



INICIO GRABACIÓN



SANJOSÉ
FUNDACIÓN DE EDUCACIÓN SUPERIOR

The background features a photograph of several hands of different skin tones cupping a small globe. A large, semi-transparent blue circle is overlaid on the right side of the image, and a smaller, solid blue circle is positioned at the bottom of the larger one. The word 'COMUNICACION' is centered within the large blue circle.

COMUNICACION



CONCEPTOS

La comunicación entre procesos se encuentra en el núcleo de todos los sistemas distribuidos. No tiene sentido estudiar los sistemas distribuidos, sin examinar cuidadosamente las formas en que los procesos desarrollados en diferentes máquinas pueden intercambiar información. En los sistemas distribuidos, la comunicación siempre se basa en el paso de mensajes de bajo nivel, tal como lo ofrece la red subyacente. Expresar la comunicación a través del paso de mensajes, es más difícil que utilizar primitivas basadas en memoria compartida, como se hace en plataformas no distribuidas.

Los sistemas distribuidos modernos, con frecuencia consisten en miles o incluso millones de procesos esparcidos en una red con comunicación poco confiable como internet. A menos que las primitivas de comunicación para redes de computadoras sean remplazadas por algo más, el desarrollo de aplicaciones de gran escala es extremadamente difícil.





Protocolos en capas



Debido a la ausencia de memoria compartida en los sistemas distribuidos, toda la comunicación se basa en el envío y la recepción (de bajo nivel) de mensajes. Cuando el proceso A debe comunicarse con el proceso B, primero elabora un mensaje en su propio espacio de dirección. Después ejecuta una llamada de sistema y ocasiona que el sistema operativo envíe el mensaje sobre la red hacia B.

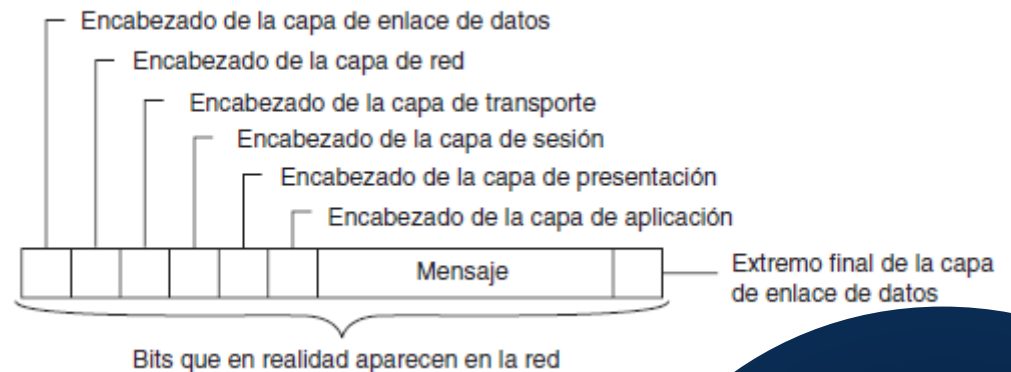
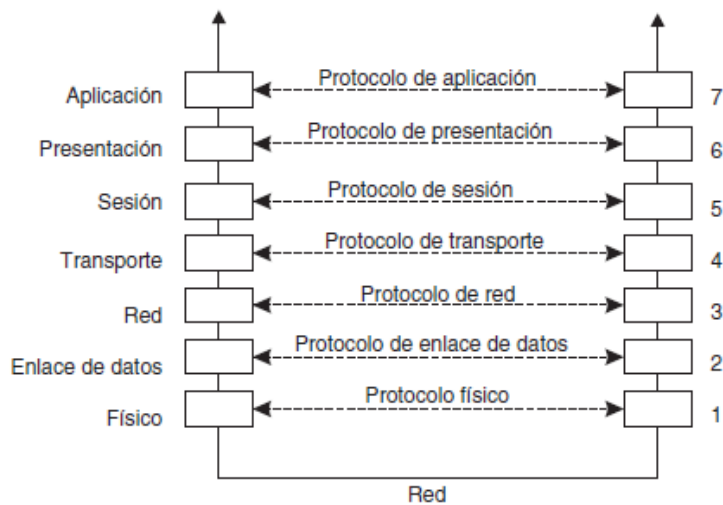
Aunque esta idea básica parece bastante sencilla, para evitar el caos, A y B deben acordar el significado de los bits por enviar. Si A envía una nueva y brillante novela escrita en francés, y codificada con el código de caracteres EBCDIC de IBM, y B espera el inventario de un supermercado escrito en inglés, y codificado en ASCII, la comunicación será menos que óptima.

Para facilitar el manejo de varios niveles y cuestiones involucradas en la comunicación, la International Standards Organization (ISO) desarrolló un modelo de referencia que identifica claramente los diversos niveles involucrados, les proporciona nombres estándar, y señala qué nivel debe hacer cuál trabajo. Este modelo se conoce como el Modelo de Referencia de Interconexión de Sistemas Abiertos (Day y Zimmerman, 1983), normalmente abreviado como ISO OSI o bien, en ocasiones, sólo como el modelo OSI.

MODELO CAPAS OSI



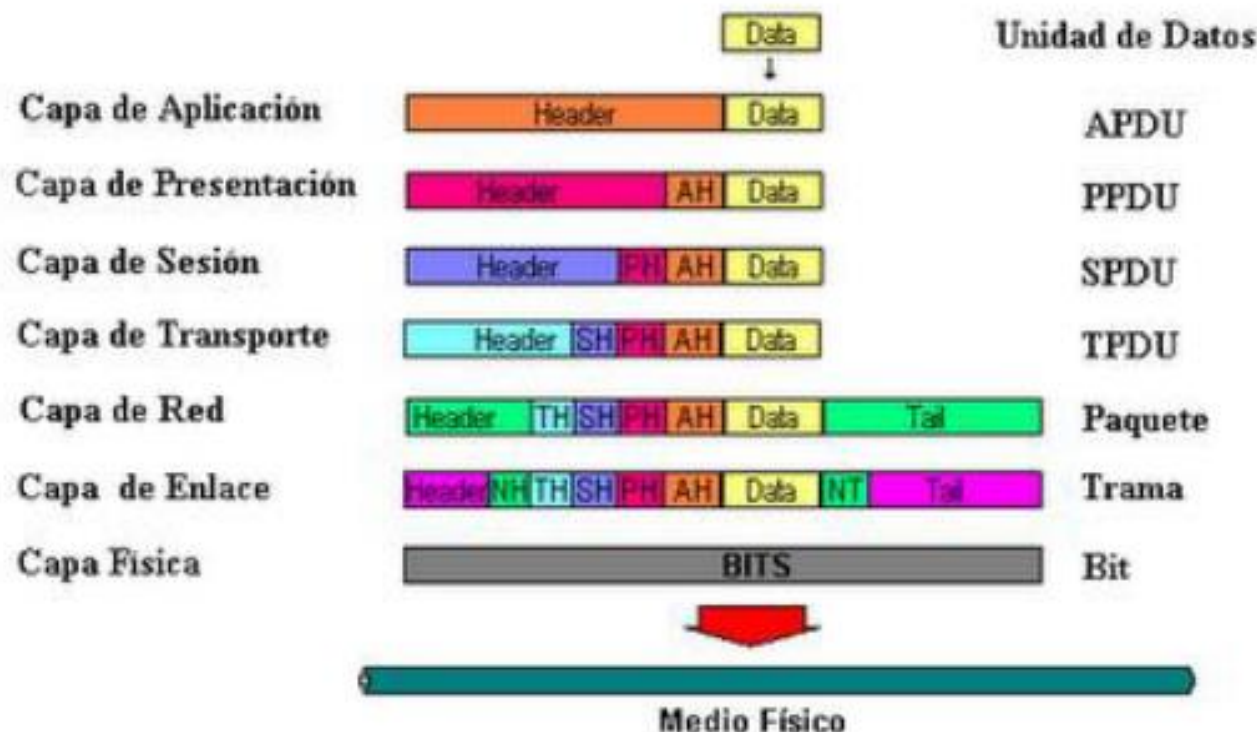
En el modelo OSI, la comunicación se divide en siete niveles o capas, cada capa maneja un aspecto específico de la comunicación. En este sentido, el problema puede dividirse en partes manejables, cada una de las cuales puede resolverse de manera independiente de las otras. Cada capa proporciona una interfaz hacia la siguiente capa superior. La interfaz consiste en un conjunto de operaciones que definen el servicio que la capa podrá ofrecer a sus usuarios.





Protocolos en capas

El modelo OSI está diseñado para permitir que los sistemas abiertos se comuniquen. Un sistema abierto, es aquel que está preparado para comunicarse con cualquier otro sistema abierto mediante reglas estándar que regulen formato, contenido, y significado de los mensajes enviados y recibidos.





Protocolos middleware

El middleware es una aplicación que lógicamente reside (la mayor parte del tiempo) en la capa de aplicación, pero que contiene muchos protocolos de propósito general que garantizan sus propias capas, independientemente de otras aplicaciones más específicas. Podemos establecer cierta diferencia entre los protocolos de comunicación de alto nivel y los protocolos implementados para establecer diversos servicios middleware.

Los protocolos de autenticación no están muy relacionados con alguna aplicación específica, pero pueden estar integrados en un sistema middleware como un servicio general. De igual manera, los protocolos de autorización mediante los cuales se garantiza el acceso a usuarios y procesos autenticados, a los recursos para los que tienen autorización, tienden a ser de naturaleza general e independientes de aplicaciones.

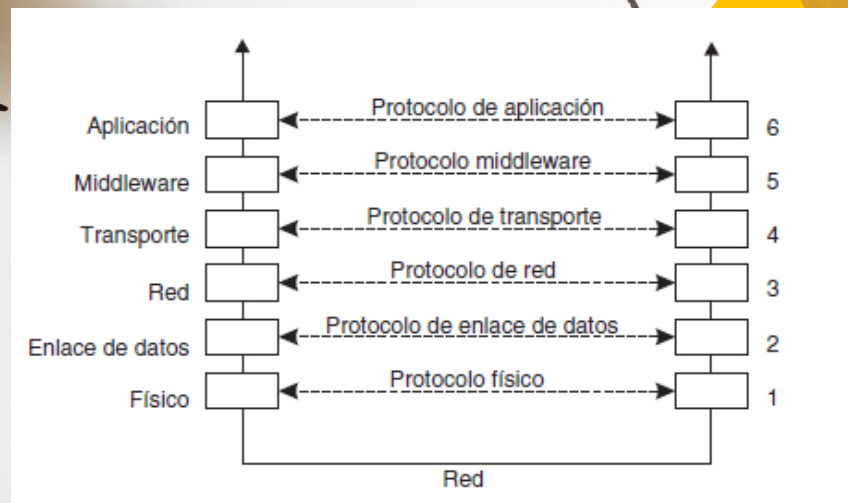
Los protocolos de comunicación middleware soportan servicios de comunicación de alto nivel, existen servicios de comunicación de alto nivel para establecer y sincronizar flujos para transferencia de datos en tiempo real, tales como los necesarios para aplicaciones multimedia, algunos sistemas middleware ofrecen servicios confiables de multidifusión que escalan a miles de destinatarios esparcidos en una red de área amplia.



Protocolos middleware

Algunos de los protocolos de comunicación middleware podrían pertenecer a la capa de transporte, pero es posible que existan razones específicas para mantenerlos a un nivel más alto. Por ejemplo, los servicios de multidifusión que garantizan la escalabilidad pueden implementarse sólo cuando se consideran los requerimientos de la aplicación. En consecuencia, un sistema middleware puede ofrecer diferentes protocolos (ajustables), que a su vez se implementan utilizando diferentes protocolos de transporte, pero ofreciendo una sola interfaz.

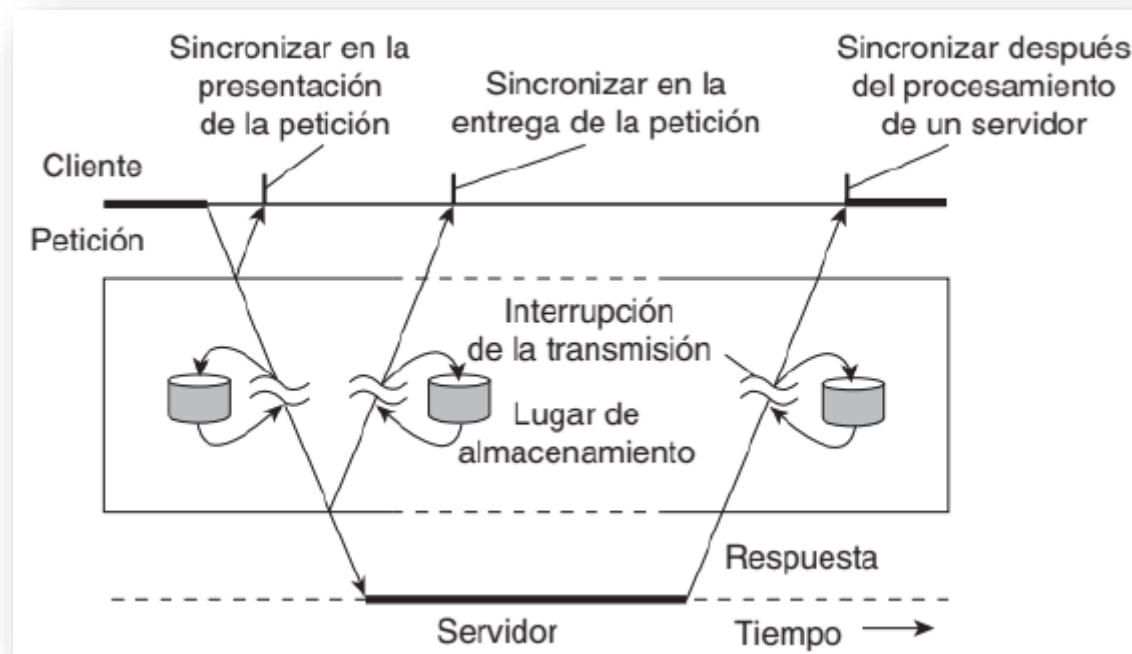
Al adoptar este método de capas llegaremos a un modelo de referencia adaptado para comunicación. En comparación con el modelo OSI, las capas de sesión y presentación se reemplazaron con una sola capa middleware que contiene protocolos de aplicación independientes. Estos protocolos no pertenecen a las capas más bajas que acabamos de explicar.



Tipos de comunicación



Un sistema de correo electrónico es un ejemplo típico en el que la comunicación es persistente. Con comunicación persistente, el middleware de comunicación almacena el mensaje que ha sido presentado para transmitirse el tiempo que tome entregarlo al destinatario. En este caso, el middleware almacenará el mensaje en uno o varios de los lugares de almacenamiento. Como consecuencia, no resulta necesario que la aplicación remitente continúe con la ejecución después de presentar el mensaje. De igual manera, la aplicación destinataria no necesita estar en ejecución cuando el mensaje se presenta.





LLAMADAS A PROCEDIMIENTOS REMOTOS

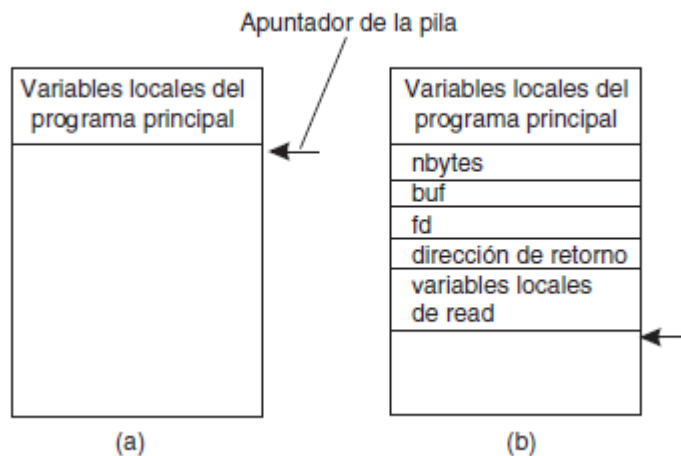
Muchos sistemas distribuidos se han basado en el intercambio explícito de mensajes entre procesos.

Sin embargo, los procedimientos send y receive no ocultan en absoluto la comunicación, lo cual es importante para lograr la transparencia en los sistemas distribuidos.

Operación básica RPC



Para comprender cómo funciona la RPC, es importante que primero comprendamos totalmente cómo funciona una llamada a un procedimiento convencional (es decir, en una sola máquina). Considere una llamada en C como: `count read(fd, buf, nbytes)`; donde `fd` es un entero que indica un archivo, `buf` es un arreglo de caracteres en el que se leen los datos, y `nbytes` es otro entero que indica cuántos bytes leer. Si la llamada se hace desde el programa principal, Para realizar la llamada, el procedimiento que llama coloca los parámetros en la pila, en un orden del último al primero,



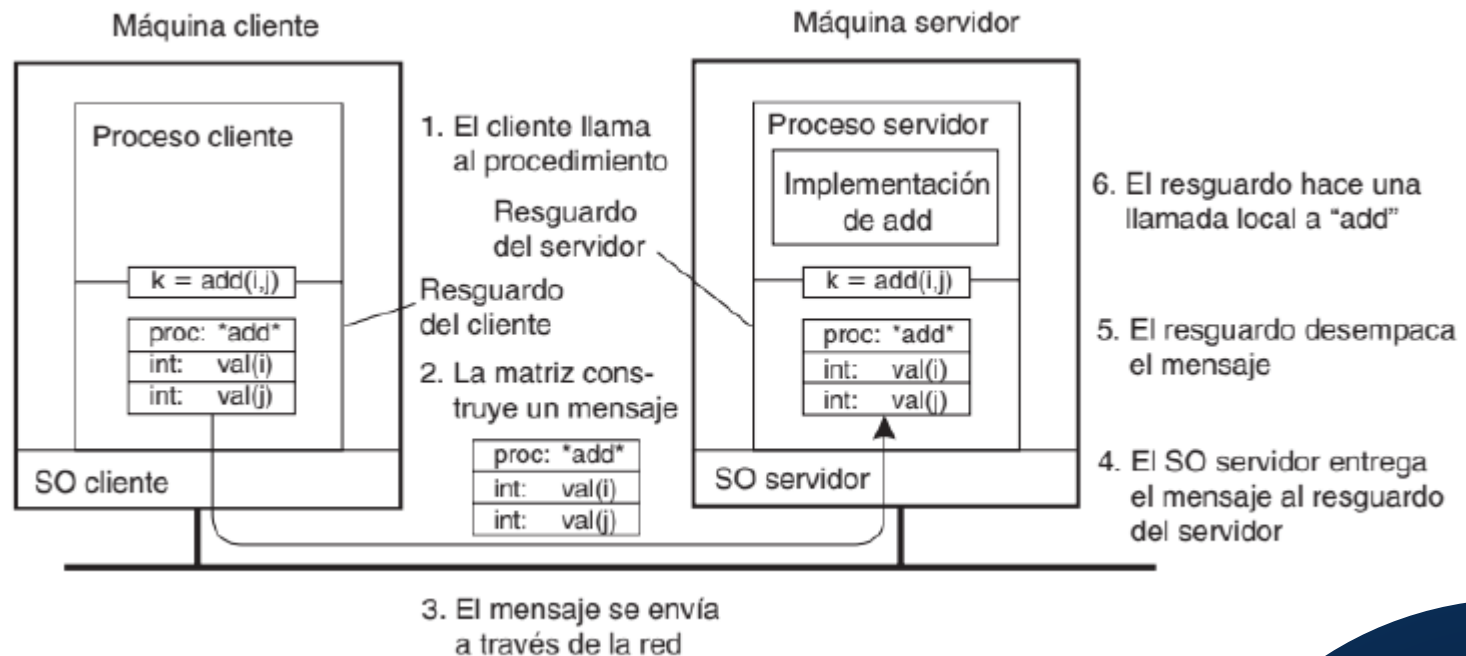
Operación básica RPC

La idea tras la RPC es hacerla parecer como una llamada local. En otras palabras, deseamos que la RPC sea transparente.



Paso de parámetros

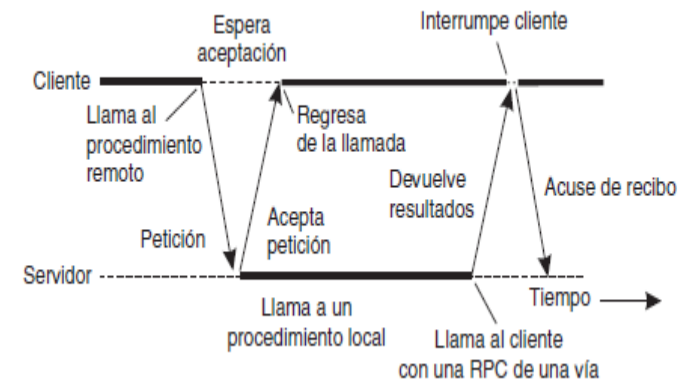
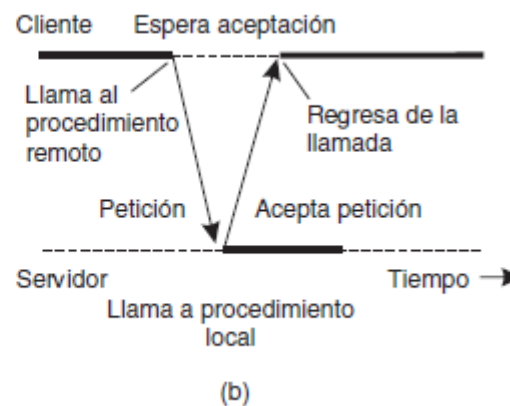
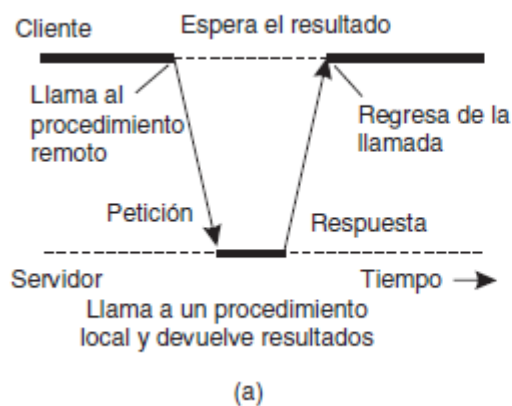
La función del resguardo del cliente es tomar sus parámetros, empacarlos en un mensaje, y enviarlos a la matriz servidor.



RPC asíncrona



Igual que en las llamadas a procedimientos convencionales, cuando un cliente llama a un procedimiento remoto, el cliente se bloqueará hasta que se devuelva una respuesta. El comportamiento estricto de solicitud-respuesta es innecesario cuando no hay un resultado por devolver, y sólo conduce a bloquear al cliente mientras pudo haber continuado y realizar trabajo útil antes de la petición de llamada a un procedimiento remoto. Ejemplos de que a menudo no hay necesidad de esperar una respuesta incluyen: la transferencia de dinero de una cuenta a otra, agregar entradas a una base de datos, iniciar servicios remotos, el procesamiento por lotes, etcétera.



The background features a photograph of two hands shaking in a firm grip, symbolizing agreement or partnership. This image is partially covered by two overlapping circles: a light blue one on the left and a dark blue one on the right. The text is centered within the light blue circle.

COMUNICACIÓN ORIENTADA A MENSAJES

Comunicación transitoria orientada a mensajes



Muchos sistemas y aplicaciones distribuidos se construyen directamente sobre el sencillo modelo orientado a mensajes ofrecido por la capa de transporte.

interfaz de sockets: En cuanto a su concepto básico, un socket es un punto final de comunicación en el que una aplicación puede escribir información destinada a enviarse fuera de la red subyacente, y desde el cual puede leerse información entrante. Un socket forma una abstracción sobre el punto final real de comunicación, el cual utiliza el sistema operativo local para un protocolo de transporte específico.

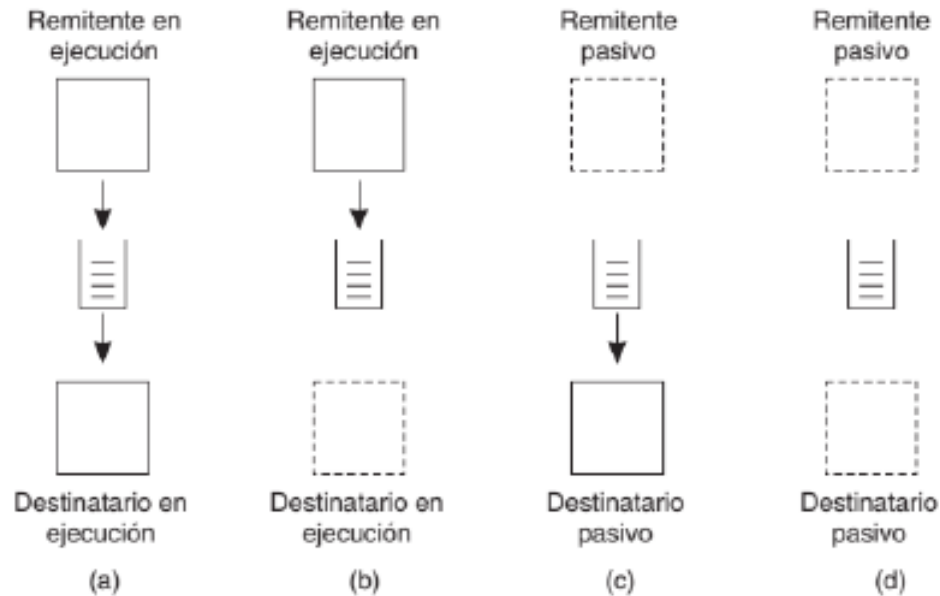
Por lo general, los servidores ejecutan primero cuatro primitivas, normalmente en el orden dado. Cuando se llama a la primitiva socket, el que llama crea un nuevo punto final de comunicación para un protocolo de transporte específico. De manera interna, crear un punto final de comunicación significa que el sistema operativo local reserva recursos para alojar los mensajes enviados y recibidos por el protocolo especificado.

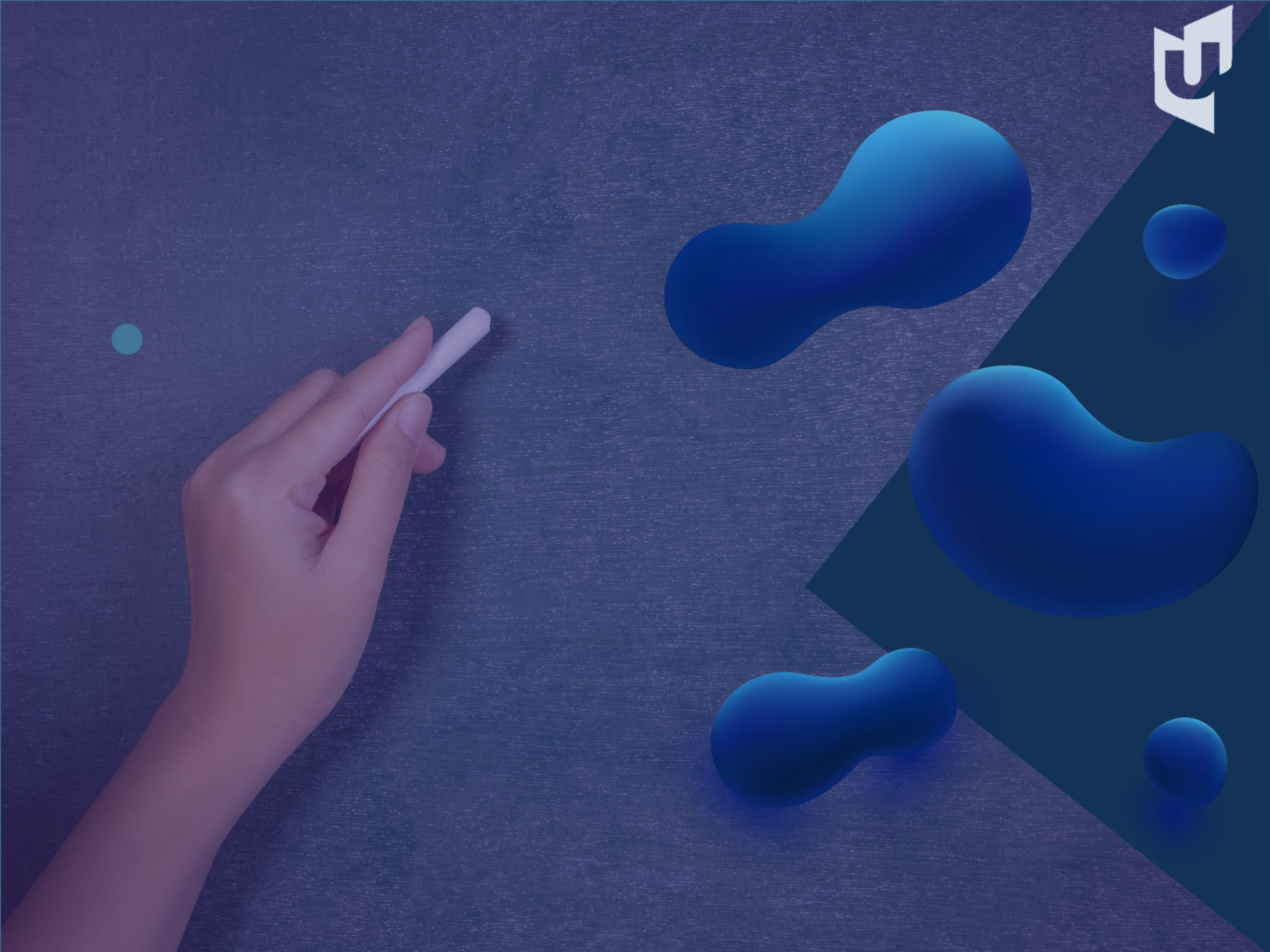
| Primitiva | Significado |
|-----------|--|
| Socket | Crea un nuevo punto final de comunicación |
| Bind | Asocia una dirección local a un socket |
| Listen | Anuncia la conveniencia de aceptar conexiones |
| Accept | Bloquea a quien llama hasta que llega una petición de conexión |
| Connect | Activa el intento de establecer una conexión |
| Send | Envía algunos datos a través de la conexión |
| Receive | Recibe algunos datos a través de la conexión |
| Close | Libera la conexión |

Comunicación persistente orientada a mensajes



Los sistemas de colas de mensajes proporcionan un amplio soporte para comunicación asíncrona persistente. La esencia de estos sistemas es que ofrecen capacidad de almacenamiento de término medio para mensajes, sin la necesidad de que el remitente o el destinatario estén activos durante la transmisión del mensaje.

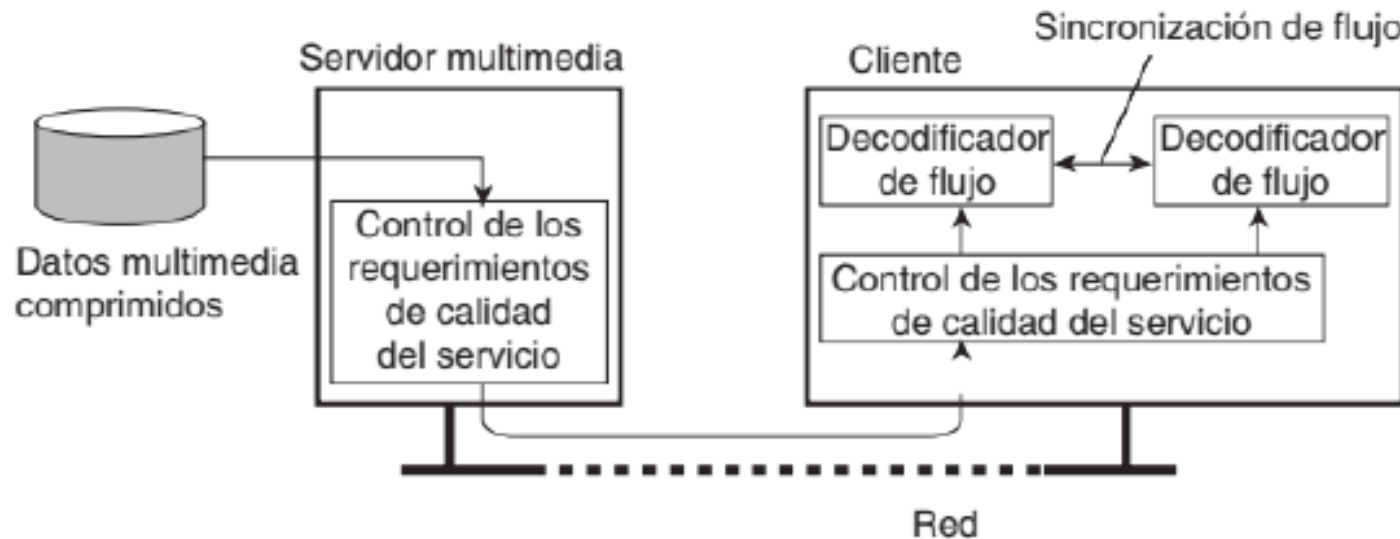






Soporte para medios continuos

El soporte para intercambiar información dependiente del tiempo con frecuencia se conoce como soporte para medios continuos. Un medio se refiere al recurso mediante el cual se transmite la información.



Arquitectura general para pasar a través de una red un flujo de datos multimedia almacenados.



FUNDACIÓN DE EDUCACIÓN SUPERIOR

SAN JOSÉ

INSTITUCIÓN TECNOLÓGICA

FIN DE
GRABACIÓN