

## TALLER 3

# SISTEMAS DISTRIBUIDOS

PRESENTADO A:

CARLOS ALBERTO LONDOÑO LOAIZA

PRESENTADO POR:

ELKIN RAMIREZ

LORENA SANCHEZ

HERIBERTO DAVID

1. Realizar un cuadro comparativo con los tipos de computación descritas (monolítica, distribuida, paralela, cooperativa).

Tipos de Computaciones				
	Monolítica	Distribuida	Paralela	Cooperativa
Características	Es la forma de computación más sencilla entre todas. En la que solo se hace de un único ordenador ( una sola CPU) y la cual no está conectada a ninguna red y por lo tanto solo se puede utilizar los recursos que se tiene acceso de forma inmediata	La computación distribuida se basa en los sistemas distribuidos. En los cuales ordenadores interconectados independientemente a una red. Son capaces de colaborar entre sí para la ejecución de diferentes tareas (aclarar que los computadores son considerados como independientes cuando no comparten su memoria o el espacio de ejecución de programas.	Es la técnica de computación mediante la cual muchas instrucciones pueden ser ejecutadas de manera simultánea. Se basa en que los problemas grandes pueden ser divididos en problemas más pequeños para poder ser resueltos de forma concurrente, es decir en paralelo. Por lo general deben de tener el mismo sistema operativo. Mismos tiempos de sincronización y capacidades de hardware igual.	Es un tipo de computación hibrida entre la programación distribuida y paralela la cual se encuentra actualmente dirigida por proyectos de investigación, como es el caso de búsqueda de inteligencia extraterrestre (SETI Seachr For Extreaterrestrial intelligence)
Ventajas	Funcionan muy rápido Fácil desarrollo Precisa cierta potencia de proceso	Economía Trabajo en conjunto Tienen mayor confiabilidad Capacidad de crecimiento Se pueden compartir recurso Mayor comunicación entre las personas del grupo e trabajo	Ahorro de tiempo y dinero Se pueden solucionar problemas muy grandes y que no son posibles resolver de forma convencional Se mejora la eficiencia al dividir los problemas en tareas más pequeñas	Dividen la computación a larga escala entre las estaciones de trabajo de las máquinas de internet
Desventajas	Requiere mayor hardware Son infinitamente más lentas en peticiones sencillas Requiere habilitar acceso Su actualización es más costosa Ocupan mayor ancho de banda, provocando congestionamientos de red local	Diseño e implementación del sistema Mayor mantenimiento del sistema Los sistemas distribuidos son más complejos los sistemas impredecibles	Encontrar la solución a un problema tiene una complejidad mayor Se utilizan más recursos de las maquinas Pueden existir problemas de sincronización	Poca comunicación con los colaboradores.

2. Explicar el concepto de la programación concurrente basada en hilos y los mecanismos existentes para evitar las condiciones de carrera entre ellos.

Se define como programación concurrente a una rama de la informática que trata las técnicas de programación que se usan para expresar el paralelismo

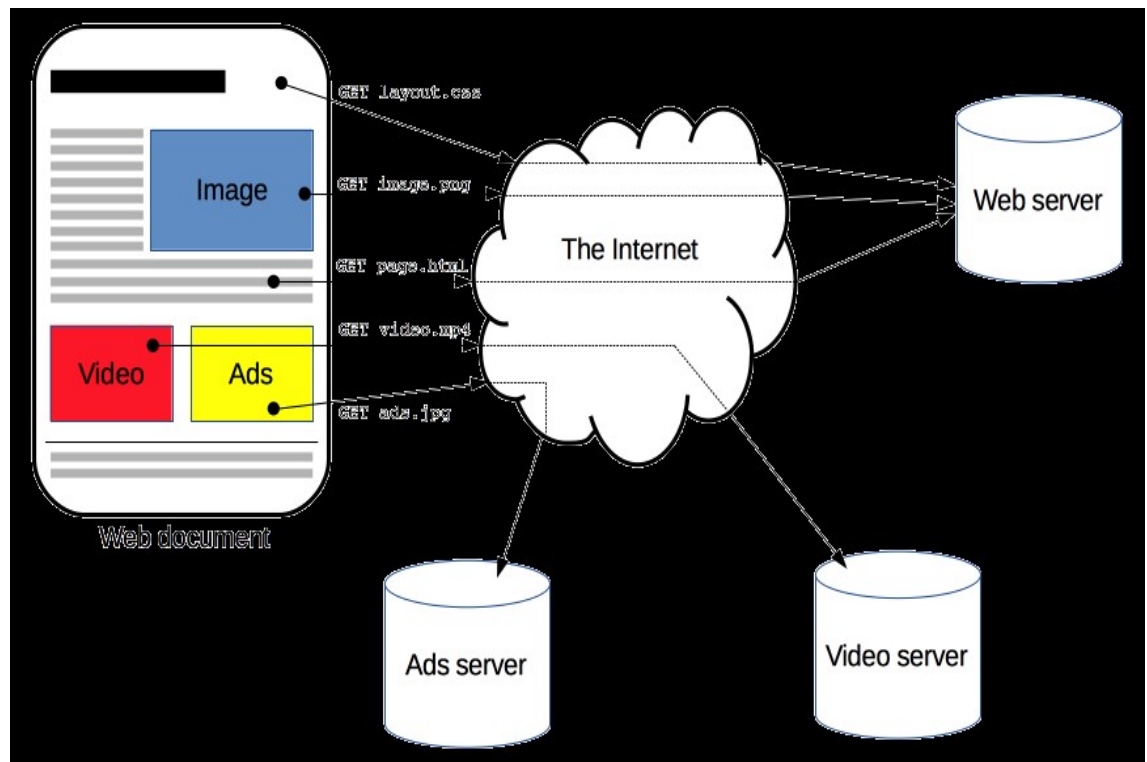
entre diversas tareas, y también para resolver problemas de comunicación y sincronización entre diversos procesos.

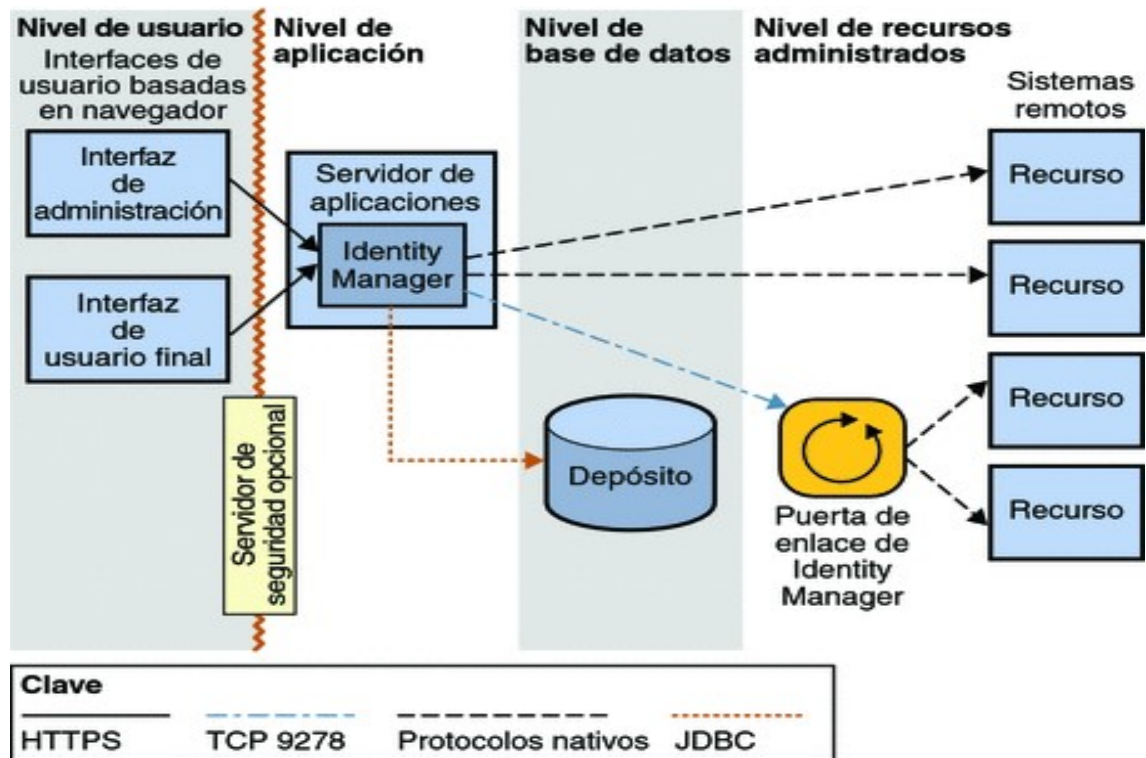
Uno de los principales problemas de la programación concurrente es el no saber puntualmente el orden en que se ejecutan los programas. (Especialmente para programas de comunicación de información).

Los hilos en la programación concurrente cumplen un papel importante puesto que son un pilar para poder trabajarla ayudando así a que haya más procesos ejecutándose a la vez, pero esto de alguna manera no es real, puesto que esto depende realmente del número de procesadores de la máquina en que se esté trabajando. Trabajar con hilos en programación concurrente no garantiza que sea un trabajo de programación en paralelo, sabemos que sincronizar varios hilos no es tan sencillo. Puesto que el problema es que todos puedan acceder a la misma información, pero en un orden diferente y esto puede bloquear la aplicación o llevarla a un estado inválido; Para evitar estos problemas utilizamos JOIN sobre cada hilo para esperar que una tarea termine.

**3. Realizar un gráfico que permita evidenciar el funcionamiento de tres protocolos básico de internet, tales como HTTP, SSH, FTP, SMTP, HTTPS, e indique el puerto por defecto donde se ofrece dicho servicio y cuál es el esquema de mensajes necesario entre el cliente y el servidor para implementado, describiendo los intercambios de mensajes con diagramas de eventos y secuencia.**

Protocolos			
protocolo	Descripción	Puerto de funcionamiento	Funcionamiento
SSH	Secure Shell	22/TCP	<p><b>SSH</b> (o <i>Secure SHell</i>) es el nombre de un protocolo y del programa que lo implementa cuya principal función es el acceso remoto a un servidor por medio de un canal seguro en el que toda la información está cifrada. Además de la conexión a otros dispositivos, SSH permite copiar datos de forma segura (tanto archivos sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir contraseñas al conectar a los dispositivos y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH y también puede redirigir el tráfico del (Sistema de Ventanas X) para poder ejecutar programas gráficos remotamente. El protocolo TCP asignado es el 22.</p> <p>El Protocolo seguro de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol Secure o HTTPS), es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hipertexto, es decir, es la versión segura de HTTP.</p>
HTTPS	Protocolo seguro de transferencia de hipertexto	443/TPC	<p>El sistema HTTPS utiliza un cifrado basado en la seguridad de textos SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.</p> <p>El puerto estándar para este protocolo es el 443. (También comúnmente usado el 4433)</p>
HTTP	Hypertext Transfer Protocol	80-(22(TCP))	<p>HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc</p>





4. ¿Qué diferencia existe entre la creación de múltiples procesos y la creación de múltiples hilos?, ¿qué información comparten unos y otros si los crea el mismo proceso padre?, ¿cuál satura más al computador?

Los hilos y los procesos en la programación concurrente es algo que es muy utilizados y que van de la mano, puesto que en un proceso pueden correr varios hilos y cada uno de ellos se ocupara de una función.

Puesto que un hilo puede definirse como cada secuencia de control dentro de un proceso que ejecuta instrucciones de forma independiente.

En el trabajo con múltiples hilos y múltiple proceso puede ocurrir concurrencia a dos niveles, un entre procesos y otra entre hilos de un mismo proceso.

Según lo conocido es más fácil que se pueda saturar al computador por la creación de múltiple proceso más que por la creación de múltiples hilos.

Teniendo en cuenta que los múltiples procesos son entidades pesadas, esto quiere decir que están más ligadas a la parte del núcleo del procesador, por lo contrario que la parte de los hilos está más ligada al espacio del usuario del sistema (los hilos comparten código, datos, etc.)

5. ¿Qué es el middleware en un sistema distribuido?, ¿cuál es su función?, ¿cómo opera?, Realice un gráfico que muestre las diferentes capas de un sistema distribuido e indique sus nombres y funciones.

El middleware es un software que ofrece varias funciones y proporciona servicios a las aplicaciones, fuera de lo que ofrece el sistema operativo. Cualquier software entre el kernel y las aplicaciones de usuario puede ser middleware. Es un software que permite la conexión entre dispositivos.

El middleware es importante en el desarrollo simple de aplicaciones y sistemas, puesto que proporcionan de forma sencilla la abstracción mediante la enmascararían e igualdad de redes de comunicaciones.

El Middleware tiene la capacidad de conectar distintos sistemas y facilitar la interacción entre un cliente y cualquier aplicación que provee un servicio arquitectura cliente - servidor, independientemente de la plataforma. Por lo tanto, se trata de un software de capa alta que se encuentra encima de la red física y las capas de transporte que se ubica entre el sistema operativo y las aplicaciones, manejando todas las tareas complejas requeridas para proveer el acceso a datos y aplicaciones entre plataformas.

Middleware permite diferentes modos de interacción como: invocaciones síncronas, paso de mensajes asíncrono, la coordinación a través de objetos compartidos entre otras. Las siguientes son las funciones principales de los middlewares en los sistemas distribuidos:

- Ocultación de la distribución, es decir, el hecho de que una aplicación se compone generalmente de muchas partes interconectadas que se ejecutan en ubicaciones distribuidas.
- Cómo ocultar la heterogeneidad de los componentes de hardware, sistemas operativos y protocolos de comunicación. Proporcionar uniformes, estándares, interfaces de alto nivel a los desarrolladores de aplicaciones e integradores, por lo que las aplicaciones pueden ser fácilmente integradas, reutilizadas, adaptadas, y hechas para inter operar.
- Suministrar un conjunto de servicios comunes a las diversas funciones de uso general, con el fin de evitar la duplicación de esfuerzos y para facilitar la colaboración entre las aplicaciones.

Entre los aspectos importantes de destacar en los middlewares podemos ver continuación:

Sirven como Balanceadores de carga, puesto que, si un server está muy cargado de procesos y otro esta sin trabajo, el balanceador e cargas le asignara trabajo al servidor libre para poder dar mayos balance y rendimiento al trabajo y procesos.

Sirven para realizar mantenimiento a Servidores, puesto que si hay procesos corriendo en un server que necesita un mantenimiento o una actualización. Es posible realizar una migración a otro server y así desconectar de la estación el primer server.

Los middlewares priorizan los trabajos que están realizando.

**6. Las funciones de envío y recepción necesitan sincronizarse entre sí, y para ello surgen tres posibilidades: crear funciones bloqueantes, aplicar técnicas de polling y callback. Explique cada técnica.**

Los procesos remotos que deben comunicarse entre sí hacen uso de funciones específicas de envío, recepción, etc. Que producen eventos. En general es necesario que los eventos sigan una ruta específica, que permita que los procesos se entiendan, un orden concreto de eventos que forma parte de las reglas de comunicación que conforma el protocolo. Ejemplo comunicación entre dos persona. Los procesos deben sincronizar sus eventos, pero surge un problema. Los procesos remotos no tienen ni idea que sucede en el otro extremo, solo saben aquello que les llegue en un mensaje, pero no tienen forma de saber cuándo llevar a cabo un evento para sincronizarlo con otro evento asociado entre otro proceso las técnicas son.

Eventos bloqueantes. Es aquel que produce que el proceso que involucra pasa a estado bloqueado, a la espera de que otros procesos remotos realice el envío de una respuesta ante tal evento. Mientras esa respuesta no llega, el proceso no puede continuar su ejecución, lo que permite sincronizar los procesos remotos, haciendo que uno espere al otro, ejemplo llamadas con retardo.

En el mecanismo más básico pero puede causar interbloqueos, que consiste en que dos procesos realizan un evento que los bloquea simultáneamente, en espera de otro procesos, que al estar también bloqueado produce una espera indefinida del uno por el otro y viceversa.

#### POLLING

Cosiste en que un proceso se sincroniza con otro ejecutando de forma recurrente un evento concreto. De forma que solo puede salir de ese invocándose cuando el proceso remoto envía la respuesta esperada. El resultado es el mismo que cuando se produce un bloqueo: el proceso queda encerrado en un bloqueo de instrucciones maquina a la espera de un evento externo pero el programador del proceso puede incluir en dicho bucle instrucciones necesarias que el proceso bloqueado pueda aprovechar para realizar en lugar de bloquearse.

#### CALLBACK

Consiste en que el proceso A le indica al proceso remoto B una forma de ponerse en contacto con el cuándo dicho proceso remoto disponga de los datos requeridos. De esta forma el proceso remoto B cuando tiene la información que le ha pedido el proceso A preparada, puede pasar a invocar ciertas instrucciones del procesos A. de esta forma el proceso A puede continuar con su trabajo sin esperar para sincronizarse con el B y aquellas tareas las cuales sea necesaria dicha sincronización se ejecutan cuando sea posible. Ejemplo personas que salen juntas para polling y callback.



**7. Un problema grave de los sistemas distribuidos consiste en los interbloqueos entre procesos, que ocurren cuando dos o más procesos se esperan mutuamente. El uso de temporizadores e hilos o procesos hijos para invocar funciones bloqueantes es una buena opción, explique en qué consiste cada técnica.**

Solución a interbloqueos: temporizadores y eventos no bloqueantes. Una de las posibles soluciones a este problema en sistemas distribuidos es:

#### Creación de hilos

Los hilos y los procesos en la programación concurrente es algo que es muy utilizados y que van de la mano, puesto que en un proceso pueden correr varios hilos y cada uno de ellos se ocupara de una función.

Puesto que un hilo puede definirse como cada secuencia de control dentro de un proceso que ejecuta instrucciones de forma independiente.

En el trabajo con múltiples hilos y múltiple proceso puede ocurrir concurrencia a dos niveles, un entre procesos y otra entre hilos de un mismo proceso.

Según lo conocido es más fácil que se pueda saturar al computador por la creación de múltiple proceso más que por la creación de múltiples hilos.

Teniendo en cuenta que los múltiples procesos son entidades pesadas, esto quiere decir que están más ligadas a la parte del núcleo del procesador, por lo contrario que la parte de los hilos está más ligada al espacio del usuario del sistema (los hilos comparten código, datos, etc.)

#### Creación de un temporizador

Indica a un programa el tiempo máximo de espera antes de abortar una tarea o función. Uno de los usos más frecuentes del control de procesos mediante temporizadores es para abandonar procesos que deberían acabar normalmente pero que por cualquier circunstancia no prevista o incontrolada no se acaba. La idea básica es que en situaciones en que un sistema debe esperar a que suceda algo, en lugar de esperar por tiempo indefinido, la espera se cancelará después de que el tiempo de espera haya transcurrido.

#### Operaciones asíncronas

Las operaciones que si no provocan bloqueo del proceso que las invoca se denominan de tal nombre asíncronas.

**8. Existen cuatro posibilidades de envío y recepción en cuanto el tipo de eventos bloqueantes o no bloqueantes (síncrono - síncrono, síncrono - asíncrono, asíncrono -asíncrono, asíncrono - síncrono). Explique cada uno de ellos.**

Las funciones síncronas y asíncronas Las funciones que conllevan a eventos bloqueantes se denominan síncronas, y las que conllevan un proceso de bloqueo se llaman asíncronas.

Envío síncrono y recepción síncrona.

Que los envíos se invoquen antes que la recepción.  
Que la recepción se invoque antes que el envío.

Envíos y recepción síncronos  
Que los envíos se invoquen antes que la recepción.  
Que la recepción se invoque después que el envío.

**9. ¿Qué es un socket?, explique los dos tipos de sockets que existen.**

un socket es programación es un túnel de comunicación que ayuda a que 2 aplicaciones se comuniquen, los sockets son la base del internet y de sus protocolos como HTTP, FTP .SMTP, etc

Podemos utilizar los sockets para sacarle provecho. Es decir, mediante sockets podemos hacer que dos aplicaciones se comuniquen entre sí.

Tipos de sockets

**Tipo sock\_dgram:** son sockets para comunicación de modo no conectado, con el envío de datagramas de tamaño limitado (tipo telegrama).

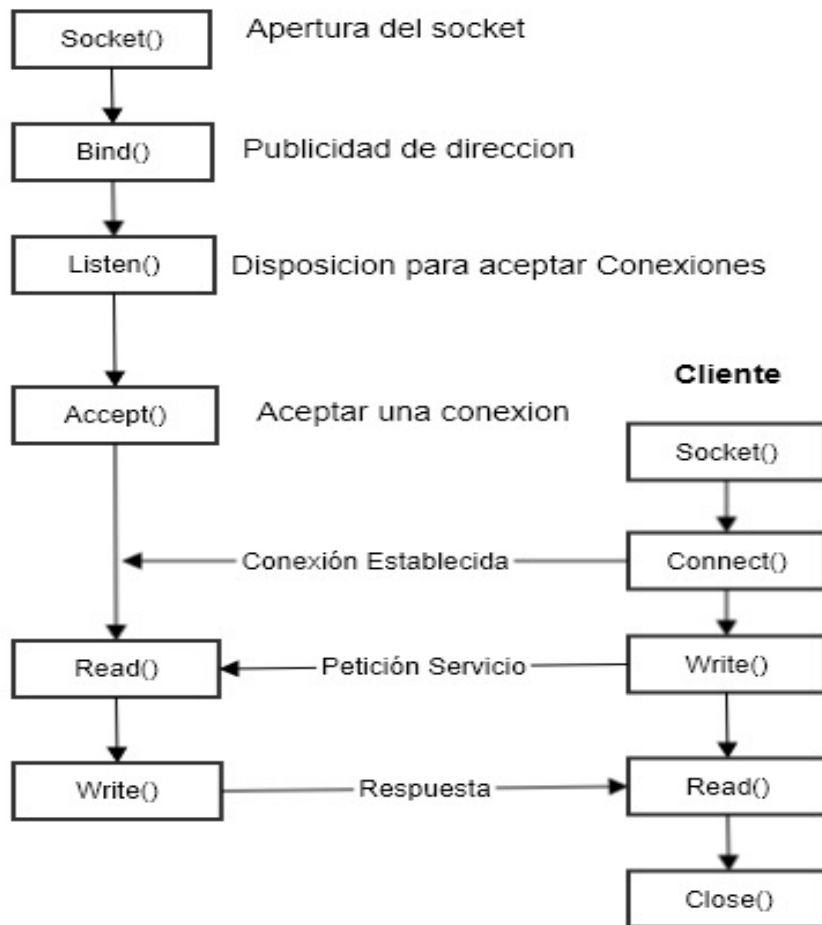
**Tipo Sock\_Stream:** para comunicaciones fiables en modo conectado. De dos vías y con tamaño variable de los mensajes de datos. Por debajo en dominios de internet. Subyacentes al protocolo TCP.

**Tipo sock\_raw:** Permite el acceso a protocolos de más bajo nivel como el IP (NIVEL DE RED).

**Tipo sock\_seqpacket:** Tiene las características del sock\_stream, pero además el tamaño de los mensajes es fijo.

**10. Los sockets permiten hasta seis operaciones básicas, dependiendo de su tipo: bind, aceptar solicitudes de conexión, conectar con otro socket, enviar, recibir y conectar. Evidencie mediante un gráfico como se presente esta conexión entre dos equipos.**

## Servidor



11. La sincronización de operaciones con sockets puede provocar interbloqueos. Indique dos posibles soluciones o algoritmos para este problema.

Una solución posible Seria con los Semaforos , otra seria con Exclusiones Mutuas de proceso y retenciones y esperas.

Un semáforo binario es un indicador (S) de condición que registra si un recurso está disponible o no. Un semáforo binario sólo puede tomar dos valores: 0 y 1. Si, para un semáforo binario,  $S = 1$  entonces el recurso está disponible y la

tarea lo puede utilizar; si  $S = 0$  el recurso no está disponible y el proceso debe esperar

Retención y espera: los procesos que tienen, en un momento dado, recursos asignados con anterioridad, pueden solicitar nuevos recursos y esperar a que se le asignen sin liberar antes alguno de los recursos que ya tenía asignados.

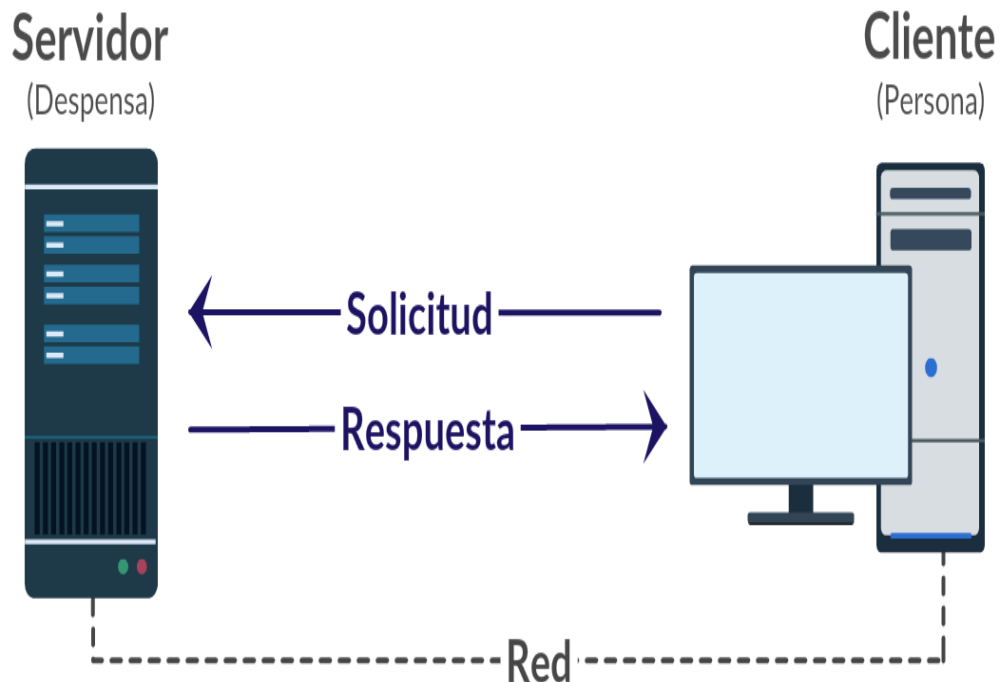
Exclusión mutua: cada recurso está asignado a un único proceso de manera exclusiva.

**12. Cómo se puede garantizar la seguridad en una comunicación entre sockets, explique.**

Por medio de protocolos de seguridad SSL.

Los SSL proporcionan Autenticación y Privacidad de la información entre extremos sobre internet mediante el uso de criptografía, solo el servidor es autenticado( es decir se garantiza si identidad) mientras que el cliente se mantiene sin autenticar.

**13. El tipo de sistema distribuido en la actualidad se llama cliente-servidor, en que consiste. Represente mediante un gráfico un ejemplo concreto.**



En el Servidor encontramos la base de datos y la lógica de la aplicación. En el cliente podemos encontrar todo lo relacionado con las interfaces de acceso al sistema.

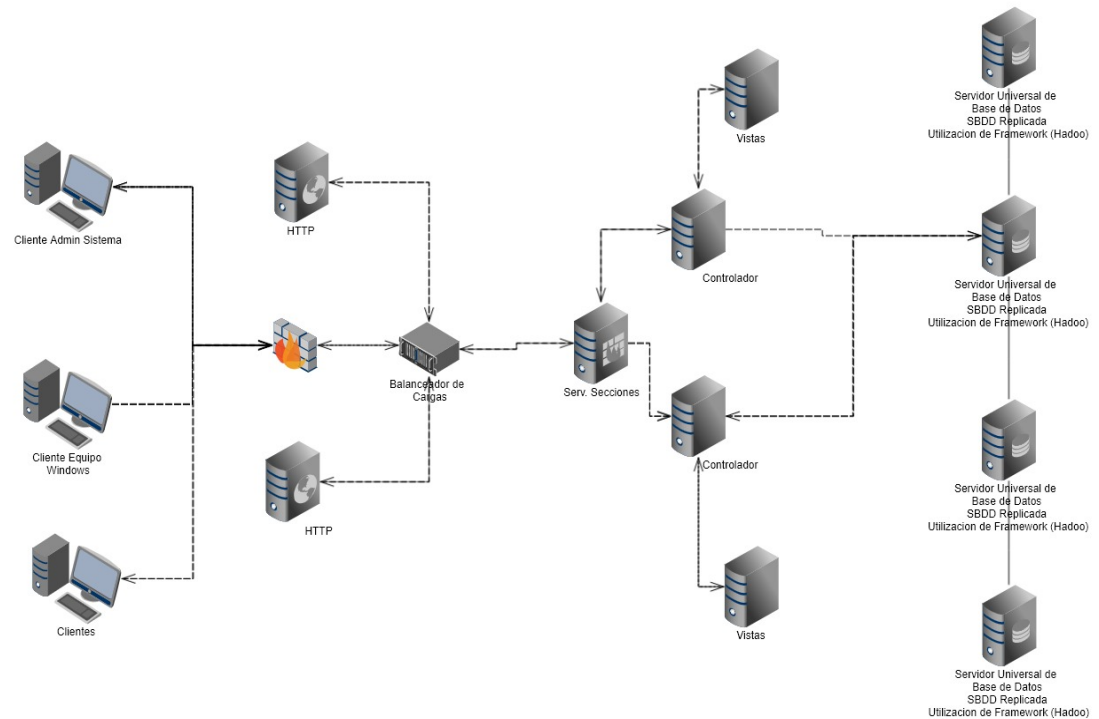
**14. Los servidores pueden ser iterativos o concurrentes, que quiere decir cada concepto.**

Un servidor iterativo es aquel que puede ir atendiendo de a uno a uno en un ciclo para que no haya situaciones bloqueantes, esto se hace de la siguiente manera.

- Se acepta la conexión
- Se lee la petición
- Se lee desde el archivo y se describe en el socket hasta encontrar una marca de fin de archivo.

Los servidores concurrentes atienden a varios clientes al mismo tiempo, es más, mientras atienden a un cliente siguen escuchando más peticiones. Aunque sucede lo mismo que con los servidores iterativos deben esperar su turno para ser atendidos.

**15. Dibuje un esquema de sistema distribuido que permita que gran cantidad de usuarios puedan usar el sistema (usar balanceador de cargas, servidor de bases de datos. servidor apache - nginx, firewall, servidor de archivos, servidor de sesiones, servidor de caché), teniendo en cuenta que un servidor máximo puede atender 50 usuarios simultáneos.**



**16. Explique la función de los servidores proxy en internet en la actualidad. cómo llevan a cabo la lógica para equilibrar carga entre servidores empresariales.**

Un servidor proxy actúa como sustituto de los servidores de contenido de la empresa. Como sustituto, puede configurar el servidor proxy con reglas para direccionar y equilibrar la carga de los clústeres de servidores de contenido.

El servidor proxy también puede proteger el transporte, utilizando SSL (Secure Sockets Layer), y el contenido utilizando diferentes esquemas de autenticación y autorización. Otra característica importante es la capacidad de proteger la identidad de los servidores de contenido de los clientes web utilizando transformaciones de respuesta (reescritura de URL).

El servidor proxy también puede mejorar el rendimiento almacenando localmente el contenido en memoria caché y protegiendo los servidores de contenido cuando aumenta el tráfico.

Una configuración de servidor proxy proporciona valores que controlan cómo puede un servidor proxy proporcionar servicios para las aplicaciones de la empresa y sus componentes. En este apartado se describe cómo crear y configurar servidores proxy en un entorno de servidor de aplicaciones existente.

**17. En qué consiste la comunicación multicast o multidifusión en**

### **un sistema distribuido.**

Multicast.

Multicast es un método de comunicación de envíos simultáneos ( a nivel de IP) que tan solo serán recibidos por un determinado grupo de receptores, que están interesados en los mismos.

Como funciona.

Para que el equipo reciba los mensajes o paquetes, antes deberán estar suscritos a ese grupo haciéndolo saber mediante un mensaje tipo IGMP ( este tipo de mensajes no solo sirve para que un equipo se apunte para recibir paquetes multicast de una dirección, sino también para que un router sepa que en su interfaz tiene un equipo interesado en recibir paquetes de una determinada dirección multicast)

### **18. En qué consiste el concepto fiabilidad en la comunicación por multidifusión.**

Un elemento clave para superar una anomalía de transmisión con la multidifusión es el almacenamiento en búfer de los datos transmitidos (un historial de mensajes que se mantiene en el extremo transmisor del enlace). Este proceso significa que no es necesario ningún almacenamiento en búfer de mensajes en el proceso de la aplicación transmisora, ya que proporciona la fiabilidad.

El tamaño de este historial se configura mediante el objeto de información de comunicación (COMINFO), tal como se describe en la siguiente información. Un almacenamiento intermedio de transmisión mayor significa que hay más historial de transmisión que retransmitir si es necesario, pero debido a la naturaleza de la multidifusión, no se puede dar soporte a una garantía del 100

### **19. Qué tipo de protocolo emplean los sistemas de multidifusión, explique.**

protocolo internet IP.

### **20. Con sus propias palabras, cree en algoritmo que permita la multidifusión de mensajes. Teniendo en cuenta cuando un equipo ingresa, cuando un equipo se retira.**

Se crea un alimentador o nodo desde donde se va a enviar la información por medio de un protocolo de internet IP. Después de manda una solicitud por medio de una mensaje ICGM para poder apuntar a las direcciones que se quiere agregar al grupo multicast. Se adjuntan las direcciones ip para luego poder hacer las multidifusiones de información.

### **21. ¿Qué es un objeto distribuido, y cuándo debe usarse?**

El concepto de objetos distribuidos (de ahora en adelante OD) surgen, como

ya hemos dicho, de la unión de las AD y la POO. Esto viene a romper con el tradicional concepto cliente/servidor, ya que la frontera entre los mismos queda ahora mucho más confusa. Si trabajamos con objetos, puede que uno de ellos (cliente) realice una llamada a otro (servidor). A su vez, este último objeto puede utilizar los servicios de otro, con lo que se convierte en cliente. Con esto ya no tenemos un conjunto de clientes y servidores sino un completo sistema de información, como si tuviéramos todos los objetos en una misma máquina (ahora es la red) funcionando con el mismo programa (aplicación distribuida).

## **22. ¿Qué es el registro RMI, como puede RMI garantizar la seguridad?**

RMI (Java Remote Method Invocation) es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI.

RMI se caracteriza por la facilidad de su uso en la programación por estar específicamente diseñado para Java; proporciona paso de objetos por referencia (no permitido por SOAP), recolección de basura distribuida (Garbage Collector distribuido) y paso de tipos arbitrarios (funcionalidad no provista por CORBA).

A través de RMI, un programa Java puede exportar un objeto, con lo que dicho objeto estará accesible a través de la red y el programa permanece a la espera de peticiones en un puerto TCP. A partir de ese momento, un cliente puede conectarse e invocar los métodos proporcionados por el objeto.

La invocación se compone de los siguientes pasos:

- Encapsulado (marshalling) de los parámetros (utilizando la funcionalidad de serialización de Java).
- Invocación del método (del cliente sobre el servidor). El invocador se queda esperando una respuesta.
- Al terminar la ejecución, el servidor serializa el valor de retorno (si lo hay) y lo envía al cliente.
- El código cliente recibe la respuesta y continúa como si la invocación hubiera sido local.

## **23. Identifique tres características importantes que presenta RMI en la implementación.**

- que sirve para comunicar servidores
- mecanismo para invocar métodos de forma remota
- facilidad de su uso en programación
- posee una arquitectura muy robusta de 4 capas.



**24. La arquitectura RMI básica permite que un cliente invoque un método remoto en un servidor, y reciba una respuesta. Si el servidor debe comunicarse con el cliente solo puede hacerlo en respuesta a una petición anterior de este, a menos que se emplee la técnica de callback. Explique en qué consiste esta técnica**

Como se vio en programación distribuida en claro ejemplo Es un chat, cuando un nuevo participante entra, se avisa al resto de los participantes de este hecho; en un sistema de subastas en tiempo real, cuando empiezan las ofertas, se debe avisar a los procesos participantes.

Esta característica también es útil en un juego en red cuando se informa a los jugadores de la actualización del estado del juego.

Dentro del entorno del API básico básica de RMI presentado presentada en el capítulo anterior, es imposible que el servidor inicie una llamada al cliente para transmitirle alguna clase de información que esté disponible, debido a que una llamada a método remoto es unidireccional (del cliente al servidor).

Una forma de llevar a cabo la transmisión de información es que cada proceso cliente realice un sondeo al objeto servidor, invocando de forma repetida un método remoto, que supóngase que se llama ha comenzado oferta.

El callback de cliente es una característica que permite a un objeto cliente registrarse a sí mismo con un objeto servidor remoto para callbacks, de forma que el servidor pueda llevar a cabo una invocación al método del cliente cuando el evento ocurra. Hay que observar que con los callbacks de clientes, las invocaciones de los métodos remotos se convierten en bidireccionales, o dúplex, desde el cliente al servidor y viceversa.

**25. ¿Cómo funciona el protocolo HTTP? ¿dónde se usa, y que quiere decir que sea un mecanismo sin estado?**

HTTP es una sigla que significa HyperText Transfer Protocol, o Protocolo de Transferencia de Hipertexto. Este protocolo fue desarrollado por las instituciones internacionales W3C y IETF y se usa en todo tipo de transacciones a través de Internet.

El HTTP facilita la definición de la sintaxis y semántica que utilizan los distintos softwares web - tanto clientes, como servidores y proxis - para interactuar entre sí.

Este protocolo opera por petición y respuesta entre el cliente y el servidor. A menudo las peticiones tienen que ver con archivos, ejecución de un programa, consulta a una base de datos, traducción y otras funcionalidades. Toda la información que opera en la Web mediante este protocolo es identificada mediante el URL o dirección.

HTTP es un protocolo sin estado, es decir: no guarda ningún dato entre dos peticiones en la misma sesión. Esto plantea la problemática, en caso de que los usuarios requieran interactuar con determinadas páginas Web de forma orde-

nada y coherente.

## **26. ¿Qué son las acciones GET? POST, PUT, DELETE, cuando usarlas?**

Estas peticiones todas hacen parte de HTTP

GET = obtener

El concepto de GET es obtener información de un Servidor. Se Refiere a traer datos que están en un servidor, ya sea un archivo o base de datos.

POST= Enviar

Post sin Embargo es enviar información desde el cliente para que sea procesada y actualice o agregue información en un servidor. Como seria actualizar una lista de precios, la carga o actualización de una notica.

Cuando enviamos los datos (Request) atraves de un formulario. Estos son procesados y luego atraves de una redirección por ejemplo devolvemos(response) a alguna página con información.

PUT modifica los recursos

El método PUT es usado para solicitar al servidor que almacene el cuerpo de la entidad en una ubicación especifica dada por una URL.

DELETE = Elimina los recursos

Este método es utilizado para solicitar a un servidor que elimina un archivo en una ubicación especifica dada por la URL.

CONNECT

Este método por su parte es usado por los clientes para establecer conexión de red con un servidor mediante web HTTP.

Es la misma que se utiliza para establecer conexión por túneles (Sockerets)

## **27. Al conjunto de comunicaciones de un usuario con un servidor se le denomina sesión, donde es recomendable almacenar dicha sesión, qué técnicas existen.**

El protocolo HTTP no tiene estado: un ordenador cliente que ejecuta un navegador web tiene que establecer una conexión TCP nueva con cada petición GET o POST. El servidor web, por lo tanto, no puede confiar en una conexión TCP establecida para más que una única operación GET o POST.

La administración de sesión es la técnica que utiliza el desarrollador web para dar soporte de estado de sesión HTTP a una sesión sin estado.

Por ejemplo, una vez que un usuario ha sido autenticado en el servidor web, la próxima petición HTTP del usuario (GET o POST) no debería necesitar una

nueva solicitud de usuario y contraseña.

## **28. Que es CGI, que permite.**

Es una tecnología de la Word wide web que permite a un cliente (navegador Web) solicitar datos de un programa ejecutado en un servidor. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor Web y una aplicación externa cuyo resultado final de la ejecución de objetos MINE Que permite.

Permite a un cliente solicitar información a un programa que se ejecuta en un servidor.

También ha hecho posible la implementación de funciones nuevas y variadas en las páginas web.

## **29. Que es CORBA, que ventaja representa sobre RMI**

(CORBA) es un estándar definido por Object Management Group (OMG) que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diferentes computadoras, puedan trabajar juntos; es decir, facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos

## **30. Que es IDL y que permite.**

Es un lenguaje de especificación de interfaces que se utiliza en software de computación distribuida.

Ofrece la sintaxis necesaria para definir los procedimientos o métodos que queremos invocar remotamente. Una vez tengamos esta interfaz creada deberemos pasarla por un compilador de interfaces que generara el proxy o stub cliente y el skeleton o stub servidor.

Permite la especificación de las interfaces y permitir los métodos a invocar posteriormente.

## **31. Que es POA y que permite.**

¿Qué es el adaptador de objetos portátil (POA)?

Un adaptador de objetos es el mecanismo que conecta una solicitud utilizando una referencia de objeto con el código adecuado para atender esa solicitud.

El Adaptador de objetos portátil, o POA, es un tipo particular de adaptador de objetos definido por la especificación CORBA.

El POA está diseñado para cumplir con los siguientes objetivos:

Permita que los programadores construyan implementaciones de objetos que sean portátiles entre diferentes productos ORB.

Proporcionar soporte para objetos con identidades persistentes.

Proporcionar soporte para la activación transparente de objetos.  
Permitir que un solo servidor admita varias identidades de objetos simultáneamente.

**32. Explique mediante un gráfico cómo funciona CORBA con cada uno de sus componentes. 33. Explique el funcionamiento del protocolo SOAP.**

El funcionamiento básico de CORBA pasa por la definición de las interfaces de los componentes mediante el lenguaje IDL.  
Este lenguaje descriptivo nos proporciona mecanismos para definir tipos, módulos, interfaces, operaciones, excepciones, etc. De esta forma, el programador puede definir la forma que tendrán los componentes de su sistema. Tras esto, el código IDL será compilado al lenguaje de implementación (siguiendo unas reglas de mapping estandarizadas), generando los stubs del cliente y los skeletons del servidor.

**34. En qué consiste WSDL. que permite**

WSDL:

WSDL es una notación XML para describir un servicio web. Una definición WSDL indica a un cliente cómo componer una solicitud de servicio web y describe la interfaz que proporciona el proveedor del servicio web. Permite describir un sitio web.

**35. Que es una arquitectura orientada a servicios.**

La Arquitectura Orientada a Servicios (SOA, siglas del inglés Service Oriented Architecture) es un estilo de arquitectura de TI que se apoya en la orientación a servicios. La orientación a servicios es una forma de pensar en servicios, su construcción y sus resultados. Un servicio es una representación lógica de una actividad de negocio que tiene un resultado de negocio específico (ejemplo: comprobar el crédito de un cliente, obtener datos de clima, consolidar reportes de perforación)

El estilo de arquitectura SOA se caracteriza por:

- Estar basado en el diseño de servicios que reflejan las actividades del negocio en el mundo real, estas actividades hacen parte de los procesos de negocio de la compañía.
- Representar los servicios utilizando descripciones de negocio para asignarles un contexto de negocio.
- Tener requerimientos de infraestructura específicos y únicos para este tipo de arquitectura, en general se recomienda el uso de estándares abiertos para la interoperabilidad y transparencia en la ubicación de servicios.
- Estar implementada de acuerdo con las condiciones específicas de la arquitectura de TI en cada compañía.

- Requerir un gobierno fuerte sobre las representación e implementación de servicios.
- Requerir un conjunto de pruebas que determinen que es un buen servicio.

El desarrollo e implementación de una arquitectura SOA se rige por los principios descritos en el manifiesto SOA. Por otra parte la aplicación de la orientación a servicios se divide en 2 grandes etapas:

1. Análisis orientado a servicios (Modelado de servicios)
2. Diseño orientado a servicios

**36. Existen API para facilitar el manejo de SOAP, o incluso aproximaciones arquitectónicas, como REST, explique en qué consiste REST y qué relación tiene con RESTful.**

¿Qué es rest?, es una arquitectura para aplicaciones basadas en redes (como Internet), sus siglas significan REpresentational State Transfer y por otro lado RESTful web service o RESTful api, son programas basados en REST. Pero muchas veces se usan como sinónimos (REST y RESTful).

Para efectos prácticos REST y RESTful pueden entenderse como sinónimos aunque no lo son.

La diferencia entre RESTful web service y RESTful api, es que el api no necesariamente se debe ejecutar en una red, puede ser en una misma computadora.

**37. ¿Qué es un MOM?**

Mensajes Orientados a Middleware

**38. Explique el concepto de paradigmas de agentes móviles.**

los agentes móviles son capaces de ejecutarse en varias máquinas sin necesidad de que en ellas se encuentre su código. Como su nombre lo indica su código es móvil.

También operan sin Conexión. Pueden trabajar si la red no está funcionando. Si los agentes necesitan buscar información en la red el mismo puede esperar a que esta reanude.

Un agente móvil no solo realiza sus funciones en el ordenador que se encuentre, si no en otras que se encuentran en red. Con el fin de buscar información que se haya ordenado. Los mismo tienen la capacidad de decidir a qué servidor moverse. Estos agentes tienen la capacidad para su ejecución y cambiar hacia otro nodo sin cambiar su estado y seguir con su ejecución.

Algunas ventajas de los agentes móviles son:

- Eficiencia.

- Adaptación al cliente.
- Reducción del tráfico de red.
- Gestiona gran volumen de información.
- Permite la comunicación en tiempo Real.

Características de los Agentes Móviles.

Movilidad. Tienen la capacidad de moverse de una maquina a otra.

Proaccion. Toman la iniciativa para alcanzar sus propios objetivos.

Sociabilidad. Tienen la capacidad de comunicarse con otros agentes, personas y programas.

Adaptación. Se comportan según sus experiencias.

### **39. Que es un espacio de objetos**

Es una arquitectura de almacenamiento de dato que maneja los datos como objetos, al contrario que otras arquitecturas de almacenamiento como los sistemas de archivo, que manejan datos como una jerarquía de archivos, y como el almacenamiento de bloque, que maneja datos como bloques dentro de sectores y pistas.

Normalmente cada objeto incluye el propio dato, una cantidad variable de metadatos, y un identificador global único.

El almacenamiento de objetos puede ser implementado en múltiples niveles, incluyendo el nivel de dispositivo (dispositivo de almacenamiento de objetos), el nivel de sistema, y el nivel de interfaz. En cada caso, el almacenamiento de objetos busca habilitar las capacidades no cubiertas por otras arquitecturas de almacenamiento, como interfaces que pueden ser programadas directamente por la aplicación, un espacio de nombres que puede abarcar múltiples instancias de hardware físico, funciones de administración de datos como el manejo de la respuesta de los datos y la distribución de datos a un nivel más detallado para cada objeto.

### **40. ¿Qué es y en qué consiste la computación colaborativa, que ventajas y desventajas tiene?**

Las organizaciones que implementan tecnologías de computación colaborativa lo hacen como una forma de mejorar la productividad y creatividad de la fuerza de trabajo, permitiendo a los trabajadores individuales acceder más fácilmente entre sí y a la información que necesitan, cuando la necesitan.

La computación colaborativa surgió de las primeras generaciones de aplicaciones individuales independientes, tales como la mensajería instantánea y la videoconferencia, que tenían como objetivo superar las distancias geográficas entre las personas que trabajan juntas. Inicialmente, las tecnologías fueron vistas como una forma de imitar o reemplazar las interacciones cara a cara, al tiempo que ofrecían el valor que se obtiene al tener individuos físicamente juntos para co-

laborar.

La computación colaborativa incluye software de colaboración empresarial y herramientas de medios sociales que permiten la mensajería instantánea y los grupos de discusión. También incluye aplicaciones de flujo de trabajo empresarial que automatizan los procesos de trabajo y ayudan a impulsar la toma de decisiones con herramientas de inteligencia de negocios y analítica.

Estos tipos de tecnologías recuperan y comparten datos con las personas que trabajan juntas, quienes pueden actualizarlas o anotarlas, según sea necesario.

Las tecnologías de computación colaborativa pueden funcionar en varios sistemas operativos y dispositivos, permitiendo a los trabajadores individuales participar en sesiones de trabajo desde diferentes lugares, con diferentes equipos. Por ejemplo, dos trabajadores pueden colaborar si uno está usando equipo de videoconferencia en una oficina y el otro está usando una computadora portátil con webcam.

La computación colaborativa se basa en una sólida infraestructura de TI subyacente, como una sólida capacidad de red que puede distribuir, actualizar y almacenar tráfico de video, voz y datos en tiempo real, procedentes de numerosas ubicaciones. Los beneficios de la computación colaborativa pueden ser aumentados por las organizaciones que ajustan su cultura y procesos de trabajo para maximizar su uso.