la aplicación, junto con la base de datos, están en el servidor.



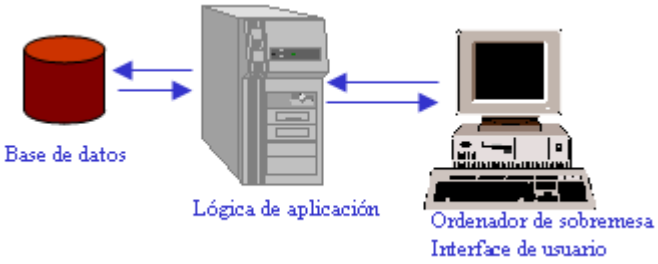
4. **Gestión Remota de Datos.-** El cliente realiza la interacción con el usuario y ejecuta la aplicación y el servidor es quien maneja los datos.



5. **Base de Datos Distribuidas.-** El cliente realiza la interacción con el usuario, ejecuta la aplicación, debe conocer la topología de la red, así como la disposición y ubicación de los datos. Se delega parte de la gestión de la base de datos al cliente.



6. **Cliente servidor a tres niveles.-** El cliente se encarga de la interacción con el usuario, el servidor de la lógica de aplicación y la base de datos puede estar en otro servidor.



7. **Niveles de una aplicación Web.** El ***nivel de interfaz*** *de usuario* está compuesto por las páginas HTML que el usuario solicita a un servidor Web y que visualiza en un cliente Web (normalmente, un navegador). El ***nivel de lógica*** *de negocio* está compuesto por los módulos que implementan la lógica de la aplicación y que se ejecutan en un servidor de aplicaciones. El ***nivel de datos*** está compuesto por los datos, normalmente gestionados por un sistema de gestión de bases de datos (servidor de datos), que maneja la aplicación web.

6.- PROTOCOLO

Definición:

Es un conjunto bien conocido de reglas y formatos que se utilizan para la comunicación entre procesos que realizan una determinada tarea. Se requieren dos partes:

- Especificación de la secuencia de mensajes que se han de intercambiar.
- Especificación del formato de los datos en los mensajes.

Un protocolo permite que componentes heterogéneos de sistemas distribuidos puedan desarrollarse independientemente, y por medio de módulos de software que componen el protocolo, haya una comunicación transparente entre ambos componentes. Es conveniente mencionar que estos componentes del protocolo deben estar tanto en el receptor como en el emisor.

Ejemplos de protocolos usados en los sistemas distribuidos:

- **IP: Protocolo de Internet.-** Protocolo de la capa de Red, que permite definir la unidad básica de transferencia de datos y se encarga del direccionamiento de la información, para que llegue a su destino en la red.

- **TCP: Protocolo de Control de Transmisión.-** Protocolo de la capa de Transporte, que permite dividir y ordenar la información a transportar en paquetes de menor tamaño para su transporte y recepción.
- **HTTP: Protocolo de Transferencia de Hipertexto.-** Protocolo de la capa de aplicación, que permite el servicio de transferencia de páginas de hipertexto entre el cliente WEB y los servidores.
- **SMTP: Protocolo de Transferencia de Correo Simple.-** Protocolo de la capa de aplicación, que permite el envío de correo electrónico por la red.
- **POP3: Protocolo de Oficina de Correo.-** Protocolo de la capa de aplicación, que permite la gestión de correos en Internet, es decir, le permite a una estación de trabajo recuperar los correos que están almacenados en el servidor.

7.- MIDDLEWARE

Definición:

Es un término que abarca a todo el software necesario para el soporte de interacciones entre Clientes y Servidores principalmente en aplicaciones distribuidas. Se puede considerar como el enlace que permite que un cliente obtenga un servicio de un servidor. Normalmente se define como una capa de software cuyo propósito es ocultar la heterogeneidad y proveer de un modelo de programación conveniente para los desarrolladores de aplicaciones. Se encuentra representado por procesos u objetos que actúan en un conjunto de computadoras y que se comunican con el fin de proporcionar soporte para compartición de recursos en un sistema distribuido

Los paquetes que soportan las llamadas a procedimientos remotos tal como los RPC de Sun y los sistemas de comunicaciones tales como ISIS son algunos de los pioneros del Middleware. En la actualidad se cuenta con muchos productos y estándares Middlewares que dan soporte a los sistemas orientados a objetos, entre ellos podemos señalar a : CORBA, Java RMI, Web Services, DCOM (Distributed Component Object Model) de Microsoft y los modelos de referencia para procesamiento distribuido abierto (RM-ODP) del ISO/ITU-T.

En general podemos mencionar dos tipos de middleware:

- **Software intermedio general**. Servicios generales que requieren todos los clientes y servidores, por ejemplo: software para las comunicaciones usando el TCP/IP, software parte del sistema operativo que, por ejemplo, almacena los archivos distribuidos, software de autenticación, el software intermedio de mensajes de clientes a servidores y viceversa.
- S**oftware intermedio de servicios**. Software asociado a un servicio en particular, por ejemplo: software que permite a dos BD conectarse a una red cliente/servidor (ODBC: Conectividad abierta de BD), software de objetos distribuidos, por ejemplo la tecnología CORBA permite que objetos distribuidos creados en distintos lenguajes coexistan en una misma red (intercambien mensajes), software intermedio para software de grupo, software intermedio asociado a productos de seguridad específicas (Conexiones Seguras: Sockets), etc.

Características:

- Independiza el servicio de su implantación, del sistema operativo y de los protocolos de comunicaciones.
- Permite la convivencia de distintos servicios en un mismo sistema.
- Permite la transparencia en el sistema.
- Modelo tradicional: Monitor de teleproceso o CICS, Tuxedo.

- Modelo OO: CORBA.

8.- OBJETOS DISTRIBUIDOS

Definición:

En los sistemas Cliente/Servidor, un objeto distribuido es aquel que esta gestionado por un servidor y sus clientes invocan sus métodos utilizando un "método de invocación remota". El cliente invoca el método mediante un mensaje al servidor que gestiona el objeto, se ejecuta el método del objeto en el servidor y el resultado se devuelve al cliente en otro mensaje.

Tecnologías orientadas a los objetos distribuidos:

Las tres tecnologías importantes y más usadas en este ámbito son:

1. **RMI.-** Remote Invocation Method.- Fue el primer framework para crear sistemas distribuidos de Java. El sistema de Invocación Remota de Métodos (RMI) de Java permite, a un objeto que se está ejecutando en una Máquina Virtual Java (VM), llamar a métodos de otro objeto que está en otra VM diferente. Esta tecnología está asociada al lenguaje de programación Java, es decir, que permite la comunicación entre objetos creados en este lenguaje.
2. **DCOM.**- Distributed Component Object Model.- El Modelo de Objeto Componente Distribuido, esta incluido en los sistemas operativos de Microsoft. Es un juego de conceptos e interfaces de programa, en el cual los objetos de programa del cliente, pueden solicitar servicios de objetos de programa servidores en otras computadoras dentro de una red. Esta tecnología esta asociada a la plataforma de productos Microsoft.
3. **CORBA.-** Common Object Request Broker Architecture.- Tecnología introducida por el Grupo de Administración de Objetos OMG, creada para establecer una plataforma para la gestión de objetos remotos independiente del lenguaje de programación.

9.- BASE DE DATOS DISTRIBUIDA

Definición:

Es una colección de datos (base de datos) construida sobre una red y que pertenecen, lógicamente, a un solo sistema distribuido, la cual cumple las siguientes condiciones:

- La información de la base de datos esta almacenada físicamente en diferentes sitios de la red.
- En cada sitio de la red, la parte de la información, se constituye como una base de datos en sí misma.
- Las bases de datos locales tienen sus propios usuarios locales, sus propios DBMS y programas para la administración de transacciones, y su propio administrador local de comunicación de datos.
- Estas base de datos locales deben de tener una extensión, que gestione las funciones de sociedad necesarias; la combinación de estos componentes con los sistemas de administración de base de datos locales, es lo que se conoce como Sistema Administrador de Base de Datos Distribuidas.
- Este gestor global permite que usuarios puedan acceder a los datos desde cualquier punto de la red, como si lo hicieran con los datos de su base de datos local, es decir, para el

usuario, no debe existir diferencia en trabajar con datos locales o datos de otros sitios de la red.

En consecuencia, la base de datos distribuida, es como una unidad virtual, cuyas partes se almacenan físicamente en varias bases de datos "reales" distintas, ubicadas en diferentes sitios.

Ejemplo de base de datos distribuida:

Considere un banco que tiene tres sucursales, en cada sucursal, una computadora controla las terminales de la misma y el sistema de cuentas. Cada computador con su sistema de cuentas local en cada sucursal constituye un "sitio" de la BDD; las computadoras están conectadas por la red. Durante las operaciones normales, las aplicaciones en las terminales de la sucursal necesitan sólo acceder la base de datos de la misma. Como sólo acceden a la misma red local, se les llaman aplicaciones locales.

Desde el punto de vista tecnológico, aparentemente lo importante es la existencia de algunas transacciones que acceden a información en más de una sucursal. Estas transacciones son llamadas transacciones globales o transacciones distribuidas. La existencia de transacciones globales será considerada como una característica que nos ayude a discriminar entre las BDD y un conjunto de base de datos locales.

Una típica transacción global sería una transferencia de fondos de una sucursal a otra. Esta aplicación requiere de actualizar datos en dos diferentes sucursales y asegurarse de la real actualización en ambos sitios o en ninguno. Asegurar el buen funcionamiento de aplicaciones globales es una tarea difícil.

Ventajas de las Base de Datos Distribuidas

- **Descentralización.-** En un sistema centralizado/distribuido, existe un administrador que controla toda la base de datos, por el contrario en un sistema distribuido existe un administrador global que lleva una política general y delega algunas funciones a administradores de cada localidad para que establezcan políticas locales y así un trabajo eficiente.
- **Economía:** Existen dos aspectos a tener en cuenta.
    - El primero son los costes de comunicación; si las bases de datos están muy dispersas y las aplicaciones hacen amplio uso de los datos puede resultar más económico dividir la aplicación y realizarla localmente.
    - El segundo aspecto es que cuesta menos crear un sistema de pequeñas computadoras con la misma potencia que un único computador.
- **Mejora de rendimiento**: Pues los datos serán almacenados y usados donde son generados, lo cual permitirá distribuir la complejidad del sistema en los diferentes sitios de la red, optimizando la labor.
- **Mejora de fiabilidad y disponibilidad:** La falla de uno o varios lugares o el de un enlace de comunicación no implica la inoperatividad total del sistema, incluso si tenemos datos duplicados puede que exista una disponibilidad total de los servicios.
- **Crecimiento:** Es más fácil acomodar el incremento del tamaño en un sistema distribuido, por que la expansión se lleva a cabo añadiendo poder de procesamiento y almacenamiento en la red, al añadir un nuevo nodo.
- **Flexibilidad:** Permite acceso local y remoto de forma transparente.
- **Disponibilidad:** Pueden estar los datos duplicados con lo que varias personas pueden acceder simultáneamente de forma eficiente. El inconveniente, el sistema administrador de base de datos debe preocuparse de la consistencia de los mismos.
- **Control de Concurrencia:** El sistema administrador de base de datos local se encarga de manejar la concurrencia de manera eficiente.

Inconvenientes de las base de datos distribuidas.

- El rendimiento que es una ventaja podría verse contradicho, por la naturaleza de la carga de trabajo, pues un nodo puede verse abrumado, por las estrategias utilizadas de concurrencia y de fallos, y el acceso local a los datos. Se puede dar esta situación cuando la carga de trabajo requiere un gran número de actualizaciones concurrentes sobre datos duplicados y que deben estar distribuidos.
- La confiabilidad de los sistemas distribuidos, esta entre dicha, puesto que, en este tipo de base de datos existen muchos factores a tomar en cuanta como: La confiabilidad de las computadoras, de la red, del sistema de gestión de base de datos distribuida, de las transacciones y de las tazas de error de la carga de trabajo.
- La mayor complejidad, juega en contra de este tipo de sistemas, pues muchas veces se traduce en altos gastos de construcción y mantenimiento. Esto se da por la gran cantidad de componentes Hardware, muchas cosas que aprender, y muchas aplicaciones susceptibles de fallar. Por ejemplo, el control de concurrencia y recuperación de fallos, requiere de personal muy especializado y por tal costoso.
- El procesamiento de base de datos distribuida es difícil de controlar, pues estos procesos muchas veces se llevan a cabo en las áreas de trabajo de los usuarios, e incluso el acceso físico no es controlado, lo que genera una falta de seguridad de los datos.

10.- DESARROLLO WEB

Caso particular de los sistemas Cliente-Servidor con representación remota. En donde se dispone de un protocolo estándar: HTTP y un Middleware denominado WebServer. En la actualidad la aplicación de sistemas informáticos basados en Internet, es una herramienta fundamental para las organizaciones que desean tener cierta presencia competitiva.

Tecnologías de la lógica de la aplicación en el servidor web:

- **CGI**: Common Gateware Interface..- Son programas que se ejecutan en el servidor, pueden servir como pasarela con una aplicación o base de datos o para generar documentos html de forma automática. Cada petición http ejecuta un proceso, el cual analiza la solicitud y genera un resultado. Son independientes del SO, y presentan la ventaja de que, dado un programa escrito en un lenguaje cualquiera, es fácil adaptarlo a un CGI. Entre los lenguajes que se usan para CGIs, el más popular es el Perl.
- **Servlets**: Pequeños programas en Java que se ejecutan de forma persistente en el servidor, y que, por lo tanto, tienen una activación muy rápida, y una forma más simple de hacerlo. Estos programas procesan una petición y generan la página de respuesta.
- **ASP** (Active Server Pages): Una página ASP es un fichero de sólo texto que contiene las secuencias de comandos, junto con el HTML necesario, y que se guarda con la extensión ".asp". Al ser llamado por el navegador, el motor ASP del IIS (Internet Information Server) se encarga automáticamente de ejecutarlo como se suele hacer con un programa cualquiera, pero cuya salida siempre será a través del navegador que le invoca. Es un entorno propietario de Microsoft y el lenguaje de secuencia de comandos predeterminado del IIS es el VBScript, aunque puede cambiarse.
- **JSP** (Java Server Pages), que consisten en pequeños trozos de código en Java que se insertan dentro de páginas web, de forma análoga a los ASPs. Ambas opciones, hoy en día, son muy populares en sitios de comercio electrónico. Frente a los ASPs, la ventaja que presentan es que son independientes del sistema operativo y del procesador de la máquina.

- **PHP** es un lenguaje cuyos programas se insertan también dentro de las páginas web, al igual que los ASPs y JSPs; es mucho más simple de usar, y el acceso a **bases de datos** desde él es muy simple. Es tremendamente popular en sitios de comercio electrónico con poco tráfico, por su facilidad de desarrollo y rapidez de implantación.

Consideraciones a tomar en el desarrollo de un sistema WEB

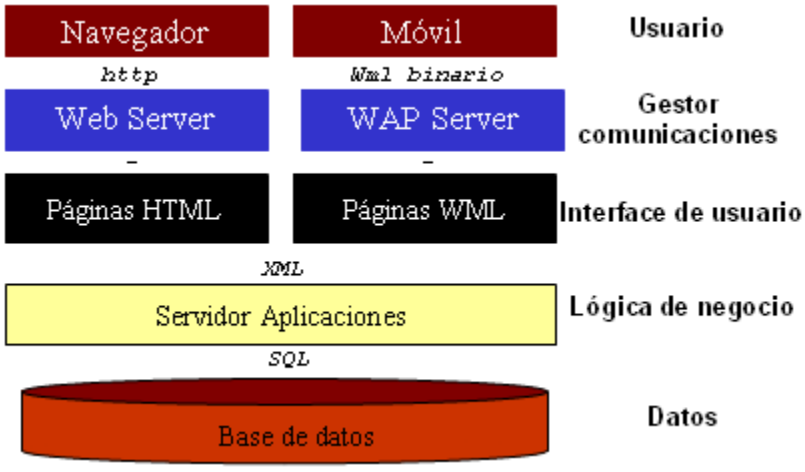
a. Separar la lógica de la aplicación de la interfase de usuario.
b. Utilizar métodos estándar de comunicación entre la lógica de aplicación y la interfase de usuario.
c. Herramientas que permitan una fácil adaptación de las aplicaciones a los nuevos dispositivos que irán apareciendo.
d. Definir el coste en comunicaciones que debe asumir la organización.
e. Tener en cuenta los procesos de réplica, periodicidad y el ancho de banda que consuman.
f. Replantear la idoneidad de la ubicación de cada proceso.
g. Extremar las pruebas al diseñar e implementar los protocolos de comunicación.

El concepto de servicio Web

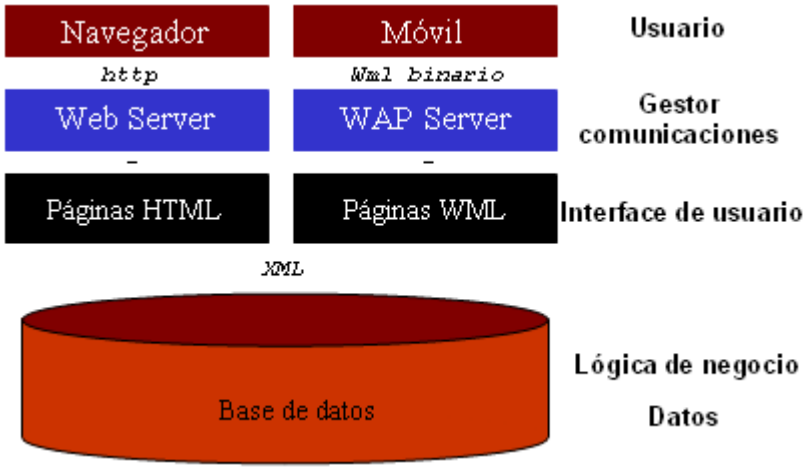
El servicio Web se ha definido como un componente de software reutilizable y distribuido que ofrece una funcionalidad concreta, independiente tanto del lenguaje de programación en que está implementado como de la plataforma de ejecución. Se puede considerar como aplicaciones auto-contenidas que pueden ser descritas, publicadas, localizadas e invocadas sobre la Internet (o cualquier otra red) y basada en estándares del W3C (especialmente XML).

Arquitecturas actuales de sistemas WEB.

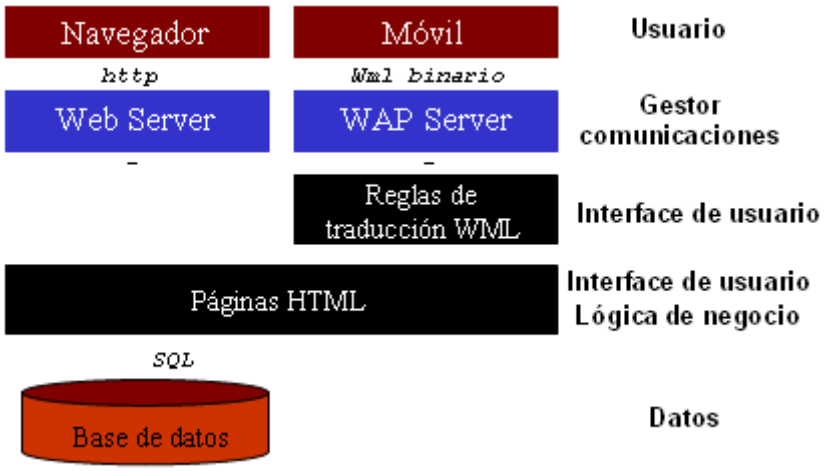
A continuación se muestran algunas arquitecturas que se presentan en nuestros días al momento de desarrollar una aplicación distribuida sobre la Web.



Variante de los fabricantes de Base de Datos



Variante de los fabricantes de pasarelas:



11.- TECNOLOGÍAS INALÁMBRICAS

Las tecnologías inalámbricas, en los últimos años, están alcanzando la madurez necesaria para permitir el acceso a una red, sin la necesidad de la utilización de los cables tradicionales de conexión.

A continuación mostramos un conjunto de tecnologías que contribuyen al desarrollo de las conexiones inalámbricas:

GSM (Global **S**ystem for **M**obile communications):

El sistema global para comunicaciones móviles, es un estándar para comunicación utilizando teléfonos móviles que incorpora tecnología digital. Permite utilizar el sistema SMS (servicio de mensajes cortos), para enviar y recibir mensajes de texto. Es la evolución tecnológica de los teléfonos móviles análogos.

GPRS (General Packet Radio Service**):**

Es un sistema de transmisión que funciona en el entorno de la telefonía móvil. En este sistema cada llamada de voz o cada conexión de datos, no ocupa de manera exclusiva un canal mientras dure esa llamada o conexión, por tanto, un usuario puede hacer uso de varios canales y un mismo canal puede ser compartido por varios usuarios. Esta basado en la conmutación de paquetes y permite la transmisión de datos a alta velocidad para el acceso a Internet.

UMTS (Universal Mobile Telecommunications System):

El Sistema Universal de Telecomunicaciones Móviles, permite disponer de banda ancha en telefonía móvil y transmitir un volumen de datos importante por la red. Con esta tecnología de tercera generación son posible las videoconferencias, descargar videos, el intercambio de postales electrónicas, paseos 'virtuales' por casas en venta, etc... todo desde el móvil.

WAP (Wireless Application Protocol)

El Protocolo de Aplicaciones Inalámbricas (WAP) es un servicio de mensajes digital inteligente para teléfonos celulares y otras terminales móviles que te permitirán visualizar contenidos de Internet en un formato de texto especial en un teléfono celular con tecnología GSM.

WAP se ha convertido en el estándar global para proveer información a las terminales inalámbricas.

WAP utiliza un microbrowser con un nuevo estándar llamado WML (similar al HTML) optimizado para terminales móviles inalámbricas.

WAP esconde la complejidad del GSM en las aplicaciones, así como la Web lo ha hecho para Internet. Expande una variedad de opciones de transporte y dispositivos, incluyendo SMS, 9.6 kbit/s GSM data y GPRS.

Bluetooth

Es la norma que define un estándar global de comunicación inalámbrica a cortas distancias, que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

La tecnología Bluetooth comprende hardware, software y requerimientos de interoperatividad.

WIFI (Wireless Fidelity):

Es la tecnología utilizada en una red o conexión inalámbrica, para la comunicación de datos entre equipos situados dentro de una misma área (interior o exterior) de cobertura.

Conceptualmente, no existe ninguna diferencia entre una red con cables (cable coaxial, fibra óptica, etc.) y una inalámbrica. La diferencia está en que las redes inalámbricas transmiten y reciben datos a través de ondas electromagnéticas, lo que supone la eliminación del uso de cables y, por tanto, una total flexibilidad en las comunicaciones.

WIMAX (Worldwide Interoperability for Microwave Access):

Es el nombre con el que se conoce la norma 802.16a, un estándar inalámbrico aprobado en enero del 2003 en el WiMax Forum, formado por un grupo de 67 compañías, que ofrece un mayor ancho de banda y alcance que la familia de estándares WiFi, compuesta por el 802.11a, 802.11b y 802.11g.

Como decimos, la diferencia entre estas dos tecnologías inalámbricas son su alcance y ancho de banda. Mientras que WiFi está pensado para oficinas o dar cobertura a zonas relativamente pequeñas, WiMax ofrece tasas de transferencia de 70mbps a distancias de hasta 50 kilómetros de una estación base. Por comparación, la tasa de transferencia de WiFi es de 11mbps y la distancia de hasta 350 metros en zonas abiertas.

12.- VENTAJAS DE LOS SISTEMAS DISTRIBUIDOS

Con respecto a Sistemas Centralizados:

- Una de las ventajas de los sistemas distribuidos es la **economía**, pues es mucho más barato, añadir servidores y clientes cuando se requiere aumentar la potencia de procesamiento.
- El **trabajo en conjunto**. Por ejemplo: en una fábrica de ensamblado, los robots tienen sus CPUs diferentes y realizan acciones en conjunto, dirigidos por un sistema distribuido.
- Tienen una **mayor confiabilidad.** Al estar distribuida la carga de trabajo en muchas máquinas la falla de una de ellas no afecta a las demás, el sistema sobrevive como un todo.
- Capacidad de **crecimiento incremental**. Se puede añadir procesadores al sistema incrementando su potencia en forma gradual según sus necesidades.

Con respecto a PCs Independientes:

- Se pueden **compartir recursos**, como programas y periféricos, muy costosos. Ejemplo: Impresora Láser, dispositivos de almacenamiento masivo, etc.

    1. Al compartir recursos, **satisfacen las necesidades de muchos usuarios a la vez**. Ejemplo: Sistemas de reservas de aerolíneas.
    2. Se logra una mejor comunicación entre las personas. Ejemplo: el correo electrónico.

Tienen mayor flexibilidad, la carga de trabajo se puede distribuir entre diferentes computadoras.

13.- DESVENTAJAS DE LOS SISTEMAS DISTRIBUIDOS

El principal problema es el software, el diseño, implantación y uso del software distribuido, pues presenta numerosos inconvenientes. Los *principales interrogantes* son los siguientes:

- ¿Qué tipo de S. O., lenguaje de programación y aplicaciones son adecuados para estos sistemas?.
- ¿Cuánto deben saber los usuarios de la distribución?.
- ¿Qué tanto debe hacer el sistema y qué tanto deben hacer los usuarios?.

La *respuesta a estos interrogantes no es uniforme* entre los especialistas, pues existe una gran *diversidad de criterios* y de *interpretaciones* al respecto.

Un aspecto primordial en este tipo de sistemas tiene que ver con las redes de comunicación. Por ejemplo: -Pérdida de mensajes, saturación en el tráfico, etc. Otro problema que puede surgir al compartir datos es la seguridad de los mismos. En general se considera que las ventajas superan a las desventajas, si estas últimas se administran seriamente.

14.- DESAFÍOS

- **Heterogeneidad de los componentes.-** La interconexión, sobre todo cuando se usa Internet, se da sobre una gran variedad de elementos hardware y software, por lo cual necesitan de ciertos estándares que permitan esta comunicación. Los Middleware, son elementos software que permiten una abstracción de la programación y el enmascaramiento de la heterogeneidad subyacente sobre las redes. También el middleware proporciona un modelo computacional uniforme.
- **Extensibilidad.**- Determina si el sistema puede extenderse y reimplementarse en diversos aspectos (añadir y quitar componentes). La integración de componentes escritos por diferentes programadores es un auténtico reto.
- **Seguridad.-** Reviste gran importancia por el valor intrínseco para los usuarios. Tiene tres componentes:
  - Confidencialidad.- Protección contra individuos no autorizados.
  - Integridad.- Protección contra la alteración o corrupción.
  - Disponibilidad.- Protección contra la interferencia con los procedimientos de acceso a los recursos.
- **Escalabilidad.-** El sistema es escalable si conserva su efectividad al ocurrir un incremento considerable en el número de recursos y en el número de usuarios.
- **Tratamiento de Fallos.-** La posibilidad que tiene el sistema para seguir funcionando ante fallos de algún componente en forma independiente. Para esto se tiene que tener alguna alternativa de solución. Técnicas para tratar fallos:
  - **Detección de fallos.** Algunos fallos son detectables, por ejemplo, usando comprobaciones.
  - **Enmascaramiento de fallos.** Algunos fallos detectados pueden ocultarse o atenuarse.
  - **Tolerancia de fallos.** Sobre todo en Internet se dan muchos fallos y no es muy conveniente ocultarlos, es mejor tolerarlos y continuar. Ej: Tiempo de vida de una búsqueda.
  - **Recuperación frente a fallos.** Tras un fallo se deberá tener la capacidad de volver a un estado anterior.
  - **Redundancia.** Se puede usar para tolerar ciertos fallos (DNS, BD, etc.)
- **Concurrencia.** Compartir recursos por parte de los clientes a la vez.
- **Transparencia.** Es la ocultación al usuario y al programador de aplicaciones de la separación de los componentes en un sistema distribuido. Se identifican 8 formas de transparencia:
  - **De Acceso.** Se accede a recursos locales y remotos de forma idéntica.
  - **De ubicación.** Permite acceder a los recursos sin conocer su ubicación.
  - **De concurrencia.** Usar un recurso compartido sin interferencia.
  - **De replicación.** Permite utilizar varios ejemplares de cada recurso.
  - **Frente a fallos.** Permite ocultar los fallos.
  - **De movilidad.** Permite la reubicación de recursos y clientes sin afectar al sistema.
  - **De prestaciones.** Permite reconfigurar el sistema para mejorar las prestaciones según su carga.
  - **Al escalado.** Permite al sistema y a las aplicaciones expandirse en tamaño sin cambiar la estructura del sistema o los algoritmos de aplicación.

15.- APLICACIONES DE LOS SISTEMAS DISTRIBUIDOS

- **Sistemas Comerciales.-** Inicialmente fueron construidos con hardware dedicado y entornos centralizados, son, por sus características de distribución geográfica y necesidad de acceso a sistemas distintos, ideales para implementarse en sistemas distribuidos. Requieren ciertas características de fiabilidad, seguridad y protección. Algunos ejemplos son:
  - Sistemas de reservas de líneas aéreas.
  - Aplicaciones bancarias.
  - Cajas y gestión de grandes almacenes.
- **Redes WAN.-** Debido al gran crecimiento de este tipo de redes (Internet), ha tomado gran importancia en el intercambio de información a través de la red. Y para esto tenemos los siguientes ejemplos:
  - Los servicios comunes que brinda Internet: Correo electrónico, servicio de noticias, transferencia de archivos, la World Wide Web, etc.
- **Aplicaciones Multimedia**.- Son las últimas incorporaciones a los sistemas distribuidos. Estas aplicaciones imponen ciertas necesidades de hardware para poder tener una velocidad y regularidad de transferencia de una gran cantidad de datos. Los ejemplos de estos sistemas son:
  - **Videoconferencia.**
  - **Televigilancia.**
  - **Juegos multiusuarios.**
  - **Enseñanza asistida por computadora.**
- **Áreas de la informática aplicada a los Sistemas Distribuidos.-** En este punto se tienen en cuenta toda la variedad de aplicaciones de los sistemas distribuidos, pues su diseño involucra a muchas áreas, por ejemplo:
  - Comunicaciones.
  - Sistemas operativos distribuidos.
  - Base de datos distribuidas.
  - Servidores distribuidos de archivos
  - Lenguajes de programación distribuidos.
  - Sistemas de tolerancia de fallos.

16.- CONCLUSIONES

- Los sistemas distribuidos abarcan una cantidad de aspectos considerables, sistemas operativos, comunicaciones, modelos de programación, etc, lo que hace que sus beneficios se pueden traducir en complejidades al momento de su implantación.
- Existen ciertos aspectos que requieren cuidado especial ya que pueden pasar de ser una ventaja a una desventaja, por ejemplo, el manejo de fallos, el control de la concurrencia, etc.
- Existen muchos temas de investigación relacionados con los sistemas distribuidos, en la sección de Desafíos se presentan algunos ejemplos.
- Es importante señalar que muchas tecnologías están en constante desarrollo y maduración, esto requiere de un estudio a profundidad de los factores que intervienen en cada aspecto de los sistemas distribuidos antes de apostar por alguna tecnología en especial.
- Es claro que la evolución constante en la tecnología sigue impulsando y estableciendo nuevos retos en el desarrollo de los sistemas distribuidos situación que se ve casi imposible de revertir

REFERENCIAS

Libros:

Distributed Systems: Concepts and Design
G. Coulouris, J. Dollimore, T. Kindberg,
Editorial: Addison Wesley,2005, 4th edition.
ISBN: 0321263545

Sistemas Distribuidos (español –versión anteior -)
George Coulouris; Jean Dollimore; Sebastián Dormido; Tim Kindberg
Editorial: Addison Wesley | 3era Edición
Idioma: Español
ISBN: 8478290494.

Distributed Systems: Principles and Paradigms**
Andrew S. Tanenbaum, Maarten van Oteen
Editorial: Prentice Hall; United States 2nd edition (Oct 2, 2006)
ISBN**:** 0132392275

Artículos:

Andrew S. Tanenbaum and Robbert Van Renesse, Distributed Operating Systems.
ACM Computing Surveys (CSUR), Volume 17, Issue 4. Pags. 419-470. ISSN:0360-0300.
The MIT Press scientific computation series. 1985.

Eliezer Levy and Abraham Silberschatz, Distributed file systems: concepts and examples.
ACM Computing Surveys (CSUR), Volume 22, Issue 4. Pags. 321-374. ISSN:0360-0300.
1990.